This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Callback Administrator's Guide

7/23/2025

Table of Contents

Get started	
Getting started with Callback	4
Callback scenarios	9
Provisioning Callback in Designer	14
Controlling user access	27
Genesys Multicloud CX REST APIs and tutorials for Callback	33
Manage callbacks	
Managing callbacks	36
Configuring Country and Number Validation rules	49
Configuring Capacity rules	59
Configuring Region Affinity rules	73
Monitor callback queues	79
Manage credentials for features	
Provisioning Push Notifications	85
Provisioning CAPTCHA	92
Registering for access to statistics	96
Provision and monitor Click-To-Call-In	
Provisioning the Click-to-Call-In scenario	97
Configuring Click-To-Call-In Groups	105
Viewing Click-To-Call-In records	113
Troubleshooting and advanced administration tools	
Troubleshooting and validating functionality	116

Search the table of all articles in this guide, listed in alphabetical order, to find the article you need.

Related documentation:

-

Getting started with Callback

Contents

- 1 What is a callback?
- 2 Getting started with Genesys Callback
  - 2.1 Create and manage Callback resources
  - 2.2 Provision Callback
  - 2.3 Start using the Callback UI
- 3 Overview of the Callback UI
- 4 Callback reports
- 5 Specifying your language preference

-
- Administrator

Genesys Callback, available from the Genesys portal, provides traditional callback services, user-originated Click-To-Call scenarios, and supports the integration of Push Notifications and CAPTCHA widgets. This article includes information about where to begin if you are new to Callback and what steps are required to get started using the Callback UI.

Related documentation:

-

Businesses sometimes cannot offer on-demand, low-wait agent help because of resource limitations or increased service usage. In these situations, the best option is to offer some form of deferred service that can connect consumers and agents at a later, mutually-beneficial time.

Genesys Callback, which takes its name from the most typical form of deferred agent voice connection, supports a user-originated call-in feature, in addition to a range of scenario and sourcing options. In short, Genesys Callback is much more than callback functionality.

Genesys Callback is enabled by Genesys Engagement Services (GES) behind-the-scenes and you might see references to GES or "Engagement Services" throughout Callback pages or within resource names.

What is a callback?

A callback is a data record that contains data for calling a customer at some time in the future. Callbacks generally consist of attached data and configuration data. Attached data is specific to the end customer and might be used in screen pops, routing, reporting, and other applications. Configuration data describes how a callback is fulfilled; this includes target expressions, timeout values, and virtual queue names.

Using Callback, you can associate a callback service with a phone number, which enables a caller to request a callback. Depending on which scenario you choose to implement, the callback is then routed to an agent who processes the call.

Getting started with Genesys Callback

Before you start working in the Callback UI, you need to provision resources, access permissions, Designer applications for callbacks, and configure the callback services. This section provides information about where to start. For additional information, also see Providing callback services with Callback.

Create and manage Callback resources

Use Platform Administration and Agent Setup to create and manage contact center resources and user access to information and features in the Callback UI. In Platform Administration, you create the virtual queues that will be used for callbacks. The queues are then configured in Designer for Callback use.

To get started, go to the Provisioning Callback page. The page tells you what contact center resources you'll need to provision in Platform Administration or Agent Setup. For general information about configuring agents, including how to assign skills to each agent, see Create and manage agent skills and Manage agents and other users. For information about creating virtual queues, see Virtual Queues.

After the resources are provisioned, you can configure user access permissions for Callback. Genesys provides predefined roles for Callback to ensure that your users have UI access permissions that are appropriate for your business needs. If you are new to Genesys Callback, then you might have to assign resources (users) to appropriate access groups. In addition, you have the option to restrict access to queues (and lists of callbacks) in the Callback UI based on your lines of business. Read Overview of the Callback UI to understand which roles are required for access to each tab in the Callback UI. For detailed information about Callback-specific roles and access groups, see Controlling User Access.

Provision Callback

You use Designer to provision callback-related applications and virtual queues. Callback works with Designer to do such things as route calls, connect consumers to agents with the correct skills, time out and purge expired or abandoned callbacks, determine if a callback is possible based on a contact center's office hours, and terminate callbacks when appropriate. The Provisioning Callback in Designer page guides you through the Designer application provisioning process, including how to determine which type of Designer applications you require for your callback scenario.

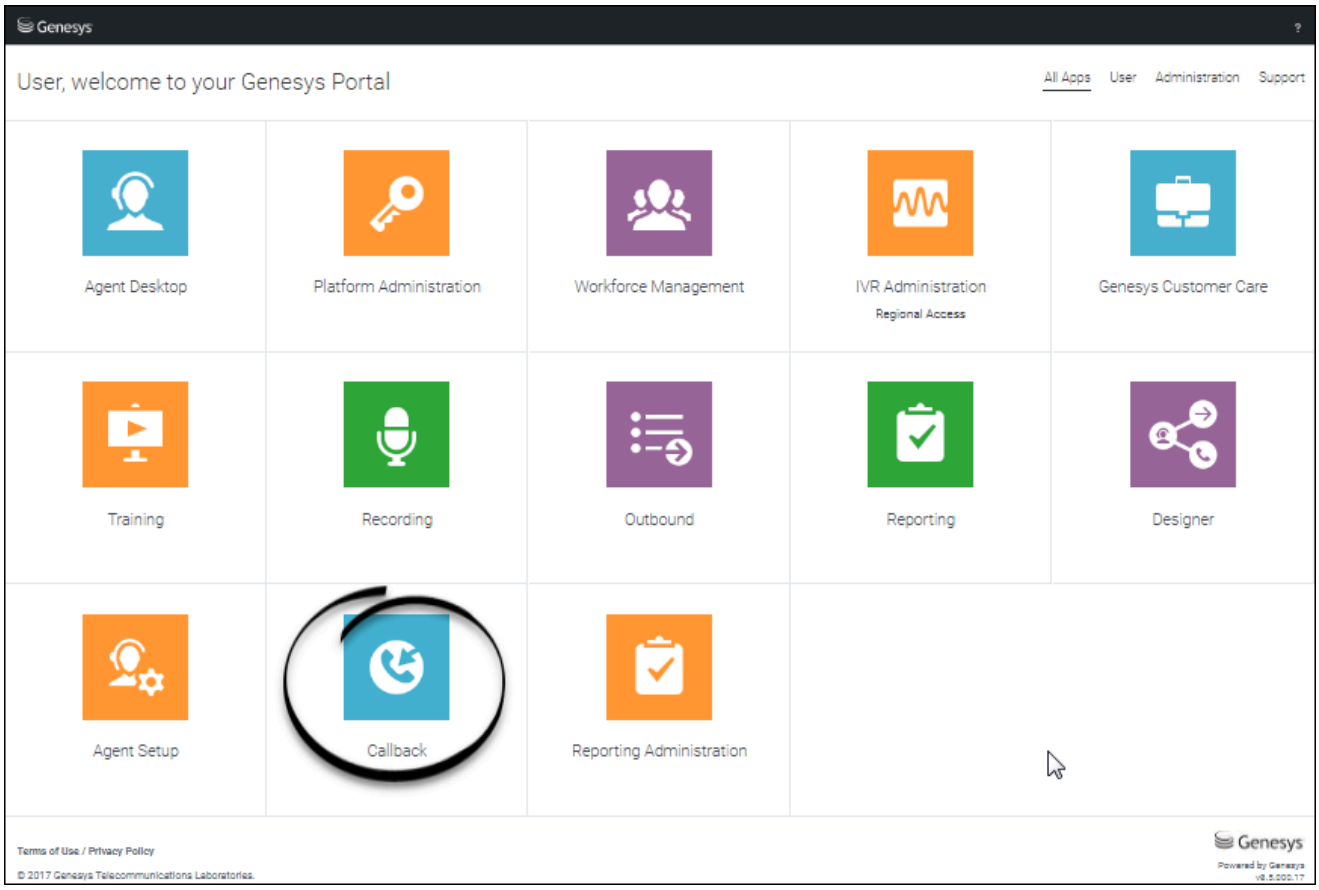
Start using the Callback UI

After you have completed Callback provisioning and testing to ensure that calls are routed correctly, and Callback users have been assigned to the correct roles and access groups, you can begin to use the Callback UI. The **Callback** tab displays the list of callback records. Users with sufficient permissions use the **Callback** tab to manage the callback records, including creating, editing, or cancelling callbacks.

Callback Administrators and Developers have access to a **Developer** tab in the Callback UI. Use the **Developer** tab to manage callback activity and features at a more technical level. For example, you can check for errors in Callback API queries or validate API keys. To learn more about the tools on the **Developer** tab, see Troubleshooting and validating functionality.

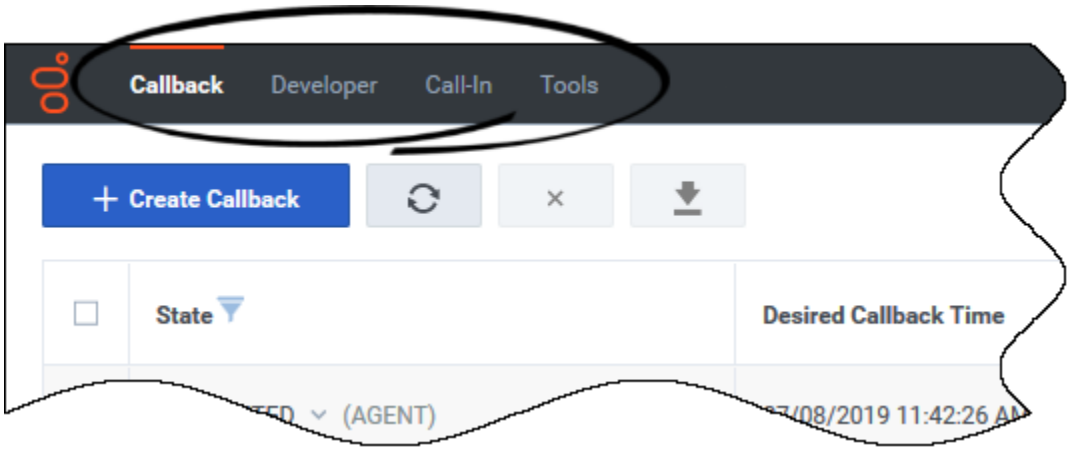
If you work with GES/Callback REST APIs, you can find links to tutorials and additional information on the Genesys Multicloud CX REST APIs and Tutorials for Callback page in the Callback documentation.

Overview of the Callback UI



Click the **Callback** icon on the Genesys Portal to open the UI. The Callback UI is supported in the following browsers:

- Firefox
- Google Chrome
- Microsoft EdgeHTML version 16.0 and up and on Chromium-based Edge



The Callback UI has multiple tabs; you might see all available tabs or only some of the tabs. Administrators will make sure that you have access to the tab or tabs that you need to do your job. Access is based on the role to which you have been assigned.

By default, your Callback UI session times out after 20 minutes of inactivity.

The following tabs are available in the Callback UI:

- **Callback**: Displays the list of callbacks. In some scenarios, you might need to create or monitor a callback request, or even cancel a set of callbacks. Callback Monitor is the minimum role required to access the **Callback** tab. To modify callback records, you require the Callback Administrator or Callback Supervisor role.
- **Developer**: You must be a member of the Callback Administrator or Callback Developer role to access the **Developer** tab. From the **Developer** tab, you can access developers' tools such as the list of errors related to your Callback API queries as well as pages on which you can validate your API keys or provision Push Notifications.
- **Call-In**: You must be a member of the Callback Administrator or Callback Developer role to access the **Call-In** tab. If you have the Click-To-Call-In scenario provisioned, use the **Call-In** tab to view the Click-To-Call-In records.
- **Tools**: You must be a member of the Callback Administrator, Callback Supervisor, or Callback Developer role to access the **Tools** tab. The **Tools** tab offers additional views and tools to assist with callback management, configuration, and troubleshooting. For example, you can search for a specific interaction within the callback records or check the EWT on a specific callback queue.

Callback reports

For information about real-time reporting for callbacks, see Real-time Reporting with Pulse. Callback activity is tracked as part of the queue reports. You can add a report widget that is based on the Callback Activity template to your Pulse dashboard.

For information about Callback historical reporting using Genesys Customer Experience Insights (GCXI), see Callback reports, Callback Details Report, and Callback Summary Report in Work with Genesys CX Insights Reports.

Callback scenarios

Contents

- 1 General information about callbacks
- 2 Immediate callbacks
- 3 Scheduled callbacks
- 4 Click-To-Call-In (Immediate)
- 5 Click-To-Call-In (Delayed)
  - 5.1 How Click-To-Call-In (Delayed) works

-
- Administrator

This article summarizes the scenarios that Genesys Callback supports. In addition to the traditional Immediate and Scheduled callback scenarios, Callback also supports Click-To-Call-In scenarios, allowing consumers to initiate a call to your contact center by tapping a Click-To-Call button in your mobile app.

Related documentation:

-

General information about callbacks

You configure callback scenarios in Designer.

It does not matter how a callback originates, the voice interaction is always converted to a virtual call and added to the queue where it is monitored so the system can provide information such as the Estimated Wait Time (EWT) to future callers.

Immediate callbacks

An *Immediate* callback is a callback that is set in motion when your customer (a *consumer*) makes a request to be called back as soon as an agent who can provide assisted service becomes available. A consumer can request an Immediate callback in the following ways:

- While the consumer is in-queue on an IVR:
  - A consumer's call arrives and the caller is offered Immediate callback. If the caller accepts, he or she confirms the phone number for the callback.
- Through an API call; in other words, the consumer makes the request from a mobile app or website:
  - A consumer is using your company's or organization's mobile app or website and encounters a situation where he or she requires assisted service by voice. The consumer taps or clicks a button to request a callback, confirms or provides the number at which he or she would like to receive the call, and receives a confirmation message that the request was received.

No matter how the Immediate callback is requested, when the consumer reaches the top of the queue, the callback is attempted as soon as an agent who satisfies the required skill expression enters the Ready state. The agent that triggered the callback is not reserved, though, and might take another call while the system tries to reconnect with the consumer. When the system has successfully connected to the consumer, then the call is routed to the next agent with the correct skills who enters the Ready state.

Because user/context data might be attached to the API request for a callback, key components of the consumer's app or web journey can be preserved for agent or reporting use.

For information about provisioning Callback, see Provisioning Callback in Designer.

Scheduled callbacks

A *Scheduled* callback is a callback that is set in motion when your customer (a consumer) makes a request to be called back in the future, at an approximate time that works for the consumer's schedule. A consumer can request a Scheduled callback in the following ways:

- While in-queue on an IVR:
  - A consumer's call arrives and the caller is offered the option to schedule a callback for some time in the future. If the caller accepts, he or she confirms the phone number for the callback and is prompted to input the time at which he or she would like to receive it. Using the caller's requested time, the system searches for the closest-matching available time to connect with an agent.
- Through an API call (from a mobile app or website):
  - A consumer is using your company's or organization's mobile app or website and encounters a situation where he or she requires assisted service by voice. The consumer taps or clicks a button to request a callback, confirms or provides the number at which he or she would like to receive the call, interacts with a date/time picker to search for availability, and receives a confirmation message that the request was received.

No matter how the Scheduled callback is requested, the system – using the desired callback time as the target time slot – makes the outbound call to the consumer when the consumer reaches the top of the queue and an agent with the required skills enters the Ready state. The agent that triggered the callback is not reserved, though, and might take another call while the system tries to reconnect with the consumer. When the system has successfully connected to the consumer, then the call is routed to the next agent with the correct skills who enters the Ready state.

Because user/context data might be attached to the API request for a callback, key components of the consumer's app or web journey can be preserved for agent or reporting use.

For information about provisioning Callback, see Provisioning Callback in Designer.

Click-To-Call-In (Immediate)

> **Important**
>
> To implement this scenario, you need to use the corresponding Call-In API to initiate the Click-To-Call-In request.

A Click-To-Call-In (Immediate) interaction is set in motion when your customer (a consumer) taps a button in a mobile app that is designed to trigger a Call-In API request:

- A consumer is using your company's or organization's mobile app and encounters a situation where he or she requires assisted service by voice. The consumer taps a button that you have provisioned in your app to connect consumers to your contact center.
- The system responds with call-in details immediately. Using that information, the app triggers a call to your contact center. The system attempts to match the caller to existing information. For more information, see Provisioning the Click-to-Call-In scenario.
- If the attempt to match the caller to a Call-In request is successful and your Designer application is configured to route the call when a match is made, then the call is queued on hold like any other call to the contact center. If the consumer is placed in a queue where the EWT is above the configured threshold, then the consumer might be offered a callback option.

Because user/context data might be attached to the API request, key components of the consumer's app journey can be preserved for agent or reporting use.

Click-To-Call-In (Delayed)

> **Important**
> To implement this scenario, you need to use the corresponding Call-In API to initiate the Click-To-Call-In request.

A Click-To-Call-In (Delayed) interaction is set in motion when your customer (a consumer) taps a button in a mobile app that is designed to trigger a Callback API request. The following description of what happens next is a brief summary. See How Click-To-Call-In (Delayed) Works for additional information. To provision the Click-To-Call-In scenario, see Provisioning the Click-to-Call-In scenario.

- A consumer is using your company's or organization's mobile app and encounters a situation where he or she requires assisted service by voice. The consumer taps a button to contact your center.
- The mobile app sends a callback request, which includes Push parameters that the system uses to contact the consumer to provide information about the callback.
- The call is queued in the Click-To-Call-In (Delayed) virtual queue.
- When the consumer reaches the top of the queue, the system sends a Push Notification to the mobile app to notify the consumer that the callback is ready.
- When the consumer accepts the callback, the system immediately replies with a Push Notification that provides a phone number to call and, if configured, a unique access code that the consumer will be asked to enter before the call is initiated.
- When the consumer calls in and enters the access code, if required, the system attempts to match the caller to an existing callback request. When a match is made, the caller is queued and routed to the next available agent who satisfies the required skill expression.

Because user/context data might be attached to the API request, key components of the consumer's app journey can be preserved for agent or reporting use.

How Click-To-Call-In (Delayed) works

Your mobile app sends a Callback request when the consumer taps the button or link that you have provisioned for the Click-To-Call-In Delayed feature. The call is then queued in the Click-To-Call-In (Delayed) virtual queue. The Callback request includes Push parameters. To use Push Notifications with Callback, see Provisioning Push Notifications.

A Click-To-Call-In (Delayed) request is routed to an agent when the following criteria are met:

1. The Call-In request can be matched to an existing Callback request in the system.
2. Your Click-to-Call-In Match Designer application is configured to route the call.
3. An agent (with the correct skills if skills are configured) is ready to accept the call.

There are, however, a number of things that can happen during an active Click-To-Call-In (Delayed) session that might impact the session's flow. For example, the system might fail to match a Click-To-Call-In (Delayed) request on the first attempt. As long as the Click-To-Call-In and Callback requests remain valid and outstanding, though, the consumer can call and try again.

There are also Designer settings that can purge the callback from the system or remove the callback from its queue:

1. The **Callback Purge Time (minutes)** for the virtual queue is reached before the consumer responds to Push Notifications or before an agent is available to assist the caller. In this case, the callback is purged from the system.
2. The end of the business day, based on the configured Business Hours, occurs before the **Callback Purge Time (minutes)** is reached, before the consumer responds to Push Notifications, or before an agent is available to assist the caller. In this case, the callback is purged from the system.
3. The **Pushed Callback Expiry Time (minutes)** setting in Designer causes the callback session to terminate because the specified time interval expires. When this happens, the callback is removed from the queue, but not purged from the system until one of the preceding two events occurs.

A Call-In API request that is associated with a Callback request that was purged or terminated will fail because no match can be made between the Call-In and Callback requests.

Provisioning Callback in Designer

Contents

- 1 Before you start
  - 1.1 Provision the callback virtual queues
- 2 Provisioning your first callback scenario in Designer
  - 2.1 Create your Designer applications
- 3 Provision a Designer application to offer callback through the IVR
- 4 Provision the Designer callback application
- 5 Provision business hours for Callback
- 6 Provision the callback services
  - 6.1 Adding a Callback virtual queue to the CALLBACK_SETTINGS data table
- 7 Testing your callback scenario

-
- Administrator

Callback is provisioned with Designer. To create and configure Callback services, Designer includes a set of blocks dedicated to Callback. This article provides information about how to provision a basic callback scenario in Designer. For information about the supported callback scenarios, see Callback Scenarios. For information about provisioning the Click-To-Call-In scenario in Designer, see Provisioning the Click-to-Call-In scenario.

Related documentation:

-

Before you start

Before you provision Callback in Designer, make sure that the following objects are created in Platform Administration and ready for use:

- A Callback Administrator.
- Route Points to be used for Inbound strategies that offer Callback.
- Route Points to be used for Outbound strategies, if Callback calls to consumers will be handled by separate Outbound strategies.
- Virtual Queues to store callbacks. Genesys recommends that you use three queues for callbacks. For more information about the virtual queues, see Provision the callback virtual queues.
- At least one agent who will process Callbacks.

> **Important**
> Genesys Callback supports voice calls only. Digital interactions are not supported.

Provision the callback virtual queues

In addition to an Inbound virtual queue, Genesys recommends that you have two additional virtual queues for callback reporting purposes. That means that you will have the following three queues for callbacks:

- Inbound virtual queue
- Callback virtual queue
- Outbound virtual queue

Genesys recommends that you use the following naming conventions for the queues:

- Inbound virtual queue: _VQ
  For example, `Sales_VQ`
- Callback virtual queue: _VQ_CB
  For example, `Sales_VQ_CB`
- Callback outbound virtual queue: _VQ_CB_OUT
  For example, `Sales_VQ_CB_OUT`

You might see the term *service* or *service name* in callback-related applications, Widgets, APIs, and in the Callback UI. The service, in this context, is a virtual queue. The service name, therefore, refers to the name of the virtual queue. For example, on the **Callback** page in the UI, the **Service Name** column identifies, by name, the virtual queue associated with each callback.

Having all three callback-related virtual queues provides the following functionality:

- For each call type, the system can keep track of and compare Estimated Wait Time (EWT) and other important queue statistics separately.
- You can configure both historical and real-time reporting. While the Inbound and Callback virtual queues collect statistics such as EWT and which calls accept the callback offer and which calls reject it, the Outbound virtual queue collects data for outbound interactions such as how long the customer had to wait for an agent to connect during the callback attempt.

After you create the virtual queues that will be used in your callback scenario, you must provision applications and Callback services in Designer. The virtual queues that you have created for callback functionality will be required to complete the Designer application and services provisioning.

Provisioning your first callback scenario in Designer

The following Callback provisioning workflow assumes that you have already completed the configuration described in the Before you start section, above.

- Provision Callback for the inbound strategy. For information, see Create your Designer Applications and Provision a Designer Application to Offer Callback Through the IVR.
- Provision Callback for the outbound strategy. For information, see Create your Designer Applications and Provision the Designer Callback Application.
- Provision Business Hours for Callback.
- Provision the Callback services in Designer.
- Test your Callback scenario configuration.

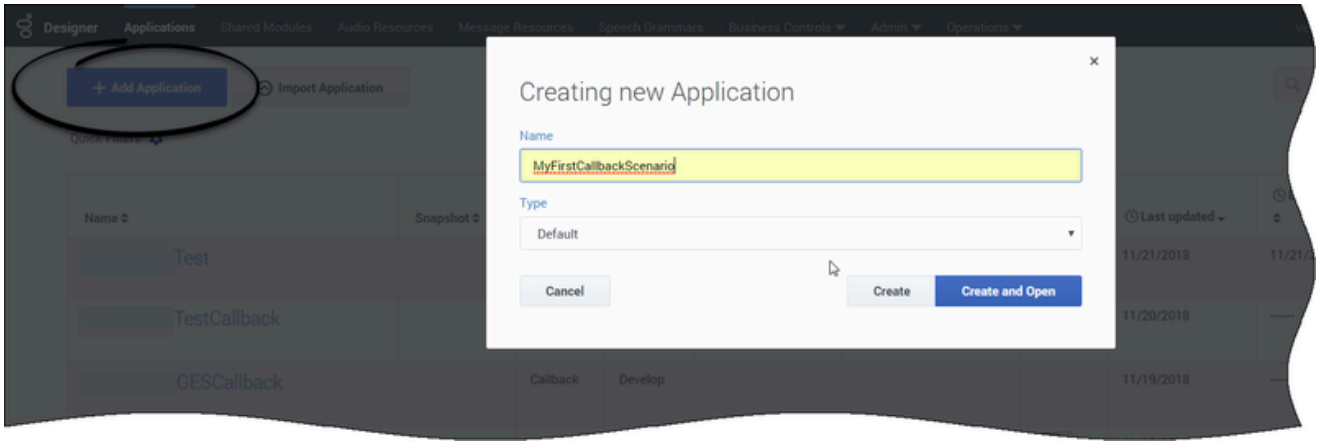
Create your Designer applications

The following table provides information about the types of Designer applications that you require for each supported callback scenario. Create the Designer applications after you have created the Inbound, Callback, and Outbound virtual queues and before you provision the Callback services.

Callback Scenario	Designer Application Type
In-Queue Callback	Default + Callback
Scheduled Callback	Default + Callback
Web Callback	Callback

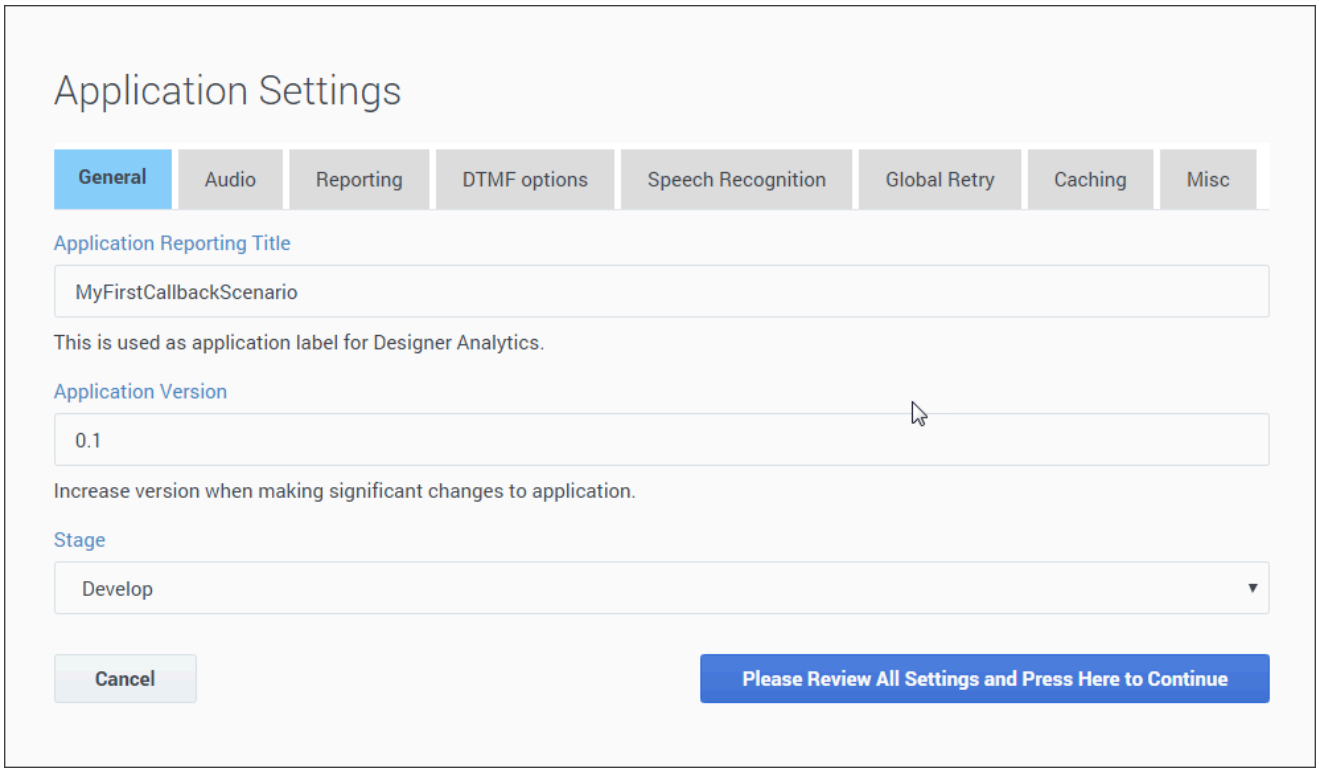
The following procedures show you how to provision Designer applications for Callback. The Default-type Designer application provides the Callback offer to a customer waiting in a queue; for example, in a scenario where the customer is connected to an IVR. The Callback-type Designer application provides the Callback attempt for scheduled and web callback scenarios.

Provision a Designer application to offer callback through the IVR

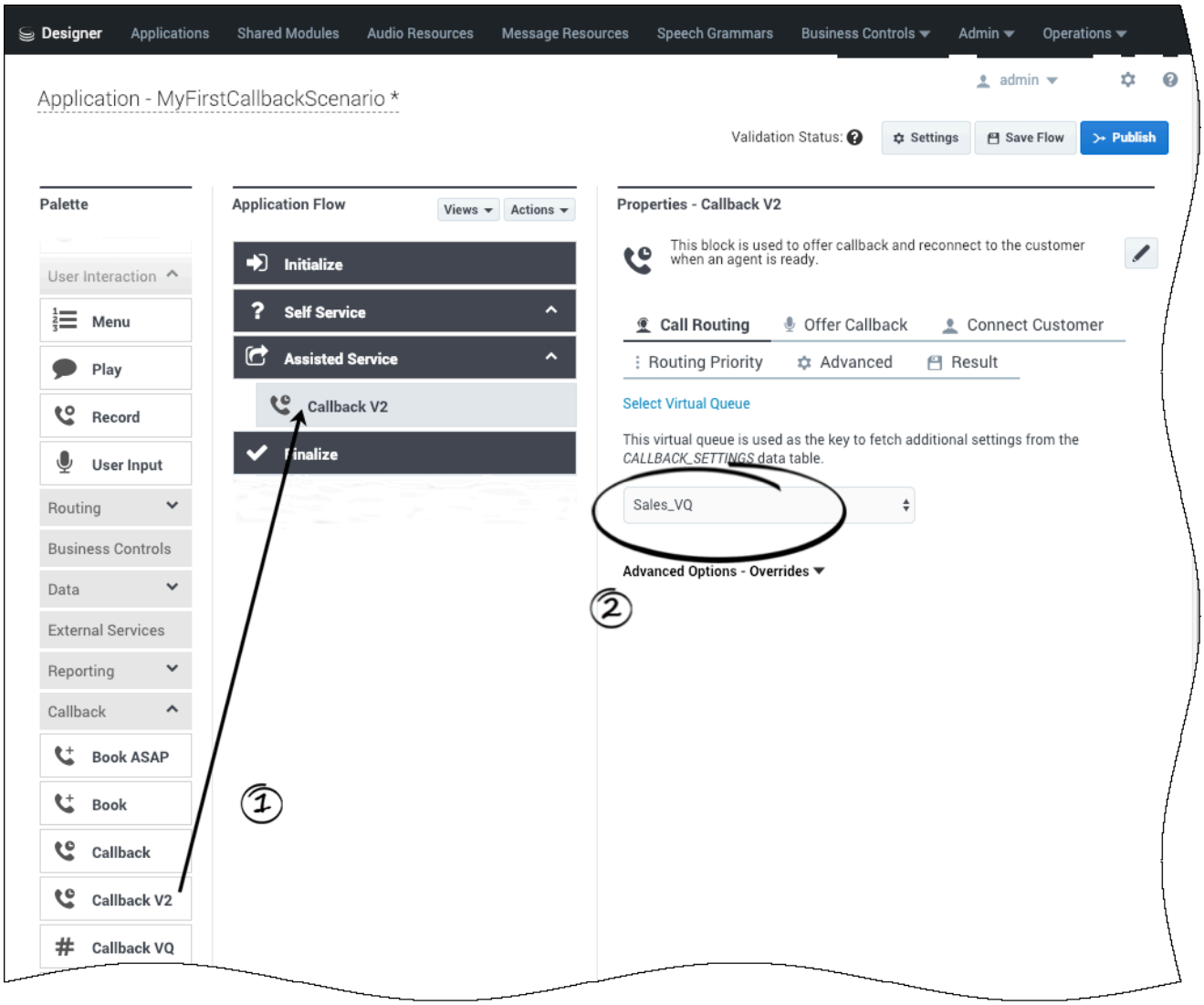


In Designer, add a new application. The application type must be `Default`. This application is used to offer callback through the IVR.

> **Important**
>
> If redirecting a caller to a Designer application that contains IVR callback, only **1-step transfers** are supported.



There are no mandatory settings changes for the Default application, however, if there are any specific settings that you typically use for Designer applications, consider if those settings are required for your Callback Default-type application and make any necessary updates.

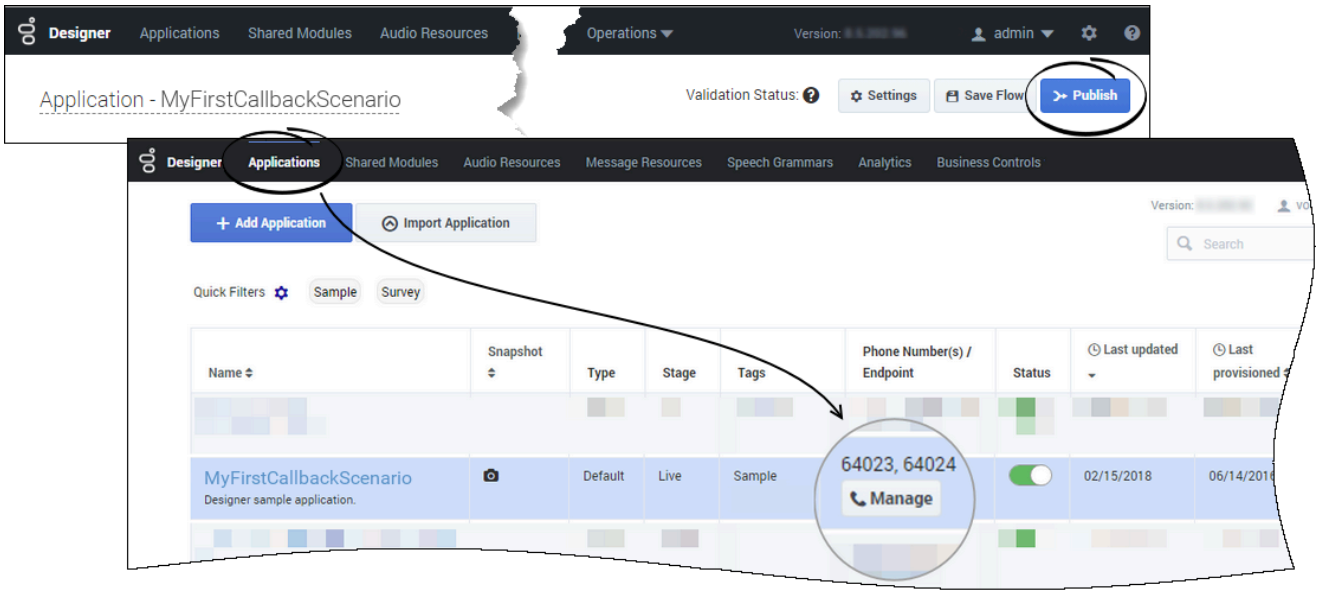


In Designer, select your Default-type application.

Scroll the **Palette** list to reach the **Callback** items. Drag and drop a **Callback V2** block into the **Assisted Service** phase of your application.

In the properties panel, under **Call Routing**, select the Inbound virtual queue that you configured for Callback.

For additional information about the Callback V2 block properties, see the Designer documentation.

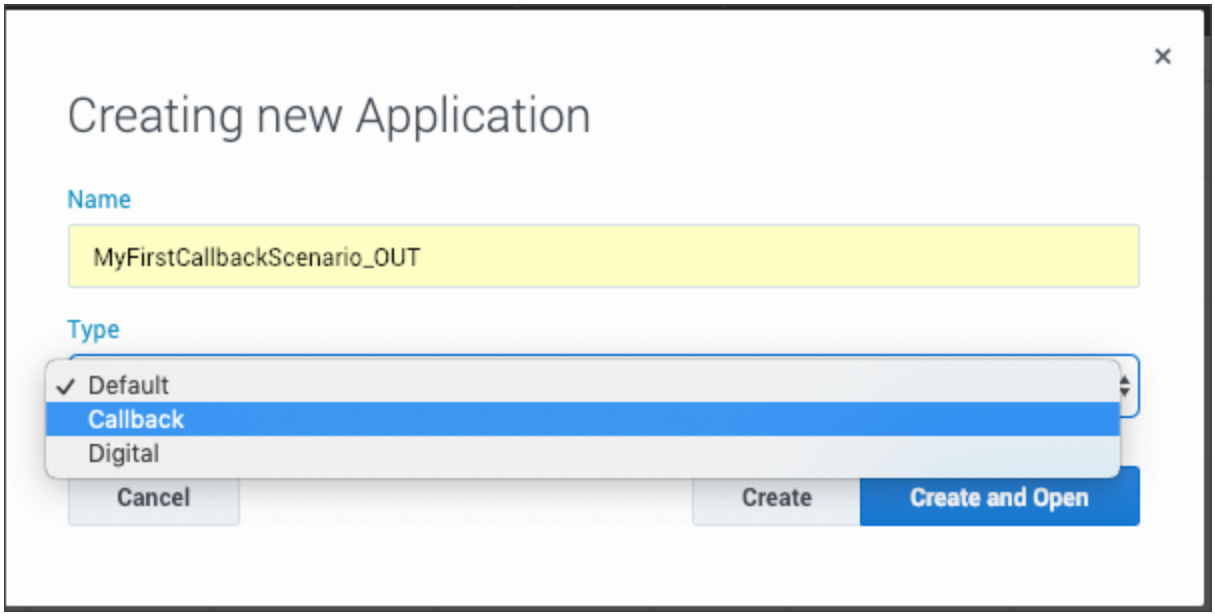


When you complete configuration of the application, you must save and publish it. Remember to assign a phone number to it.

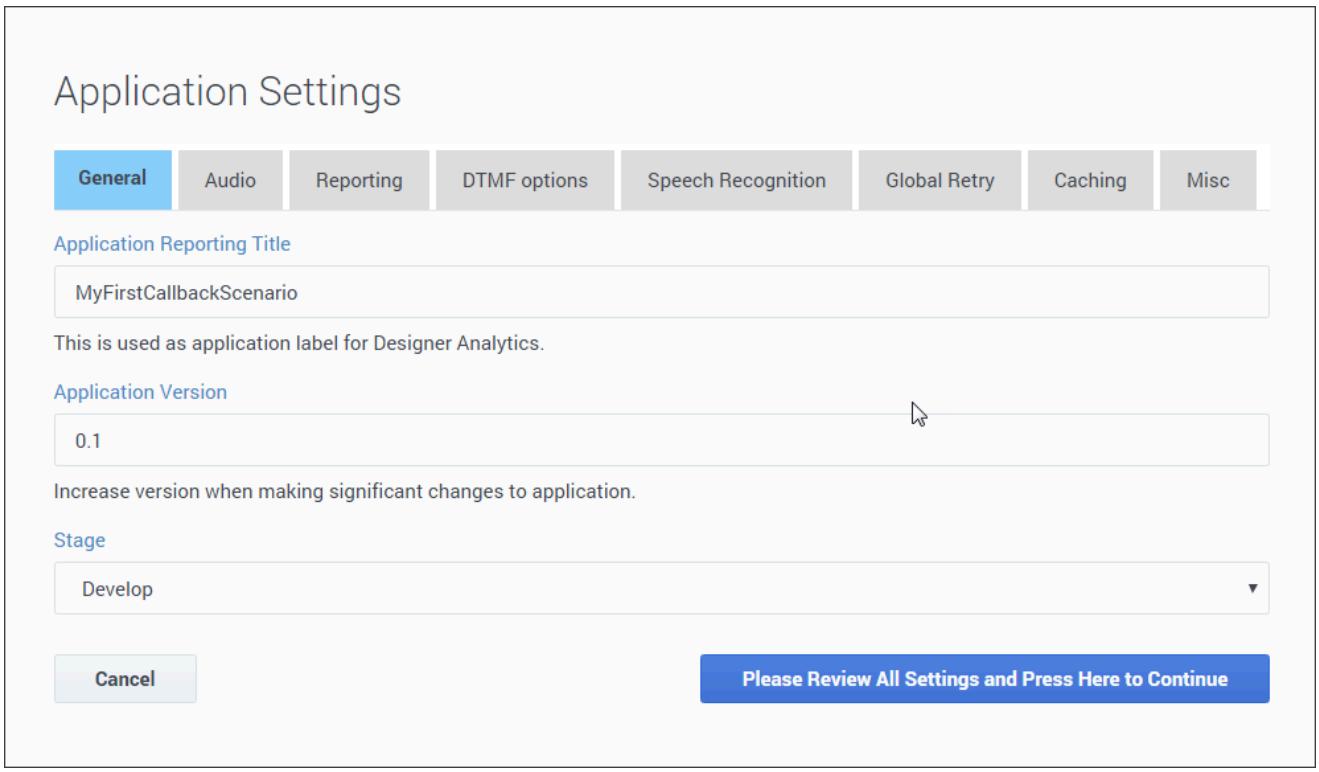
Provision the Designer callback application

This application type is used for setting up outbound callbacks (voice calls only). Digital interactions are not supported.

> **Important**
> You must create the Callback-type Designer application before you can edit the CALLBACK_SETTINGS data table.

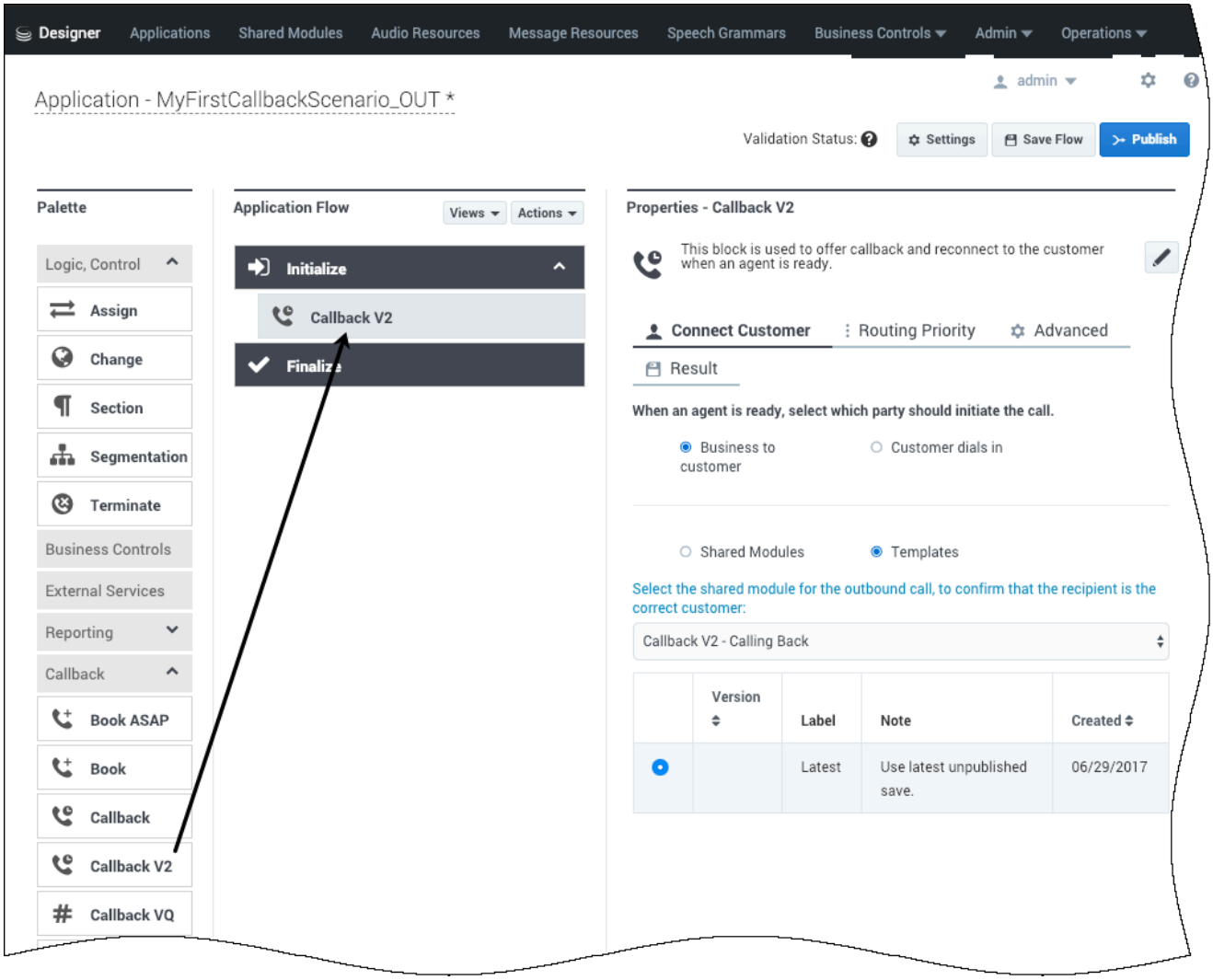


In Designer, add a new application. The application type must be `Callback`. This application is used to re-connect with customers who requested a callback.



There are no mandatory settings changes for the Callback-type application, however, if there are any specific settings that you typically use for Designer

applications, consider if those settings are required for your Callback application and make any necessary updates.



In Designer, select your Callback-type application.

Scroll the **Palette** list to reach the **Callback** items. Then drag and drop a **Callback V2** item into the **Initialize** section of your application. For information about the Callback V2 block properties, see the Designer documentation.

> **Important**
>
> In the Callback V2 properties panel, under **Connect Customer**, only the `Business to customer` option is currently supported.

When you complete configuration of the application, you must save and publish it. You click **Publish** to make your application available for use. For information about publishing a Designer application, see Saving and Publishing Your Application.

Provision business hours for Callback

In Designer, you must configure the Business Hours object, including the timezone, before you configure the CALLBACK_SETTINGS data table. The time zone that you configure is used for scheduled callbacks. You cannot save the CALLBACK_SETTINGS data table before the business hours are configured.

For information about configuring your business hours in Designer, see Business Hours in the Designer documentation.

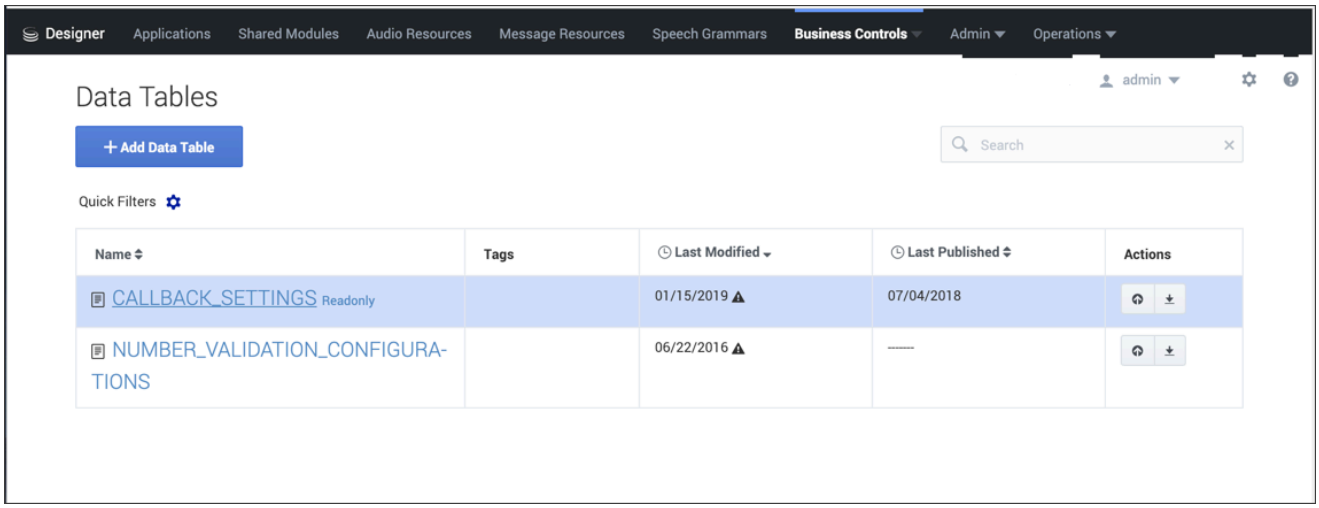
Provision the callback services

The configuration parameters for callback services are stored in the Designer CALLBACK_SETTINGS data table. For detailed information about the CALLBACK_SETTINGS table, see Callback Settings Data Table in the Designer documentation.

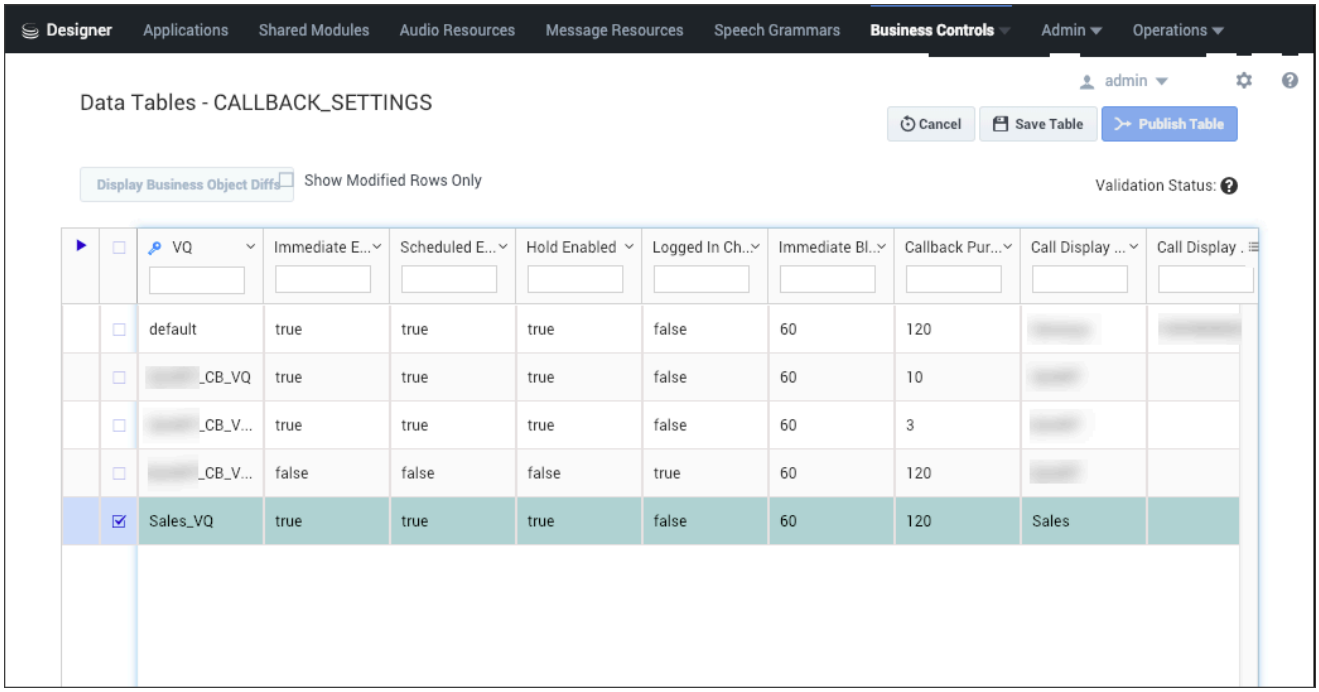
You must make sure that the following prerequisites are completed before you add your queue to the CALLBACK_SETTINGS table:

- You must create the Callback-type Designer application before you can edit the CALLBACK_SETTINGS data table.
- You must configure the business hours, including the time zone, before making the following updates to the CALLBACK_SETTINGS data table. You cannot save the data table before the business hours are configured.
- The virtual queues that you will use for callback functionality must be created and saved in Platform Administration.

Adding a Callback virtual queue to the CALLBACK_SETTINGS data table



In Designer, navigate to **Business Controls** > **Data Tables**. Click **CALLBACK_SETTINGS** and add an entry for your Callback Inbound queue as described in the Callback Settings Data Table documentation.



The `default` VQ row in the CALLBACK_SETTINGS table must include a valid Designer Callback-type application, a valid Business Hours object, and a semantically-correct skill expression.

In the row in which you are configuring the Inbound virtual queue, select your Designer Callback-type application in the **Callback Application** column of the table. In the figure, this is the `Sales_VQ` row of the table.

If you defined Callback skills for Callback agents, you can use this as a condition for the Inbound virtual queue.

For Callback-type applications, you must specify the routing point in the **Routing Point** column. The routing point for Callback-type applications is required for Callback to be fully functional. Outbound calls can sometimes fail if this is not configured.

Testing your callback scenario

Now you can test the in-queue Callback scenario. Call the phone number that is assigned to the Designer Default-type application and accept an in-queue callback for an external phone number on which you can receive calls. If an agent with a matching skill is already logged in to the voice channel in Agent Workspace, you will receive a phone call on the external phone number. Once you have accepted the callback on the external phone, the call will be connected to the logged-in agent.

To test Scheduled callback, call the phone number that is assigned to the Default-type application and choose `Scheduled Callback`. Listen carefully to the prompt. If it is using default settings, then it will ask you to specify a day and time for the callback based on the Pacific time zone. Also pay close attention to the actual date and time that you booked before accepting, especially if you have entered a time intended to be on the current day; the system might have offered you a time slot that

is a week later. The outbound call experience is identical to the in-queue callback scenario.

To test web callback, see Managing Callbacks for information about creating a callback in the Callback UI, or Genesys Multicloud CX REST APIs and Tutorials for Callback for information about creating a callback using the REST API.

Controlling user access

Contents

- 1 Migration of Roles
- 2 Accessing the Callback UI
  - 2.1 Predefined Access Groups
- 3 Line of Business segmentation

-
- Administrator

You can limit user access within the Callback UI to what is appropriate for each user's role. This article describes the predefined Access Groups and Callback Roles to which you can assign users and includes information about how to restrict user access to callback information based on your lines of business. If you are migrating Roles and Access Groups from an earlier version of Callback, be sure to read the note related to this activity.

Related documentation:

-

If you do not configure access and activity restrictions for a user, then that user has full access to everything in the Callback UI. Genesys provides predefined Callback Roles that administrators assign to users. Each Role includes permissions that control what users can see and do in the UI. For example, to perform their duties, it is sufficient for some users to view the list of callback records in the Callback UI, without the ability to modify a callback in any way. Other users might require additional permissions; for example, some users might require access permissions that allow them to create, cancel, and modify callbacks and to view errors associated with callbacks.

Review Platform administration for detailed information about which applications and documentation to use to create or modify Roles and Access Groups or to assign users to these groups.

Migration of Roles

If you have not yet moved to the Roles and Access Group settings described on this page, the original method for configuring Callback access still applies to your setup. In the original configuration, Callback access was granted when the **ges** or **gms** section in the Person object's annex included a `role` option (for example, Administrator).

To move to the new configuration method for granting access, add your user to the correct Callback or custom Access Group and remove the **ges** or **gms** section. If you don't remove the **ges** or **gms** section, the old configuration applies and the Access Group is not taken into consideration.

Accessing the Callback UI

Genesys provides the following default Callback Roles:

- **Callback Administrator**—Callback Administrators have full access to the Callback UI, which includes the ability to create, cancel, and reschedule callbacks, and to export reports. Users with this Role can also access all of the **Developer** tab features.

- **Callback Supervisor**—Supervisor users have full access to the Callback tab in the UI, which includes the ability to create, cancel, and reschedule callbacks, and to export reports. They cannot access the **Developer** tab in the UI.
- **Callback Monitor**—Monitor users can only view callbacks.
- **Callback Developer**—Developer users can view callbacks on the **Callback** tab and have full access to the **Developer** tab, which includes the ability to view recent errors, test API keys, and to provision Callback.

If a user is a member of more than one Role, the Role that allows the most access to Callback features takes precedence.

In addition to Roles, a user must belong to at least one of the following Access Groups to have access to the Callback UI:

- Administrators*
- Supervisors*
- Managers*
- Agents*

Predefined Access Groups

By default, Genesys defines a list of Access Groups and adds Callback Roles to some of these groups, as described below. Users who are already in these Access Groups are given Callback permissions by default. For example, any user in your Administrators Access Group is automatically granted the Callback Administrator Role.

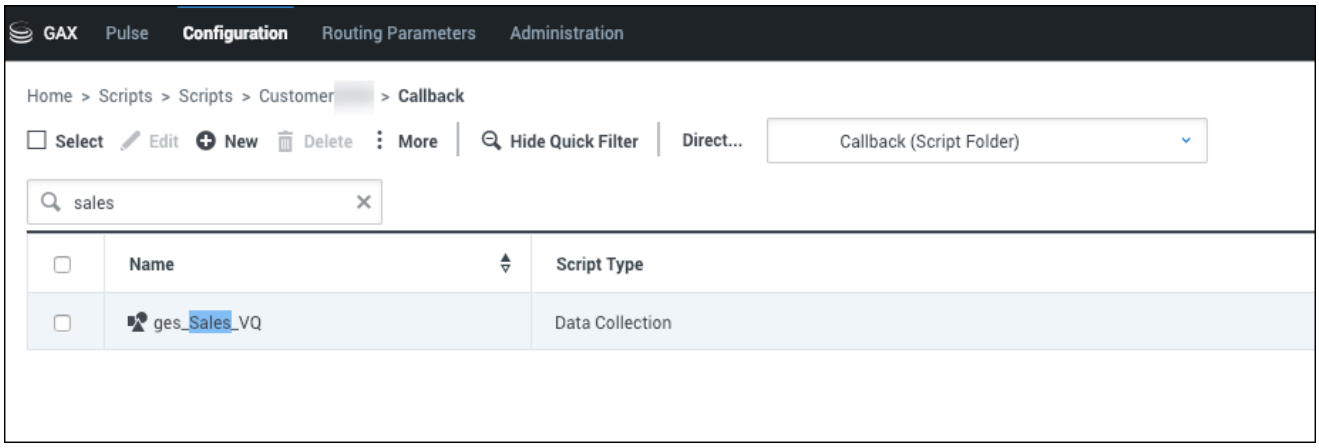
Access Group	Callback Administrator Role	Callback Supervisor Role	Callback Monitor Role	Callback Developer Role
Administrators*	✓			
Supervisors*		✓		
Managers*			✓	
Callback Developers				✓

***IMPORTANT:** The Access Group name is prefaced with your company's business name if the Access Group is not Callback-specific. For example, if your business name is ACME, then the Access Group for Administrators is called "ACME Administrators".

Line of Business segmentation

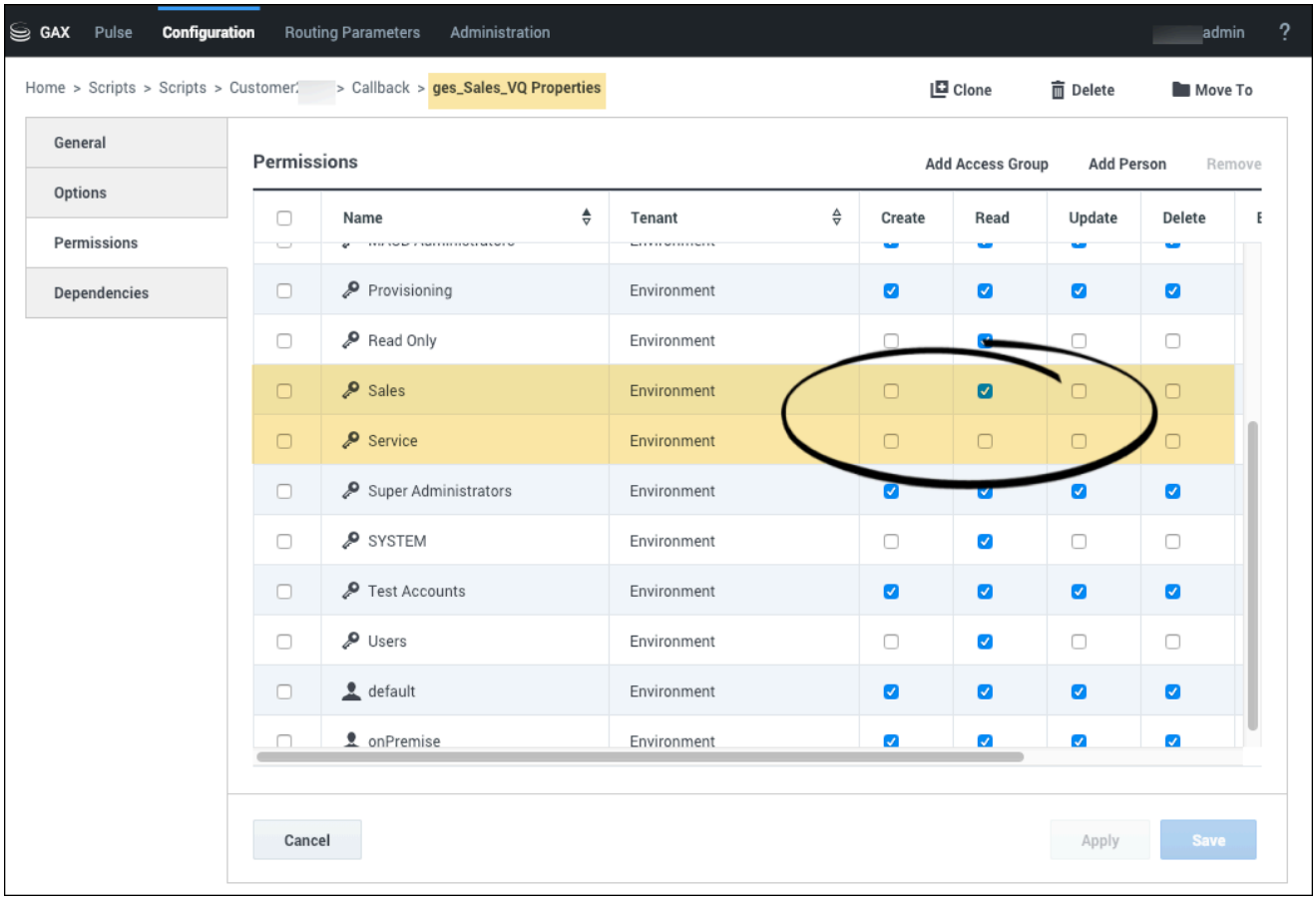
Limited Availability	The following procedures describe how to segment users based on lines of business. These procedures are applicable only to Genesys Multicloud CX deployments on Amazon Web Services (AWS).

By default, all users who are part of standard Access Groups and can access the Callback UI will have Read permission for all of the virtual queues. To restrict access to queues based on your lines of business, you can create custom Access Groups and enable or disable Read permissions as required.



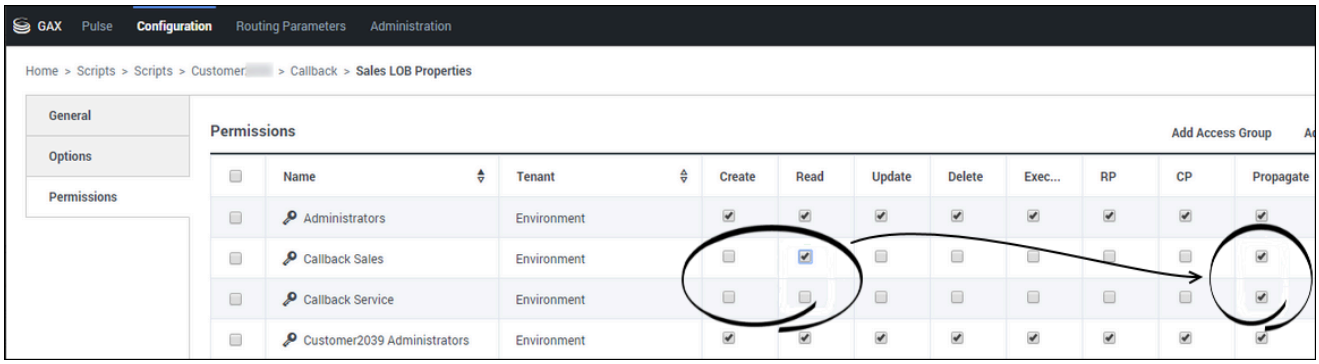
Any time you provision a virtual queue in Designer's CALLBACK_SETTINGS data table, the Callback service (virtual queue) automatically creates a Script object in **Platform Administration** > **Scripts** > **Callback**. The created Script object has the same name as the virtual queue and is prefixed with the `ges_` label. For example, if you create a virtual queue called `Sales_VQ`, there will be a Script object called `ges_Sales_VQ` in the Callback directory.

To control access to queues based on your lines of business, you must create Access Groups for your various lines of business and then enable or disable access to the script objects that represent the virtual queues for each group. For a user to access a specific queue, the Access Group to which the user belongs must have the Read permission for the script object that represents that queue. The Read permission is assigned by default to all Access Groups, which means that all Access Groups can access all virtual queues until you change the permissions. To deny access to a virtual queue, navigate to the Script object associated with the queue and remove the Read permission from Access Groups that do not require access to that queue.



For example, if your Tenant has two lines of business called **Sales** and **Service**, you could create two Access Groups for Callback: **Sales** and **Service**. Then, navigate to each script object representing these queues and add the Access Group with Read permission:

- In the **ges_Sales_VQ** Script object, retain the Read access for the **Sales** team and disable the Read permission for the **Service** team.
- In the **ges_Service_VQ** Script object, retain the Read access for the **Service** team and remove the Read permission from the **Sales** team.



To set permissions on groups of virtual queues (instead of one at a time), create subfolders under the **Scripts** > **Callback** folder and apply appropriate permissions to the subfolder. Then, move the Script objects representing the various virtual queues into the corresponding subfolder. Any Script object that is in a subfolder will inherit the permissions of that subfolder.

Check the **Propagate** box to apply the permissions to any object that is in the folder. The permissions apply to any virtual queue that is in the subfolder now and will apply to any new virtual queue that you add to the subfolder in the future.

Genesys Multicloud CX REST APIs and tutorials for Callback

Contents

- 1 Accessing Callback APIs
- 2 Using the APIs to book a callback

-
- Administrator

Depending on your role, you might work with the APIs that are available for Genesys Engagement Services (GES)/Callback. This article provides links to documentation and tutorials for the GES REST APIs and tells you what is required to access Callback APIs from web or mobile applications.

Related documentation:

-

The documentation for REST APIs is available through the Genesys Multicloud CX Developer Center.

The GES/Callback APIs are:

- Callbacks — Create, retrieve, cancel callbacks.
- Estimated Wait Time —Retrieve Estimated Wait Time.
- Availability — Retrieve time slots for a callback, matching Office Hours.
- Call In — Request the phone number to call in.
- Queue Status — Retrieve information about a queue's readiness to accept callbacks.
- Statistics — Provides a proxy to the GWS Statistics API. To use the Callback Statistics API, you must first register your GWS credentials in the Callback UI (**Developer** > **Credential Management** > **GWS Credentials** tab).

You can test queries in the Developer Center. If you receive error messages while testing your application, use the **Developer** tab to get more information.

The portal also includes Callback tutorials; select **Engagement** in the tutorial list.

Accessing Callback APIs

To access Callback APIs from Web or Mobile applications, the following items are mandatory:

1. Request an API key from your Genesys Customer Care Representative. You need to send this as the value for `x-api-key` in the header of every request.
2. In your request to Customer Care, specify the list of domains from which you expect your API requests will originate. Genesys uses this list to configure cross-origin resource sharing (CORS) domains for you (this is not currently available as self-service configuration).
3. The base URL for your API endpoints is:
   `https://gapi-.genesyscloud.com/engagement/v3`

   You will receive the information from Genesys at the same time that you receive your API key.

Consult the Genesys Multicloud CX Developer Center for complete endpoint paths.

Using the APIs to book a callback

Before you book a callback using the APIs, read Booking a callback using the callbacks create API for information about callback settings that you must check manually *before* you create the callback.

For information about all of the settings relevant for callbacks, see the CALLBACK_SETTINGS data table documentation in the *Designer User's Guide*.

Managing callbacks

Contents

- 1 Callback states
- 2 Callback administration overview
- 3 Searching for a specific interaction
- 4 Metrics
- 5 Advanced options
- 6 Create or schedule a callback
- 7 Edit a callback
- 8 Recreate a callback
- 9 Bulk cancel and export of callback records

-
- Administrator

The **Callback** tab in the Callback UI displays the list of callback records. If you have sufficient user privileges, you can create, recreate, schedule, edit or cancel callbacks on this tab. Use the **Advanced Options di**alog to customize the display of data.

Related documentation:

-

Callback Monitor is the minimum role required to access the **Callback** tab.

The **Callback** tab in the Callback application displays the list of callbacks in pages of 250 items. Depending on your time filter selection, you might continue to see a callback record on the **Callback** tab after the **Desired Callback Time has passed**.

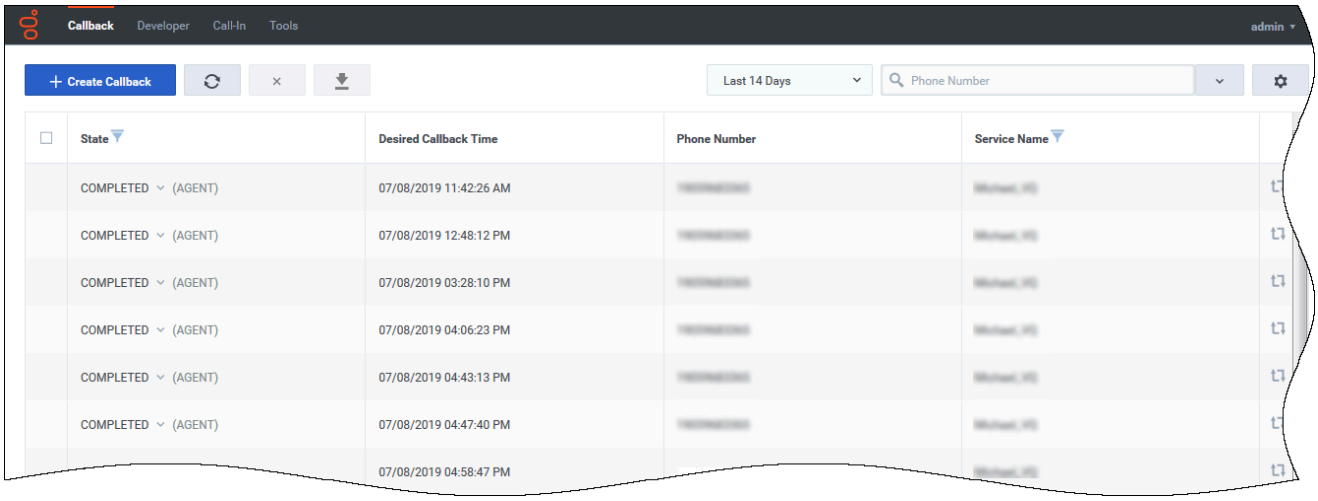
Additional tabs and dialogs might display, depending on the permissions associated with your Role.

You might see the term *service* or *service name* in callback-related applications, Widgets, APIs, and in the Callback UI. A service, in this context, is a virtual queue. The service name, therefore, refers to the name of the virtual queue. For example, on the **Callback** page in the UI, the **Service Name** column identifies, by name, the virtual queue associated with each callback.

> **Important**
> Genesys Callback supports voice calls only. Digital interactions are not supported.

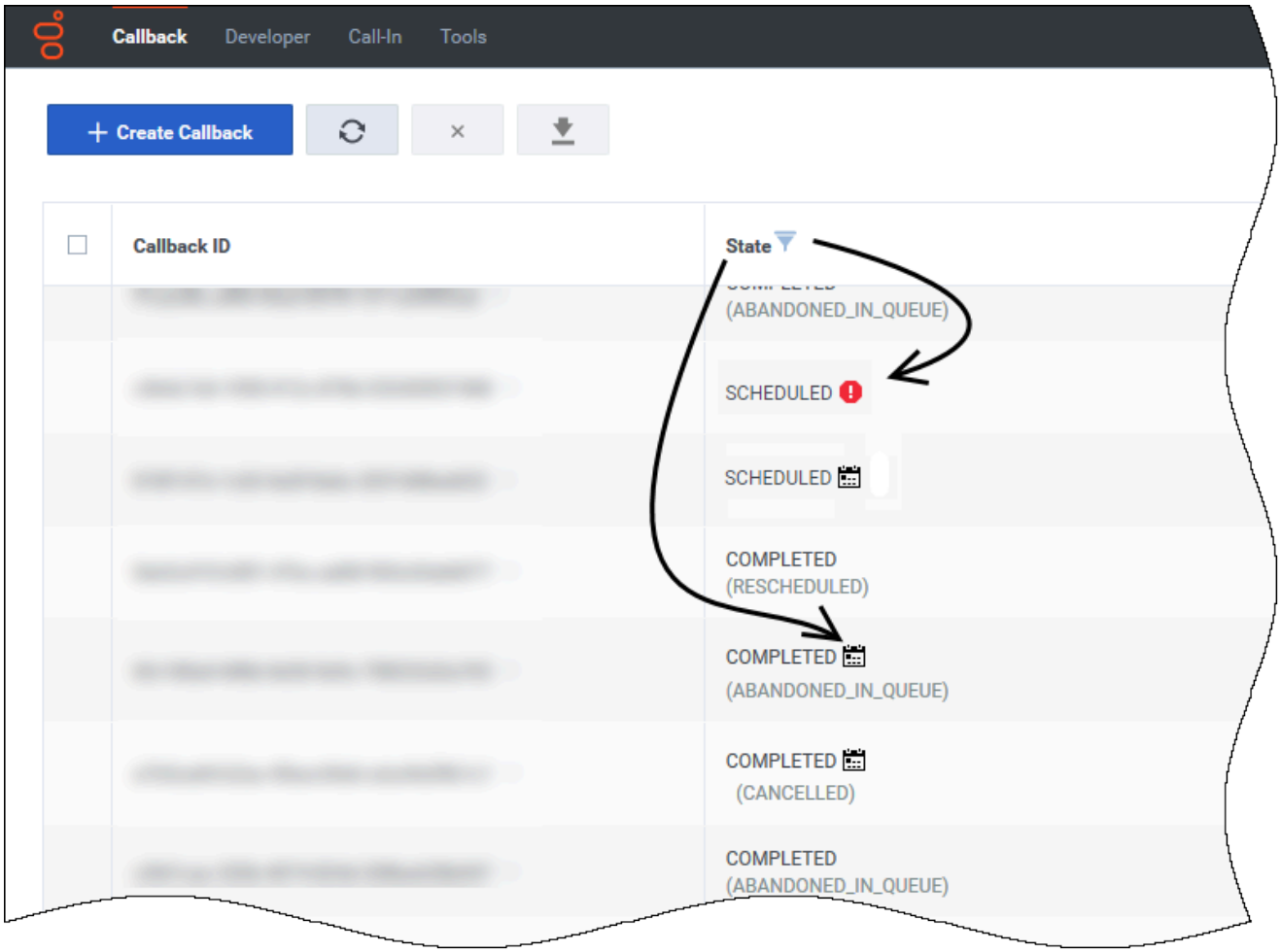
> **Important**
> Callback records are stored for 14 days. The 14-day TTL setting starts at the Desired Callback Time. The `Callback TTL (seconds)` setting in the CALLBACK_SETTINGS data table has no effect on callback record storage duration; 14 days is a fixed value for all callback records.

Callback states



Callback records can have the following states:

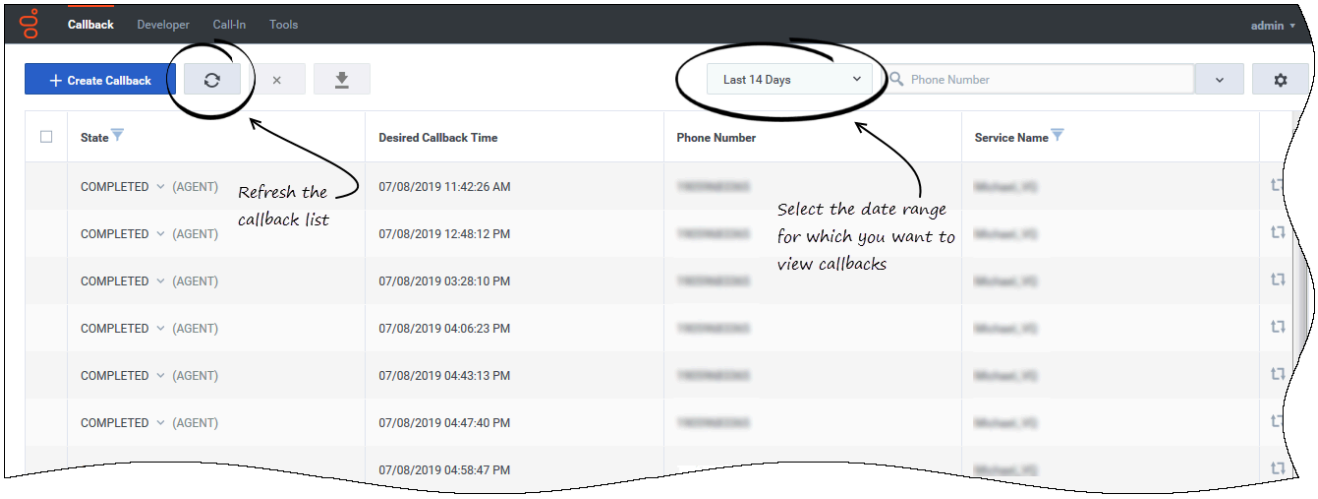
- SCHEDULED—The callback-management system is handling the request (there are no sessions started in the system). While in this state, the request is available on the **Callback** tab as long as the specified `Desired Callback Time` is upcoming.
- QUEUED—Callbacks actively waiting for an agent in ORS/URS; the agent is not assigned yet.
- ROUTING—An agent is reserved but the call is not yet routed to the agent.
- PROCESSING—The assigned agent is handling the callback. You cannot cancel a callback when it is in the PROCESSING state.
- COMPLETED—The callback was completed. The COMPLETED status can include a result or reason for the callback completion. For example, a result can indicate that the callback was completed by an agent or was cancelled by an administrator. The result displays beside COMPLETED in the **State** column of the table.



The **Desired Callback Time** is not a precise moment in time, but rather a short time increment – by default, two minutes – that begins at the time identified in the column. When a callback in the SCHEDULED state remains in that state past the **Desired Callback Time** time interval, a warning icon displays to alert you. Hovering your cursor over the icon displays the warning message.

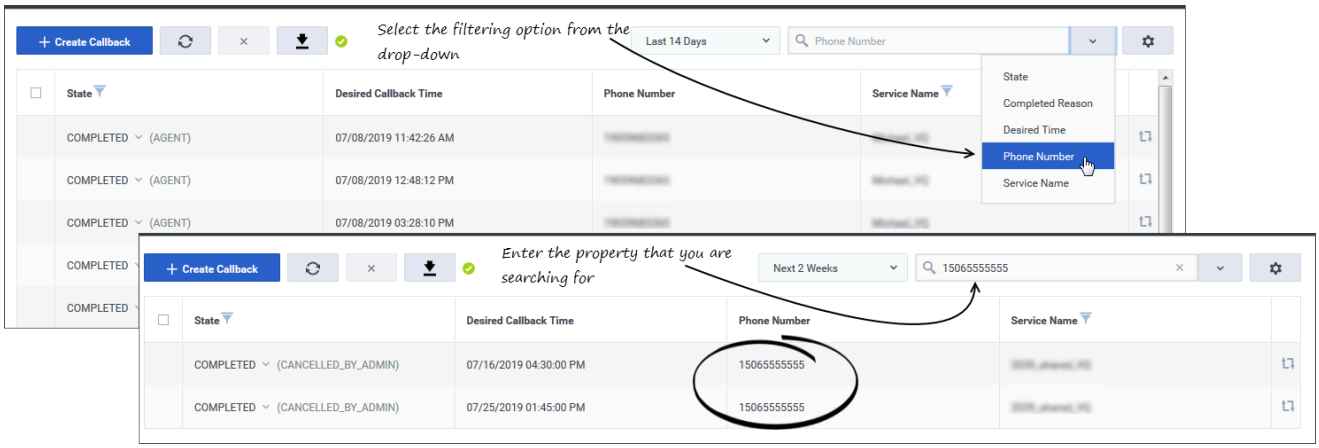
A calendar icon displays beside callbacks that originated as scheduled callbacks. If there is no icon associated with a callback's state, then the callback was initiated as an Immediate callback.

Callback administration overview

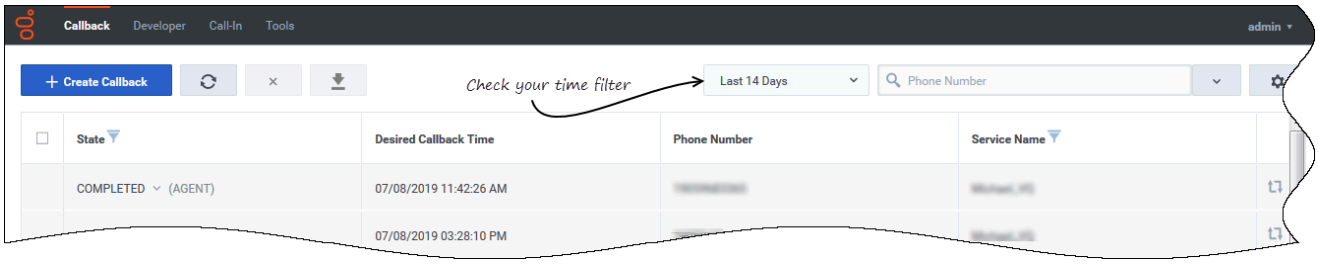


The **Callback** tab includes the following features:

- Refresh—You can force the interface to refresh the list of callbacks.
- Date range selector—The date range drop-down menu lets you filter the list of callback records based on a specific (pre-defined) period of time. In the Advanced Options window, you can add your own custom date range to the drop-down menu.

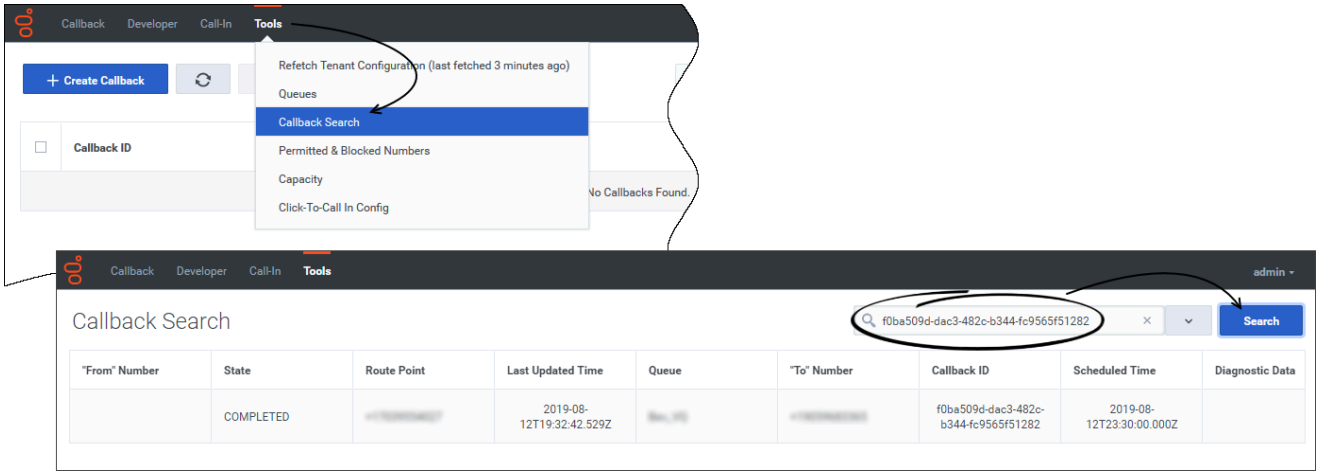


- Search box—You can select a column field in the drop-down at the right of the Search box. The interface then filters the results dynamically as you type.



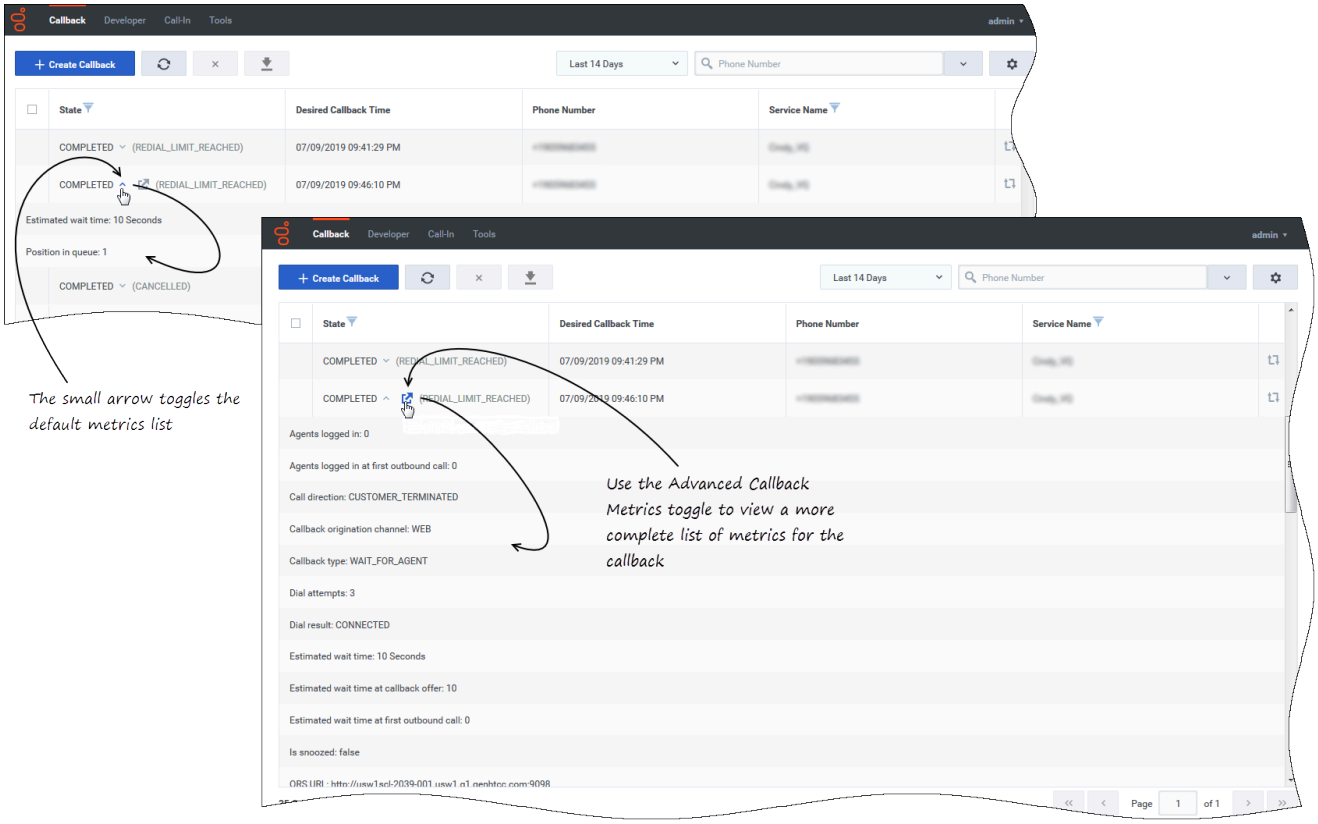
If you do not see your callback after you successfully create it, make sure that your time filters are correct. For example, if you scheduled a callback to occur in two days, it does not show up if the **Last 30 Days** label is selected. Switch to **Next 7 Days**, or something similar, to view it.

Searching for a specific interaction



To quickly find a specific callback interaction, use the **Tools** > **Callback Search** menu option. To search for an interaction, enter either the callback ID or the customer's phone number and click **Search**.

Metrics

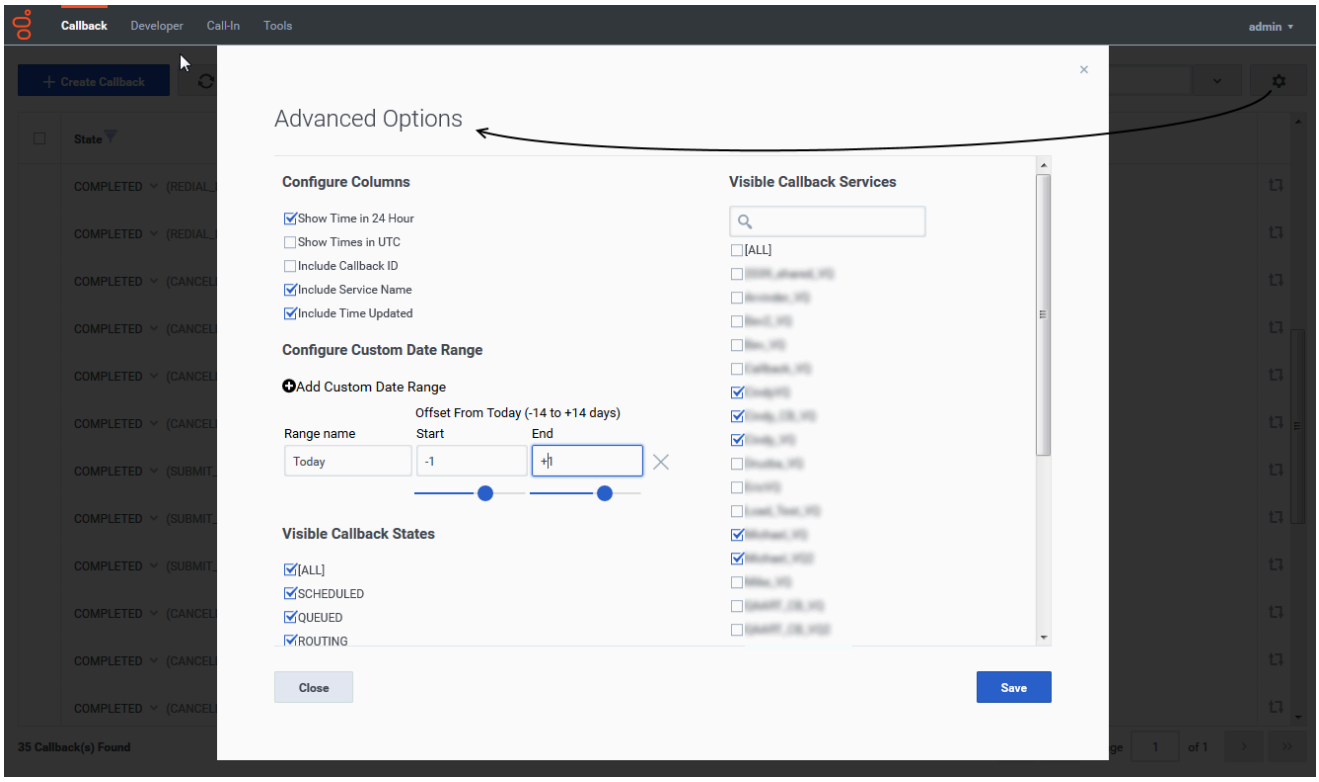


On the **Callback** tab, you have access to diagnostic data or *metrics* for each callback record. The metrics that are available for a callback depend on the callback's state. As the state of the callback progresses, more data becomes available, which means that additional metrics become available as well.

To view the metrics for a callback, click the arrow beside the record. Callback provides some basic information below the callback record (Position in Queue and Estimated Wait Time).
To view all metrics for the callback, click the Advanced Callback Metrics toggle. For example, you might want to know who cancelled a particular callback. That information is available in the list of advanced metrics.

> **Tip**
> The username of the person who created, cancelled, rescheduled, or recreated a callback is captured only when the action is performed in the Callback UI.

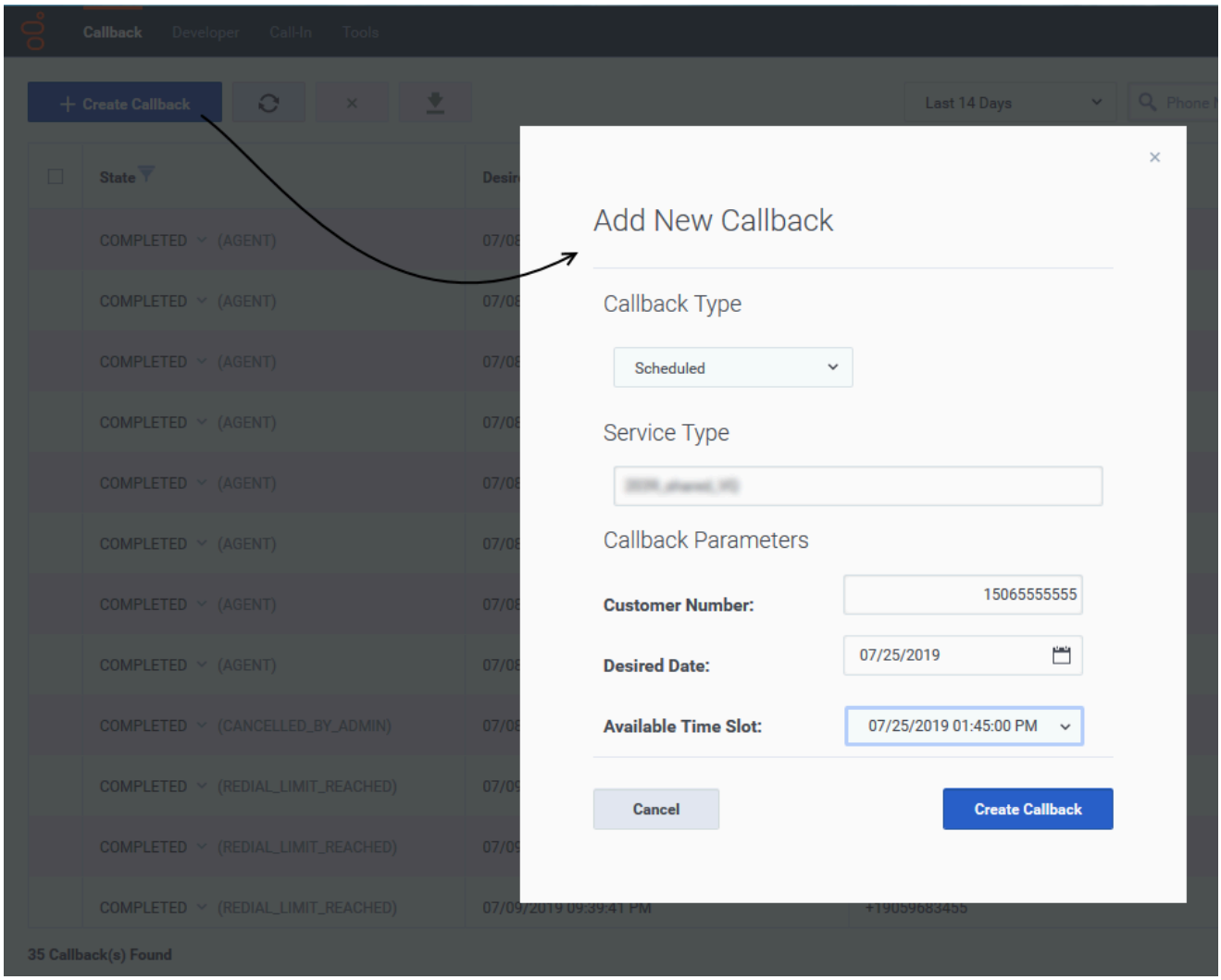
Advanced options



Advanced Options—Opens a dialog in which you can customize how the **Callback** tab displays data:

- **Configure Columns**—Select display properties for columns.
  - **Show Time in 24 Hour**—Show all times in 24-hour time rather than AM/PM format.
  - **Show Times in UTC**—Show all times in the table in UTC time rather than local time.
- **Configure Custom Date Range**—Add a time range filter.
- Filter the display of callbacks by **States** and **Services**—To filter the list of callbacks that display on the **Callback** tab, you can select specific callback states and virtual queues (services). To view callbacks in any state, select ALL in the **Visible Callback States** list. Similarly, to view callbacks from every virtual queue, select ALL in the **Visible Callback Services** list.

Create or schedule a callback

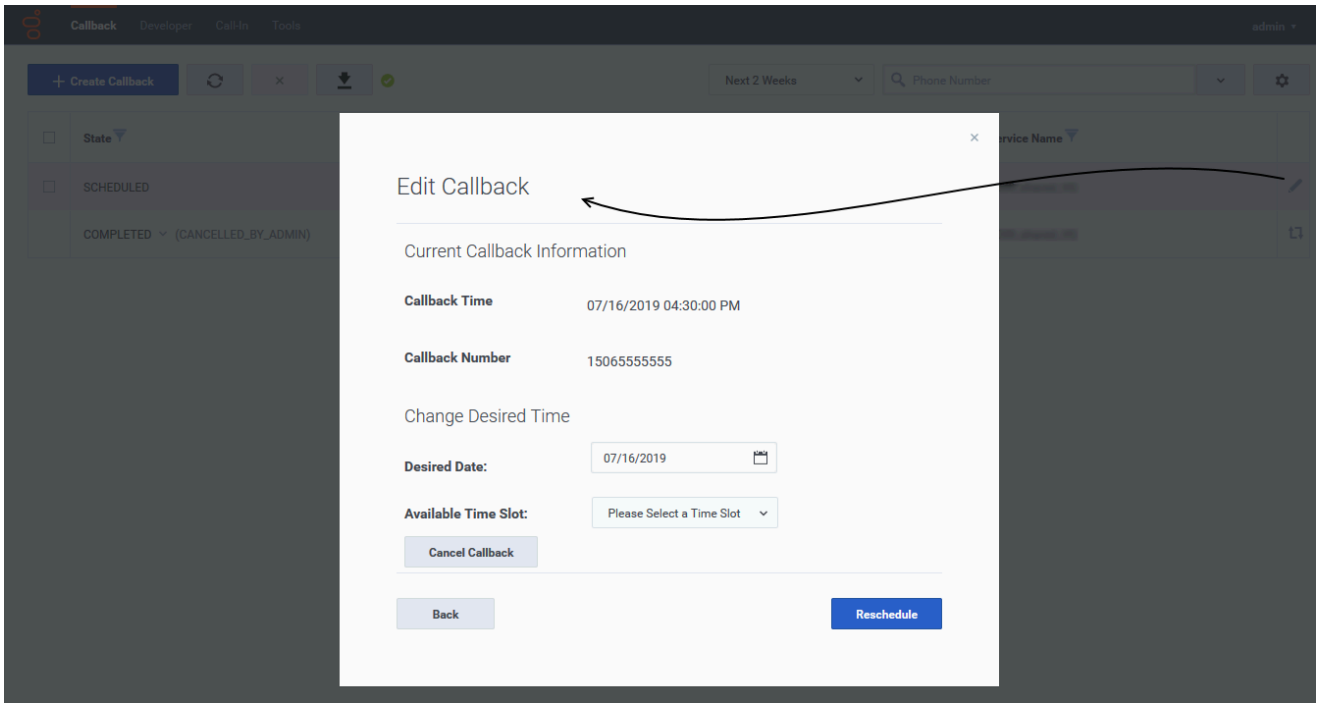


> **Important**
> This feature requires the **Callback Administrator** or **Callback Supervisor** role.

Click **Create Callback to** submit or schedule a callback request. A new dialog opens. Select the **Callback Type** (Immediate or Scheduled), then a **Service Type** (the name of the virtual queue provisioned in Designer), and enter the number to call in the **Customer Number** field. For Scheduled callbacks, select a date and time for the call.

> **Tip**
> If you provision a virtual queue for callbacks in Designer while you are logged in to the Callback UI, be aware that the new queue will not be available for use in the Callback UI until you log out and log in again because Callback fetches virtual queue permissions data only at login. In other words, Callback determines which virtual queues you have access to when you are logging in, and only then.

Edit a callback

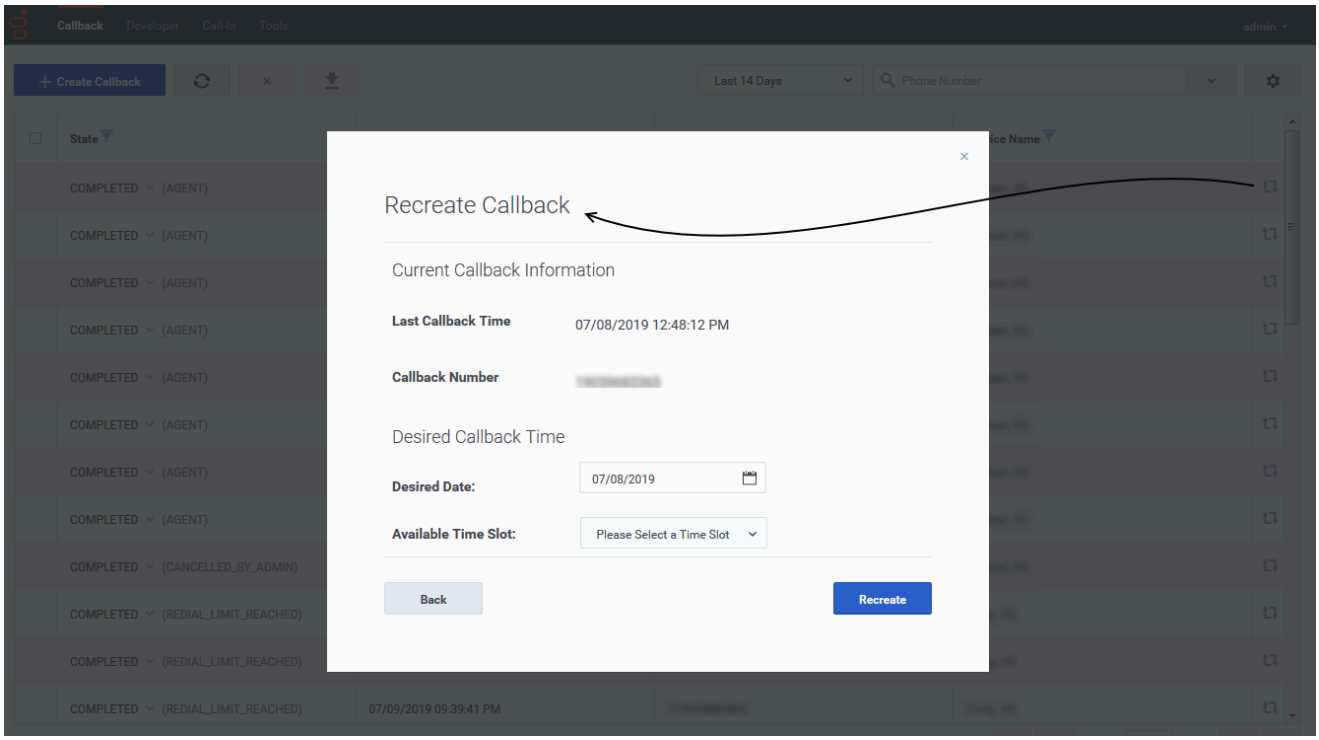


> **Important**
> This feature requires the **Callback Administrator** or **Callback Supervisor** role.

In the Callback table, for Callbacks that can be edited, a blue pencil displays in the last column. Clicking this pencil displays the edit options for that Callback.

- You can choose to reschedule the Callback (only for SCHEDULED Callbacks).
- You can choose to cancel the Callback.

Recreate a callback

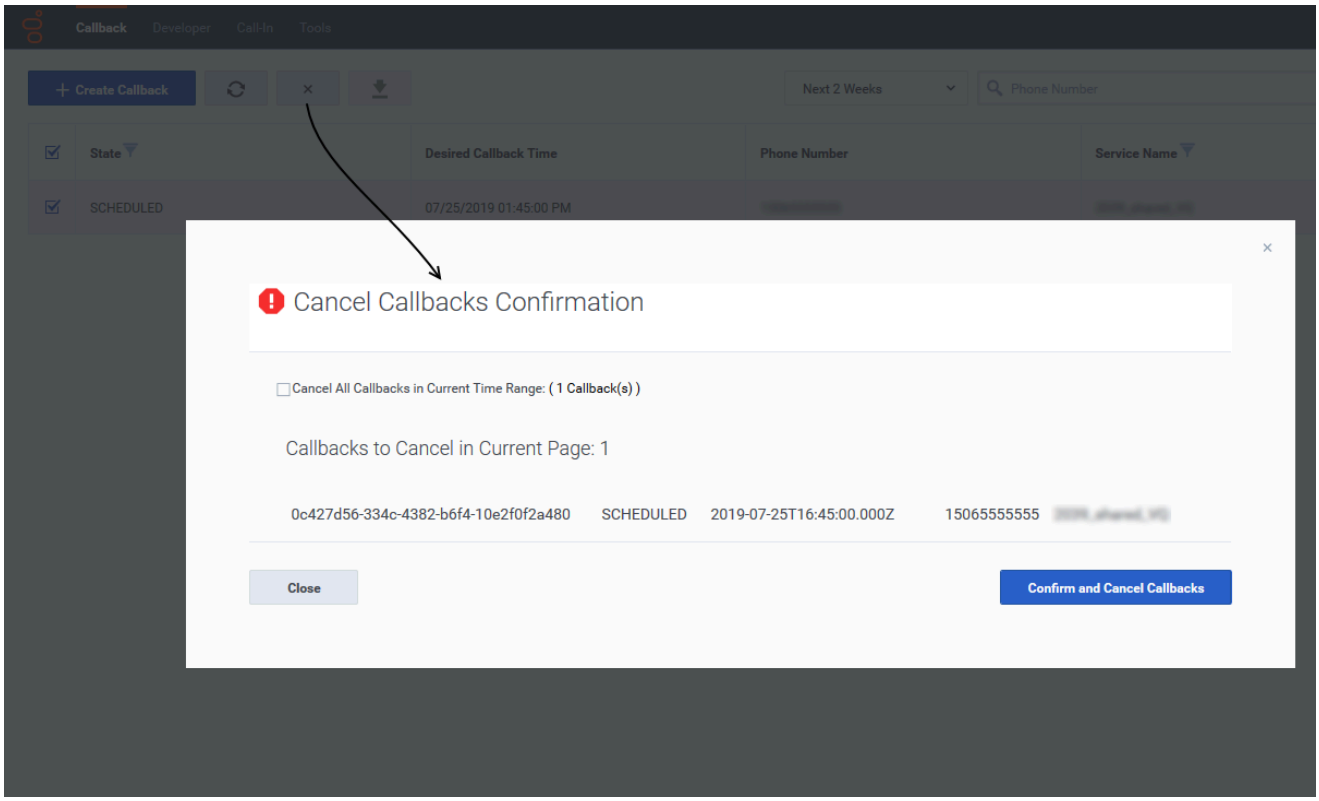


> **Important**
> This feature requires the **Callback Administrator** or **Callback Supervisor** role.

For completed callbacks which can be recreated, the icon displays in the last column of the table. Click the icon to open the **Recreate Callback** dialog for that callback.

> **Important**
> Name recordings will be preserved if a callback is rescheduled within four days of the completion of the original callback.

Bulk cancel and export of callback records



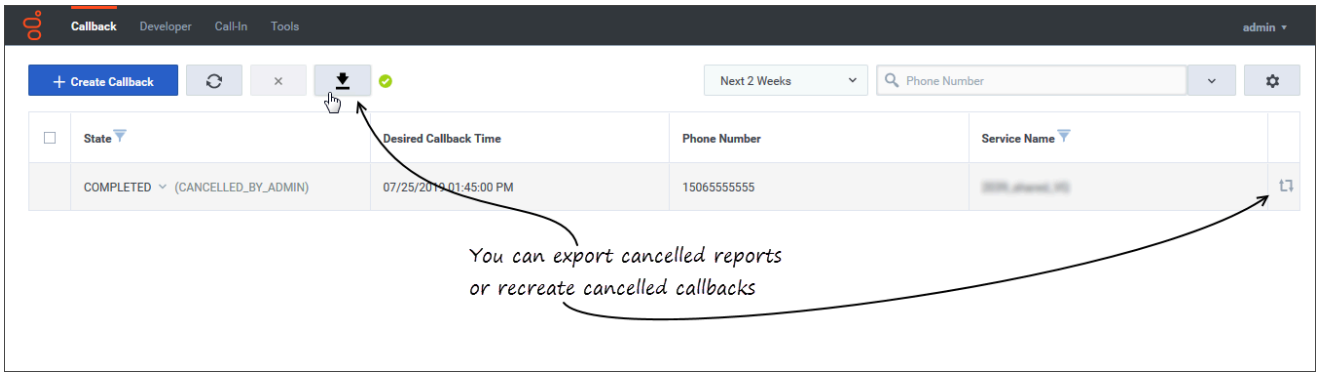
> **Important**
> This feature requires the **Callback Administrator** or **Callback Supervisor** role.

Make sure to select an appropriate Time Range to filter callbacks, then make a callback selection to activate the **Cancel Callbacks** button. When you click this button, the **Cancel Callbacks Confirmation** dialog opens, displaying the selected callback cancellations.

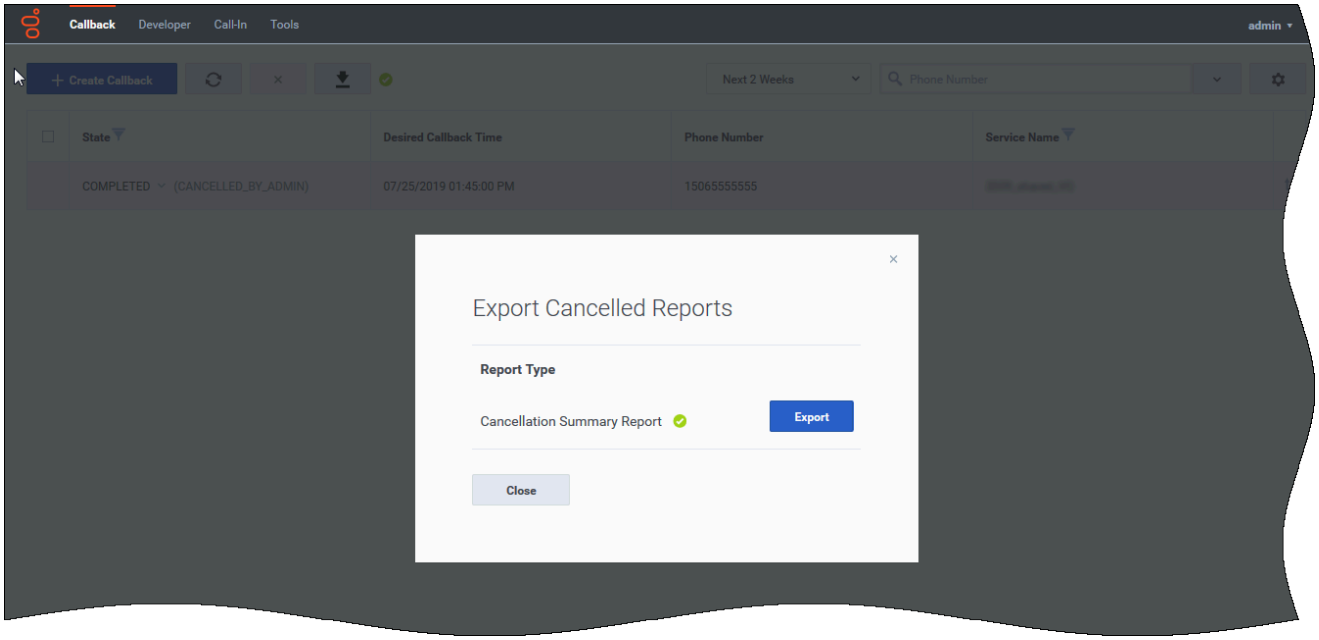
You can either:

- Continue with the selections you have made and click **Confirm** to cancel them, or
- Check **Cancel All Callbacks In Current Time Range** to override your previous callback selections and delete all callbacks in the current time range. If you select this option, the list of displayed callbacks disappears from the dialog. Callbacks that are in the PROCESSING state cannot be cancelled. If there are any callbacks in this state within the selected time range, they are excluded in the **Cancel Callbacks Confirmation** dialog box and are not cancelled.

Click **Confirm** to cancel the records.



The resulting display shows all the callbacks that you have cancelled and that are now in the COMPLETED (CANCELLED_BY_ADMIN) state.



The **Export Cancelled Reports** dialog provides the **Cancellation Summary Report** file only when you have recently cancelled some callbacks. This report shows the cancellation status of the recently-cancelled callbacks.

Configuring Country and Number Validation rules

Contents

- 1 Configuring Country and Number Validation Rules
- 2 Assigning rules to queues
- 3 Editing and deleting rules
- 4 Testing a number pattern

-
- Administrator

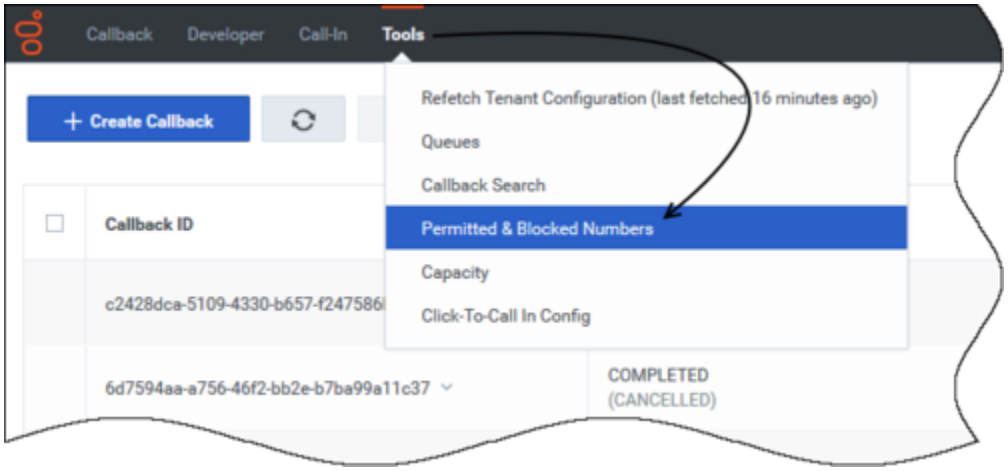
As a Callback administrator, you want the added security of blocking callbacks to certain numbers and countries. To do that, you'll configure a Country and Number Validation rule in Genesys Callback and assign it to at least one callback queue. You can test your rules before assigning them to a queue or when troubleshooting. You can also edit and delete existing rules, as required.

Related documentation:

-

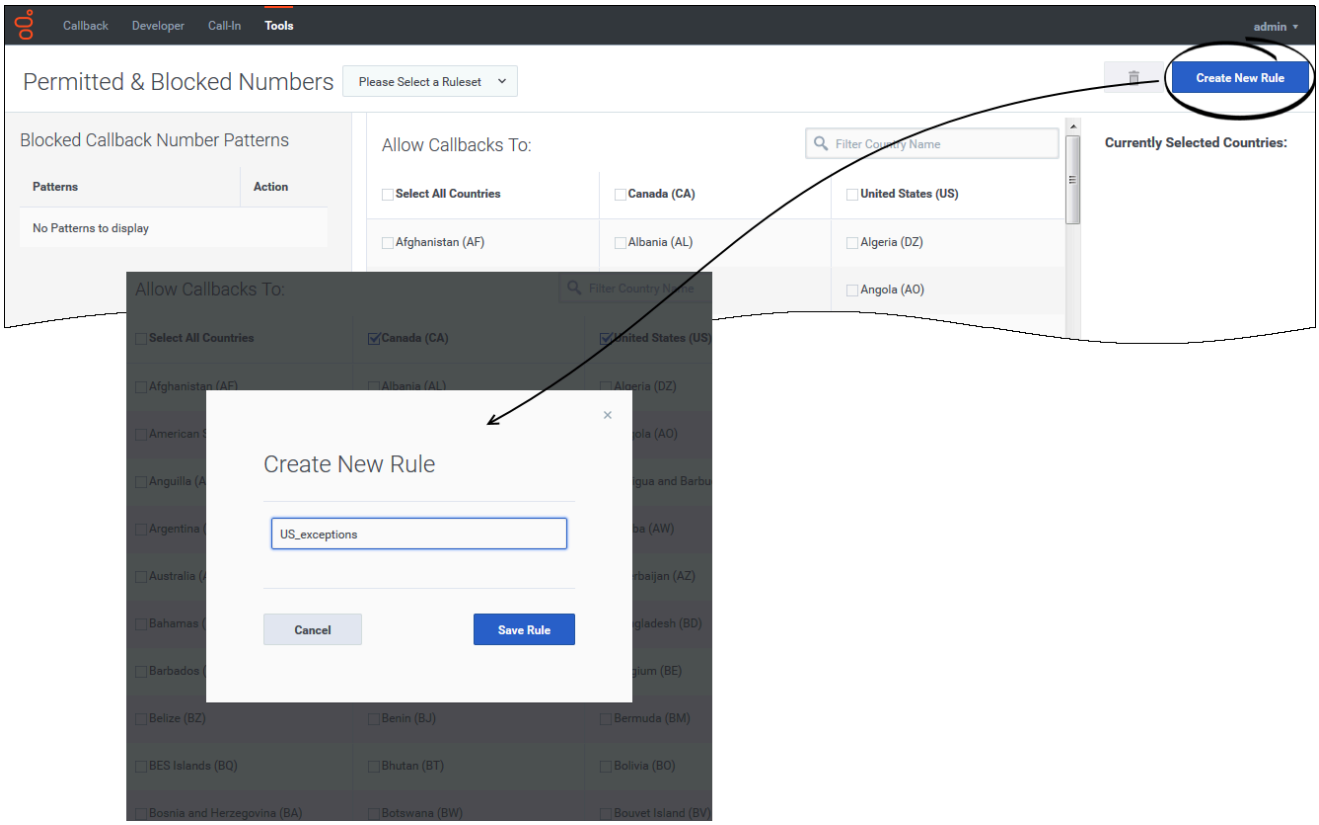
> **Important**
> To create, edit, or delete country and number validation rules, you must be a member of the **Callback Administrator**, **Callback Developer**, or **Callback Supervisor** role.

To avoid initiating callbacks to certain countries or to numbers that you know to be fraudulent, you can configure country and number validation rules on the **Tools** > **Permitted & Blocked Numbers** page, and then assign those rules to queues on the Callback Queues page. You can create a rule to specify the list of countries to which you permit callbacks for specific queues. You can also optionally enhance a rule by defining simple or complex (regular expression) numeric patterns that will block callback attempts to matching numbers.

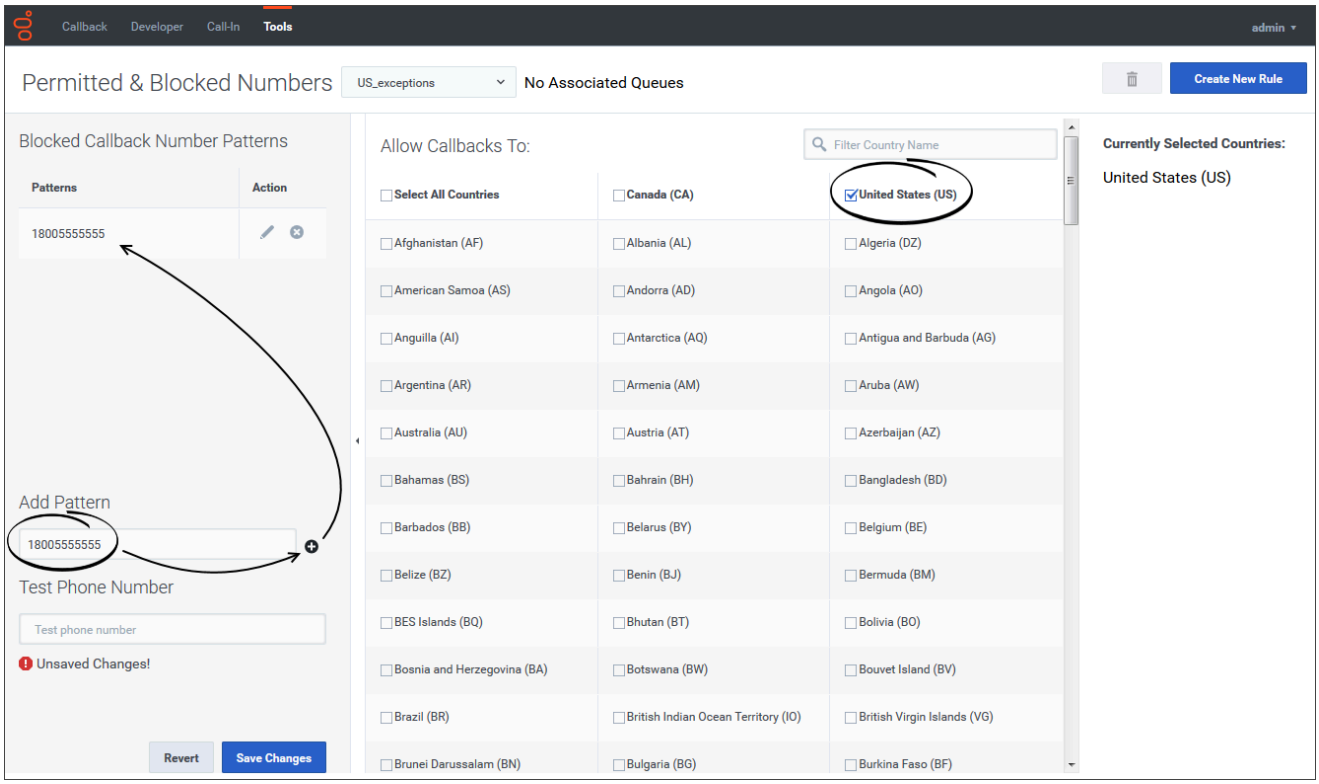


Once a rule is configured, you can assign it to as many queues as necessary. It is important to note, though, that you can assign only one rule to each queue. Before creating a rule, consider to which queue or queues you will assign it and make sure that the rule incorporates the countries to which you permit callbacks for that queue as well as any numeric patterns in any of those countries that you want to block. Rules are checked before the system initiates a callback.

Configuring Country and Number Validation Rules



The first step in creating a new country and number validation rule is to give the rule a name. Later, you will need to pick the rule from a list when you assign it to a queue, so make sure the name is descriptive enough for you to understand what it allows and denies.



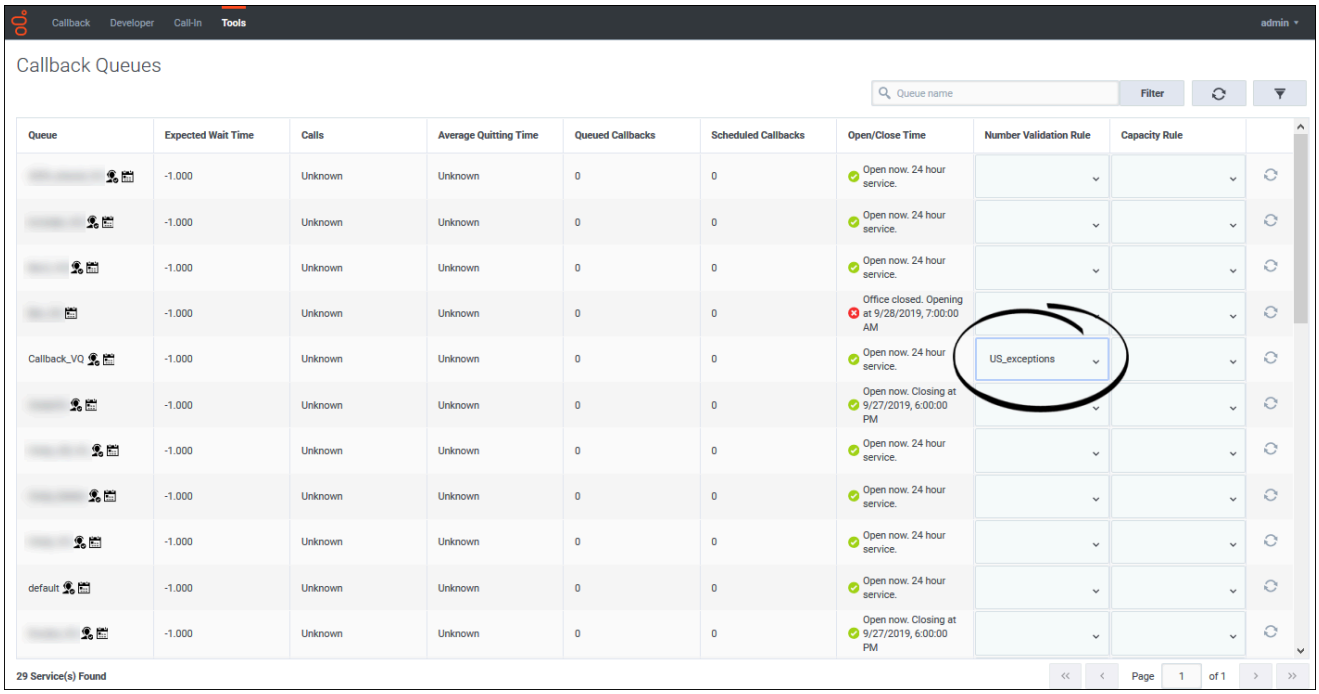
When creating a rule, you must select at least one country on the **Permitted & Blocked Numbers** page. You can enter *blocked number patterns* only after you make country selections. Adding a country to the rule means that you allow callbacks to that country. For countries that are not added to a rule, no callback attempt will be initiated on the queue or queues to which the rule is applied.

After you have selected the countries to which you allow callbacks, enter any patterns that you want to block. Patterns must be entered as regular expressions. For example, if you have a list of suspicious phone numbers for the selected countries and you do not want the system to make callback attempts to those numbers, then add those to the rule.

The figures on this page show the addition of specific numbers, but you could enter `^1234[0-9]+`, which would match any phone number that starts with 1234. While a rule must contain at least one country, the addition of blocked number patterns is optional.

To add a blocked number pattern, simply enter it in the field and click "plus" (⊕). You must enter patterns individually; that is, enter a pattern and add it to the rule before entering another pattern.

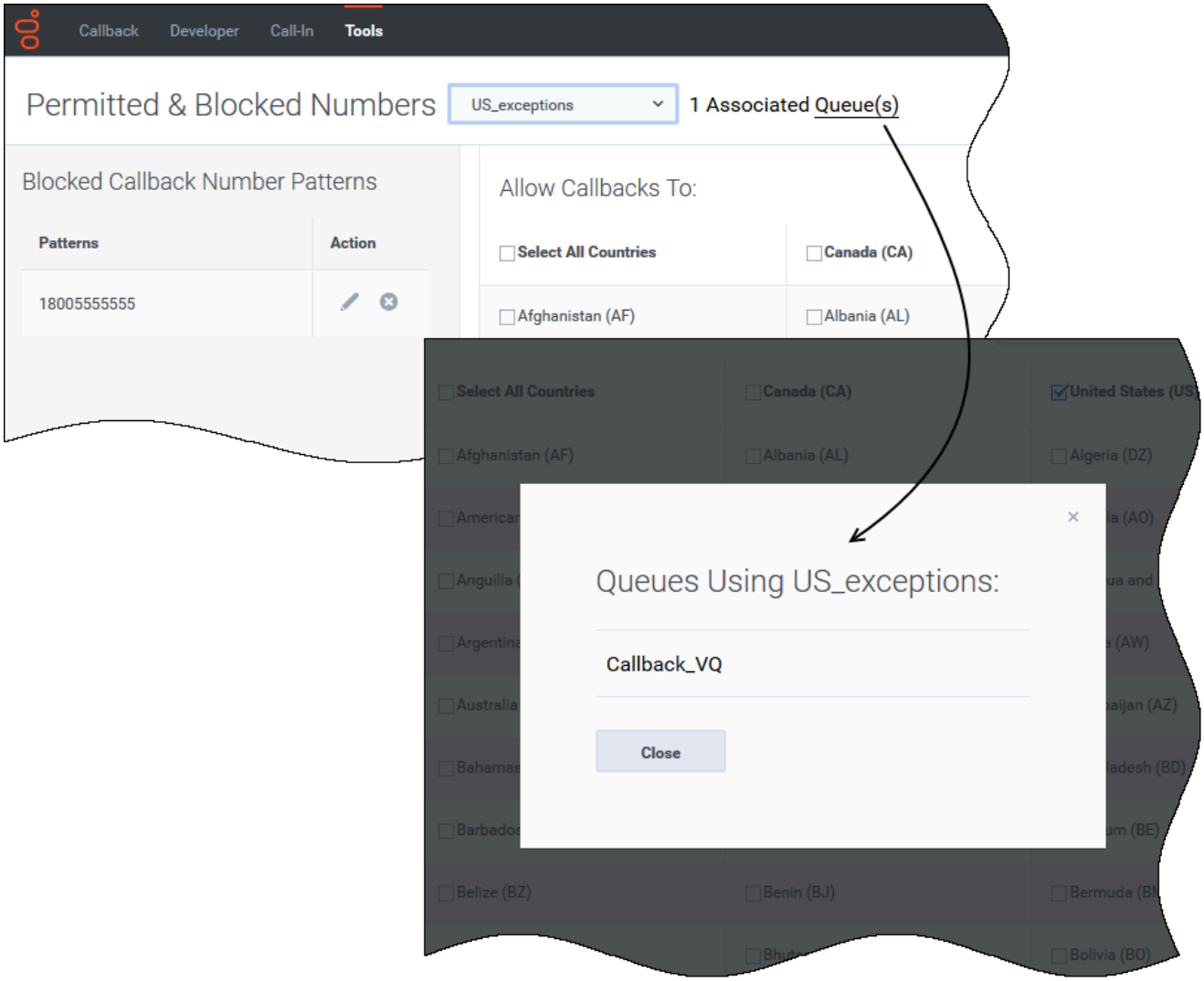
Assigning rules to queues



After you have configured at least one rule, you can assign the rules to your queues. You can assign a rule to multiple queues, but you can assign only one rule to each queue.

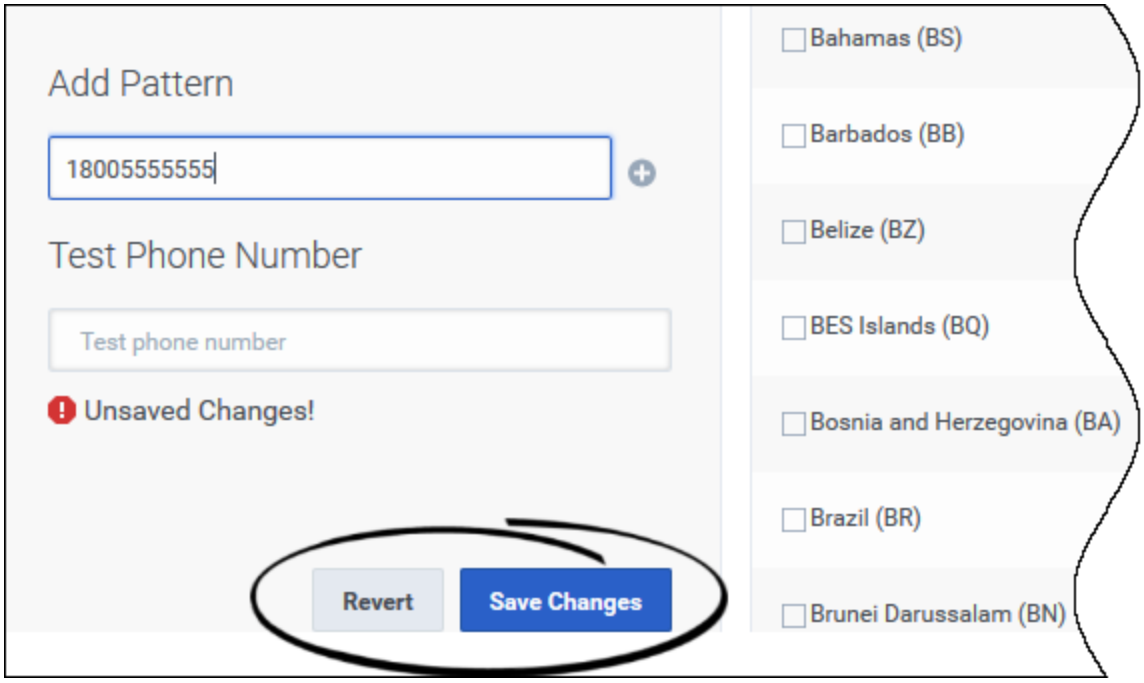
To assign a rule to a queue, navigate to the **Tools** > **Queues** page in your Callback UI. Locate a queue to which you want to assign a rule. You might need to filter the list of queues to locate what you're looking for. Once you find the queue, open the drop-down menu in the **Number Validation Rule** column. Select the rule that you want to assign to the queue. The rule is applied to the queue immediately and that change to the queue is saved automatically.

Editing and deleting rules



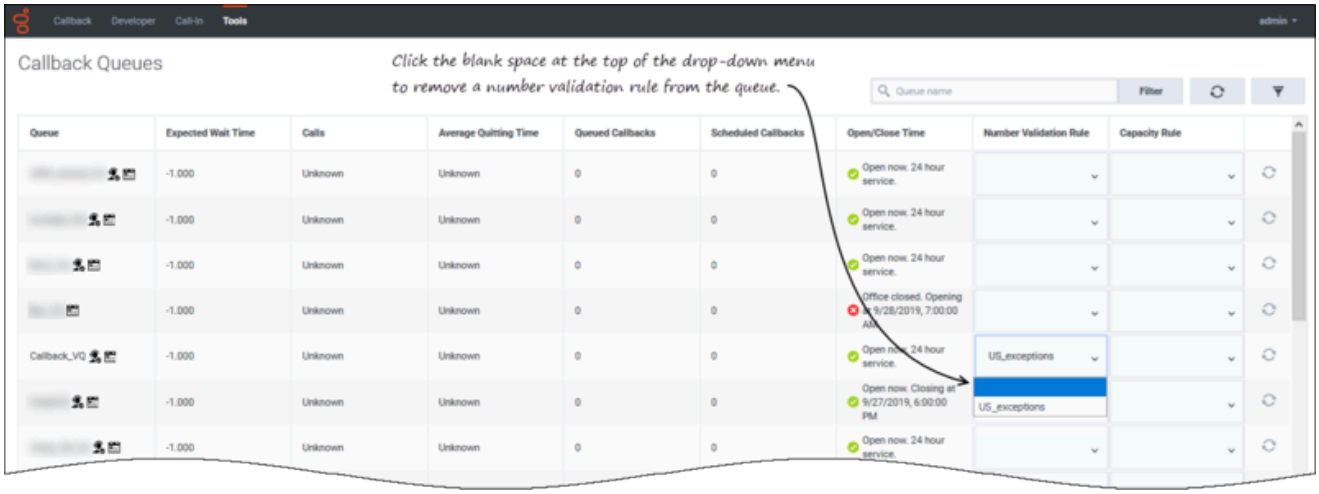
For your reference, each rule displays the list of queues to which the rule is assigned (or indicates that the rule is not yet assigned) so you will know which queues, if any, you are impacting when you edit a rule.

Changes that you make to a rule's definition are applied to any queue to which the rule is assigned as soon as you click **Save Changes**. If, while editing a rule, you decide that you want to discard your changes – perhaps you have made some mistakes or you discover that you are editing the wrong rule – click **Revert** to return the rule to its last-saved configuration.

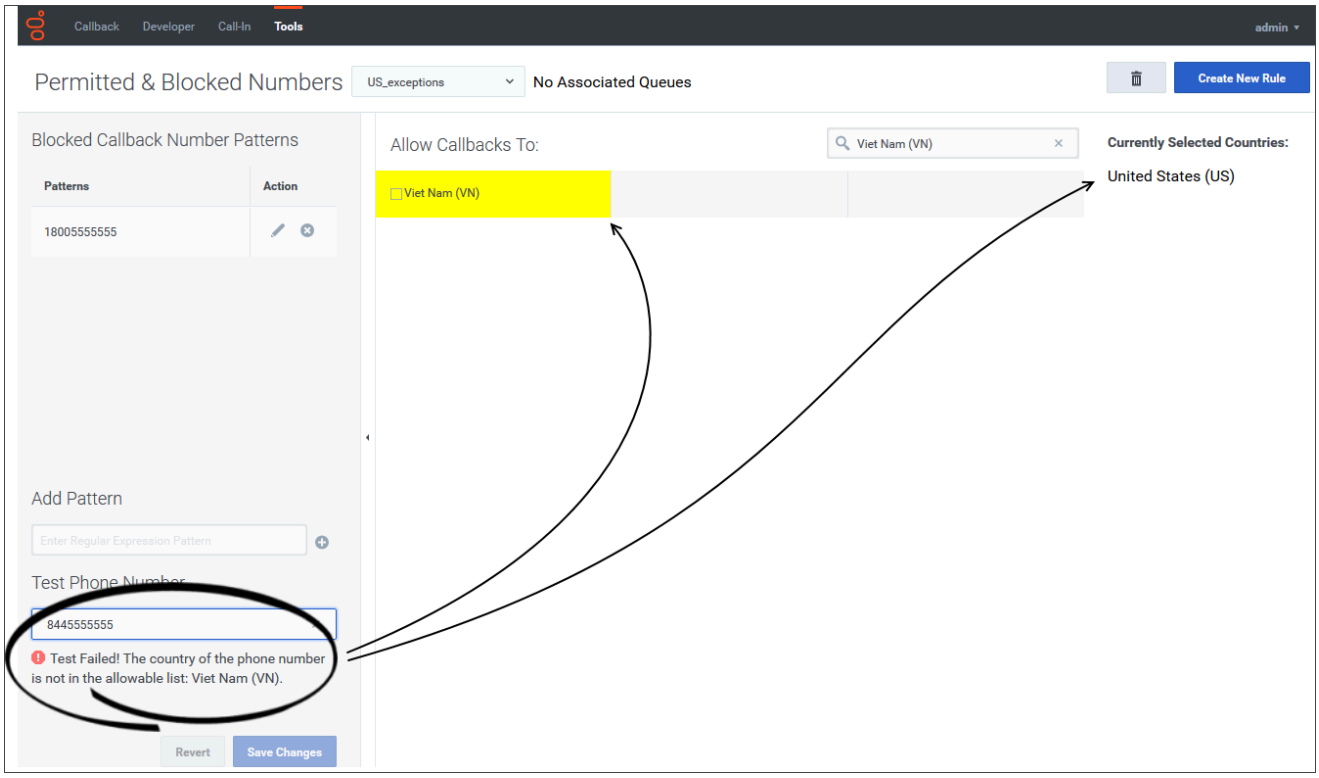


To help you manage impacts to queue activity, the Callback UI does not allow you to delete a rule that is assigned to a queue. Use the list of queues associated with the rule to help you determine if the rule can be safely removed from each assignment.

Navigate to the **Callback Queues** page in the Callback UI to remove a rule from a queue. Once the rule is removed from all queue assignments, you can delete it.



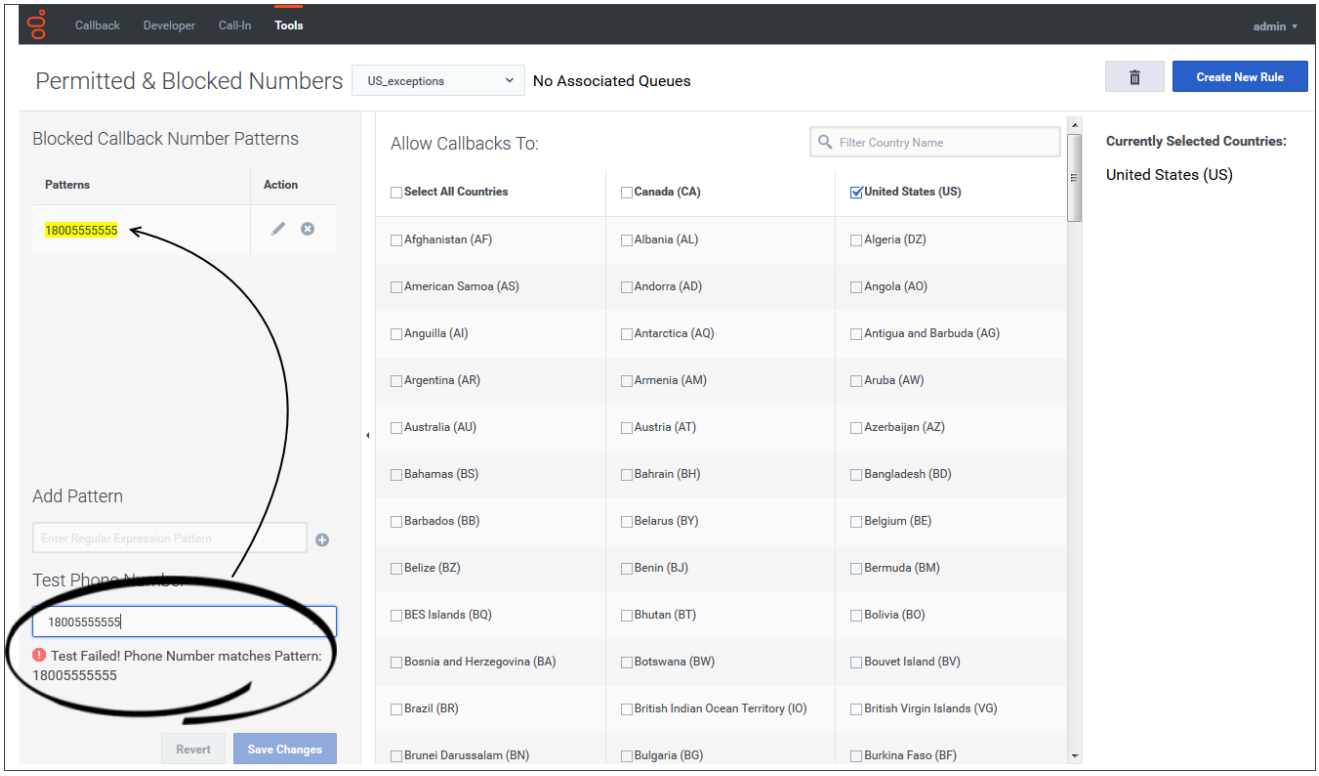
Testing a number pattern



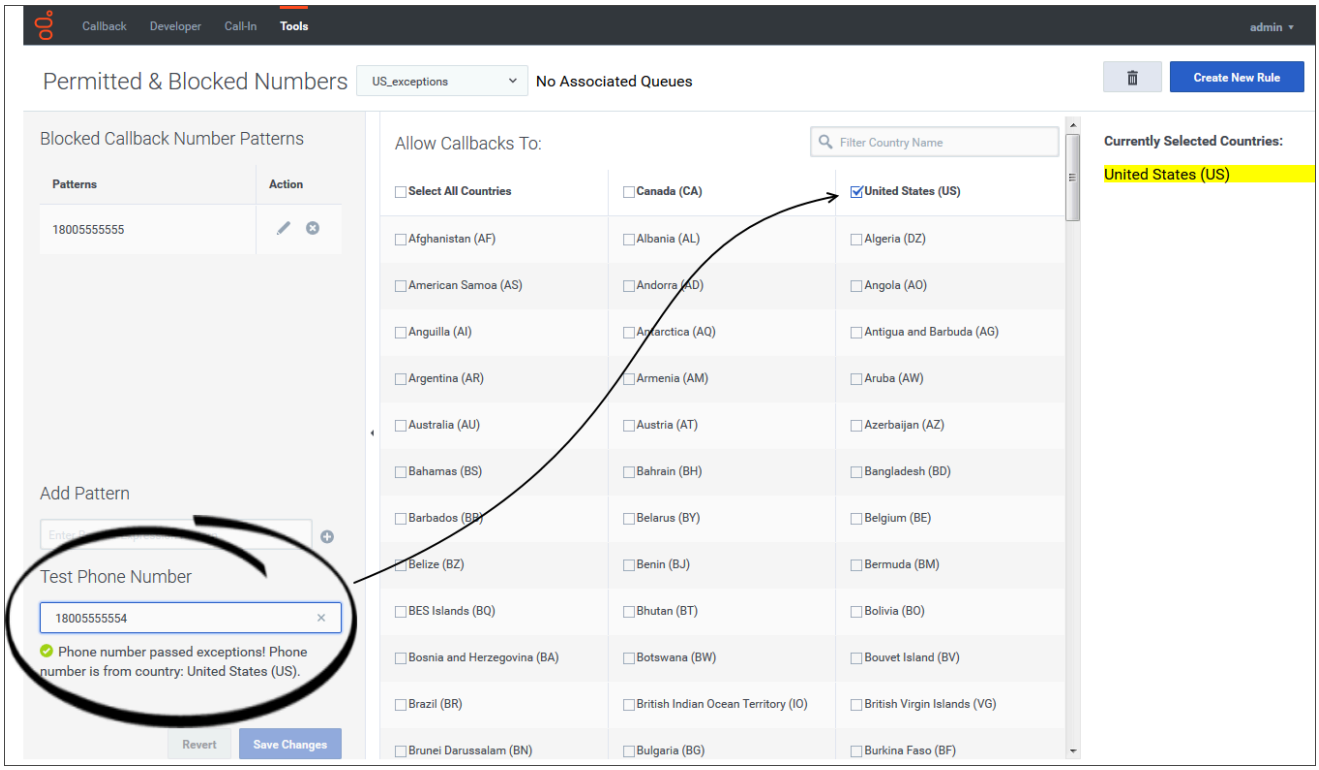
After you have created and saved a rule, you can use the rule to validate any number quickly using the testing feature on the **Permitted & Blocked Numbers** page. For example, you might be troubleshooting an error and you want to see if a rule might be interfering with the callback.

Enter a number in the **Test Phone Number** field. The number fails the rule's validation test if the number belongs to a country that is not added to that rule or if the number matches any one of the defined blocked number patterns.

If the number fails the check against allowable countries, then the system returns this reason for failure and does not validate the number against the blocked number patterns, even though the number might also fail that test.



If a number passes the country validation test, then it is checked against the blocked number patterns.



If the number belongs to a country to which callbacks are allowed for that rule and the number is not listed within the blocked number patterns, then the system validates that the number is acceptable for callback attempts.

Configuring Capacity rules

Contents

- 1 Time zone associations
- 2 How Capacity rules affect capacity settings configured in Designer
- 3 Before you start
- 4 Using a rule to adjust capacity
- 5 Configuring a sample Capacity rule
- 6 Assigning a Capacity rule to a queue
- 7 Using JSON-format capacity statements to configure a Capacity rule
- 8 How capacity is evaluated for a short time interval
- 9 Switching the timeslot length in a saved Capacity rule

-
- Administrator

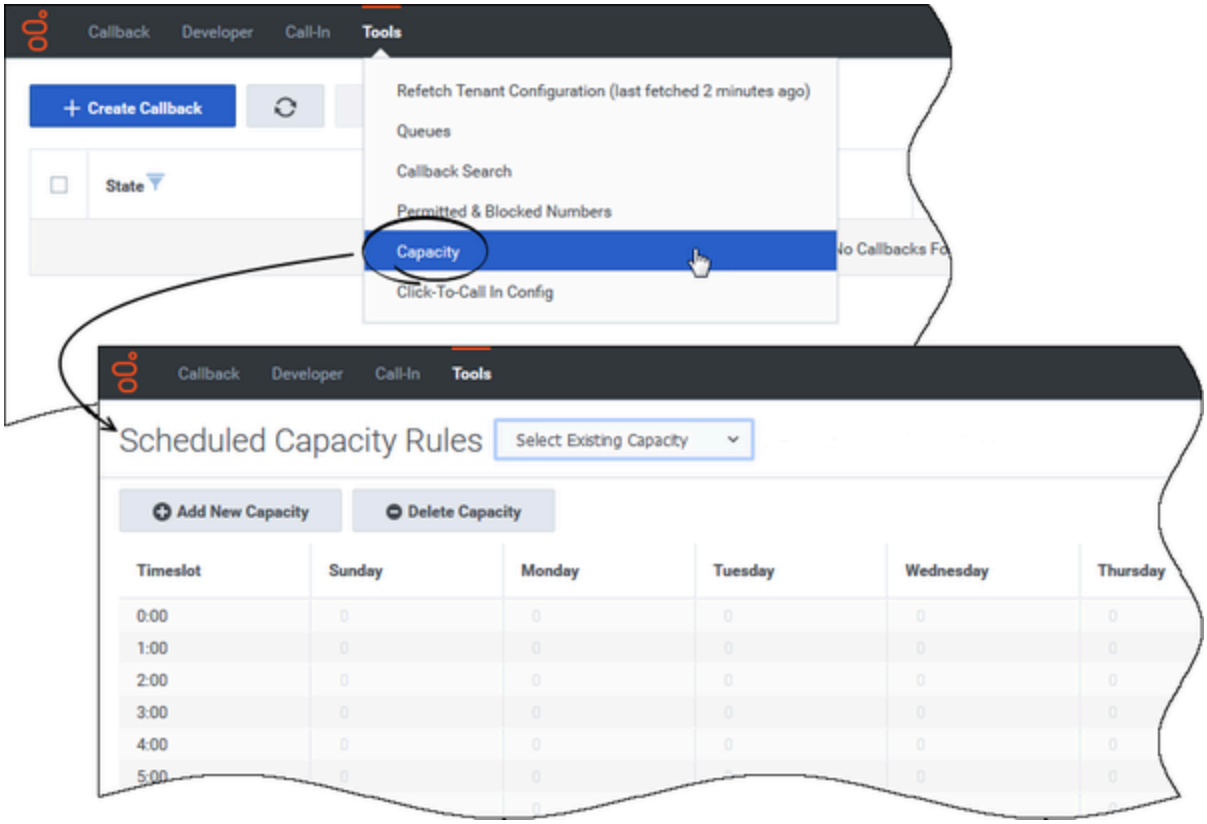
You want to manage the number of scheduled callbacks that are allowed during specific time intervals. To do that, you'll create a Capacity rule in Genesys Callback and assign it to a callback queue. We'll walk you through the process and explain the implications of using a Capacity rule in relation to Designer. We'll also tell you about a shortcut method for entering values in your Capacity rule (JSON-format capacity statements).

Related documentation:

-

Callback capacity refers to the maximum number of scheduled callbacks that are allowed in a given time interval.

> **Important**
> To create or edit Capacity rules, you must be a member of the **Callback Administrator** role. The **Callback Developer**, **Callback Supervisor**, and **Callback Monitor** roles have read-only access. Access to queues is subject to line-of-business filtering where role-based access control is enabled.



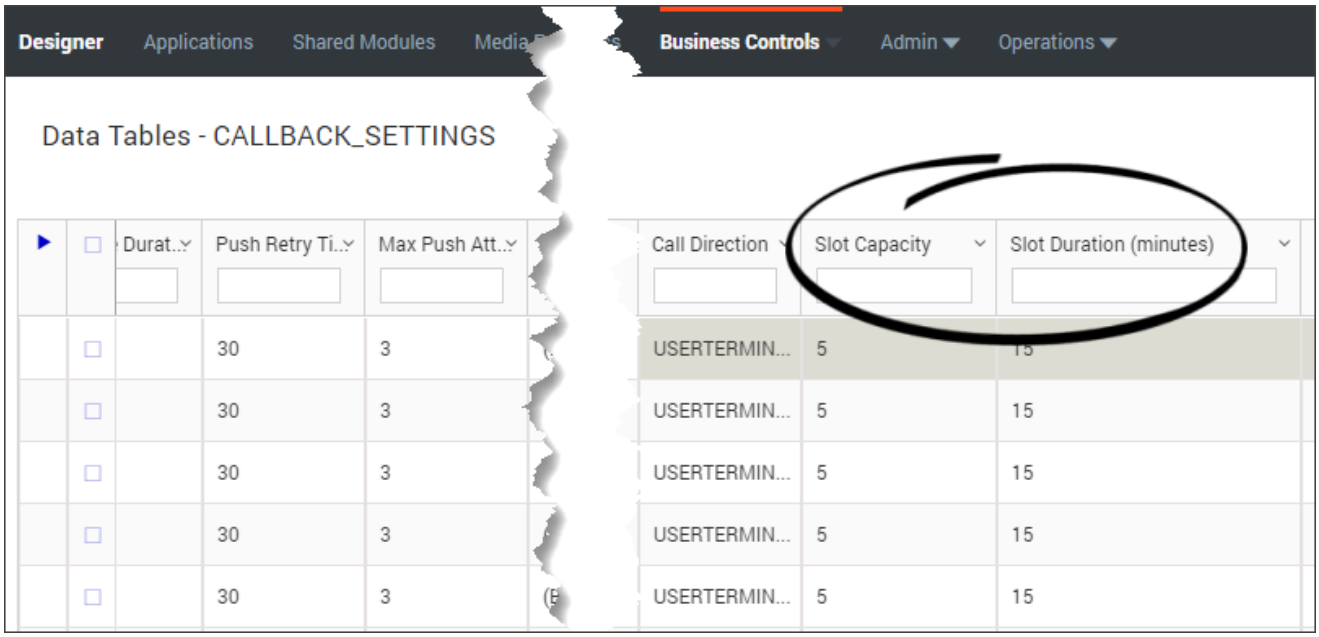
On the **Tools** > **Capacity** tab, you create Capacity rules. A Capacity rule contains a day-by-day weekly calendar on which you specify callback capacity for each time slot. A time slot is either a one-hour slot or a half-hour slot; you select the time slot duration from a drop-down menu. Once you have configured a Capacity rule, you can assign it to queues. A Capacity rule can be assigned to multiple queues, but you can assign only one Capacity rule to each queue.

Time zone associations

There is no time zone associated with a Capacity rule. During the evaluation process to determine if a scheduled callback can be booked or not, the time zone associated with the business hours service is used. Be sure you have the business hours, including the time zone, configured (in Designer) for the queue before you apply any Capacity rules. For more information about configuring business hours, see [[PEC-ROU/Current/Designer/DesBusinessHours|]] and [[PEC-ROU/Current/Designer/BusinessHrs|]] in the Designer documentation.

How Capacity rules affect capacity settings configured in Designer

A Capacity rule that you configure in the Callback UI and assign to a queue has an effect on the **Slot Capacity** and **Slot Duration** settings in Designer's CALLBACK_SETTINGS data table.



A value that you specify for a time slot on the **Capacity** page in Callback is the maximum number of scheduled callbacks that you can support *for each* ***Slot Duration*** (configured in Designer) within that time interval. For example, if you configure a Capacity rule for 9:00 to 10:00 a.m. on Monday morning with a value of 5 and you assign that rule to a queue for which the **Slot Duration** in Designer is 15 minutes, then what the rule allows is a maximum of 5 scheduled callbacks every 15 minutes between 9:00 and 10:00 a.m. on Monday morning.

A Capacity rule assigned to a queue *overrides* the **Slot Capacity** setting in the Designer CALLBACK_SETTINGS data table for that queue. For example, if you configure a rule for 3:00 to 4:00 p.m. on Wednesday afternoon with a value of 5 and you assign that rule to a queue for which the **Slot Capacity** in Designer is set to 3, then the rule allows a maximum of 5 scheduled callbacks for each **Slot Duration** during that hour on Wednesday afternoon, overriding the value of 3 that was set in the CALLBACK_SETTINGS table.

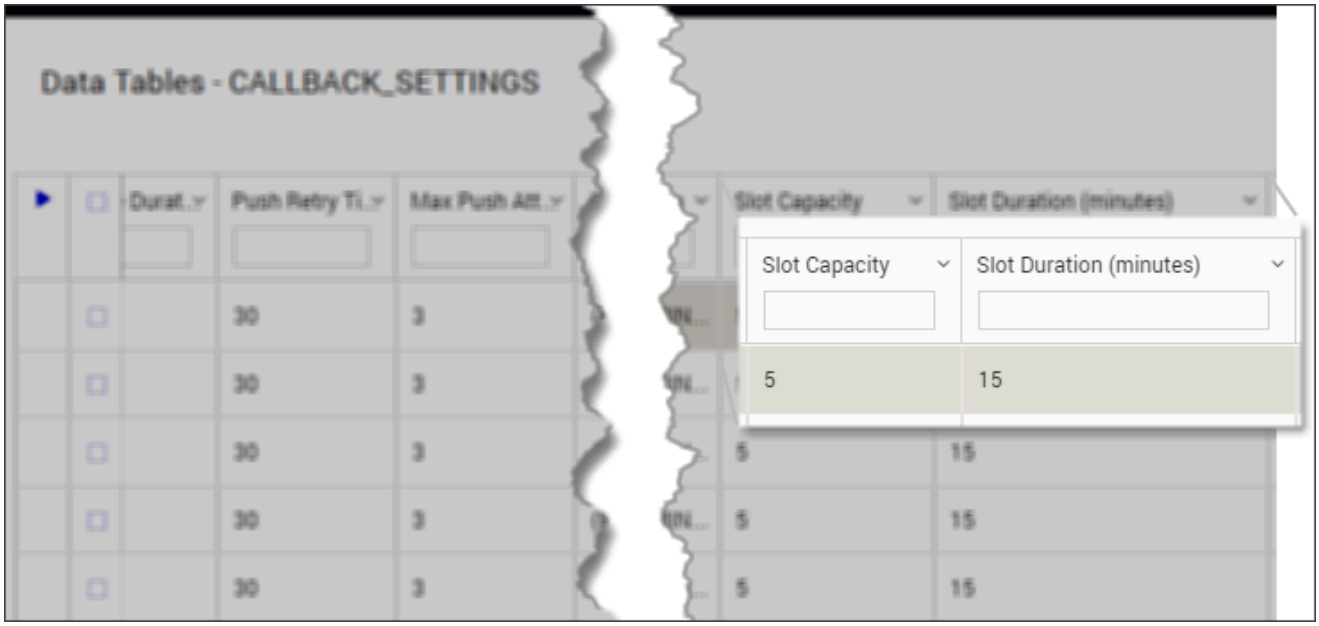
Before you start

Before you create Capacity rules and assign them to queues, make sure that you have provisioned Callback in Designer. In particular, make sure that the following configuration is complete:

- In Designer, you have provisioned your business hours, including the time zone. See also [[PEC-ROU/Current/Designer/DesBusinessHours|]] and [[PEC-ROU/Current/Designer/BusinessHrs|]] in the Designer documentation.
- In Designer, you have provisioned the callback services, which includes specifying a value for the **Slot Capacity** and **Slot Duration (minutes)** columns for your callback application. See also Callback Settings Data Table in the Designer documentation.

Using a rule to adjust capacity

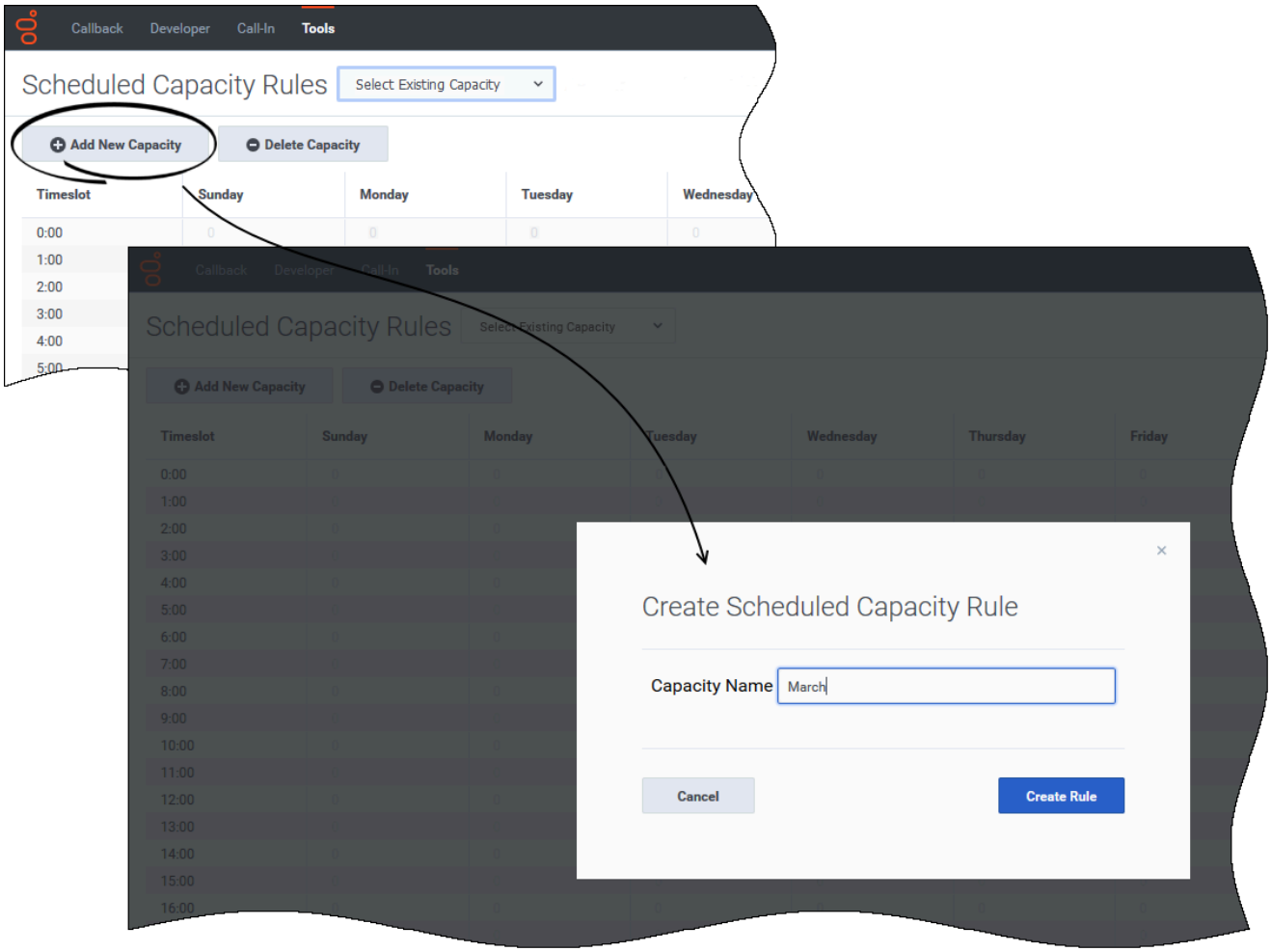
Let's assume that we have a special campaign running in the month of March because we are launching a new product feature. We anticipate higher than normal call volumes, so we plan to increase the number of skilled agents to support customer inquiries. We also anticipate more callback requests, so we plan to adjust the capacity for scheduled callbacks during the entire month of the launch.



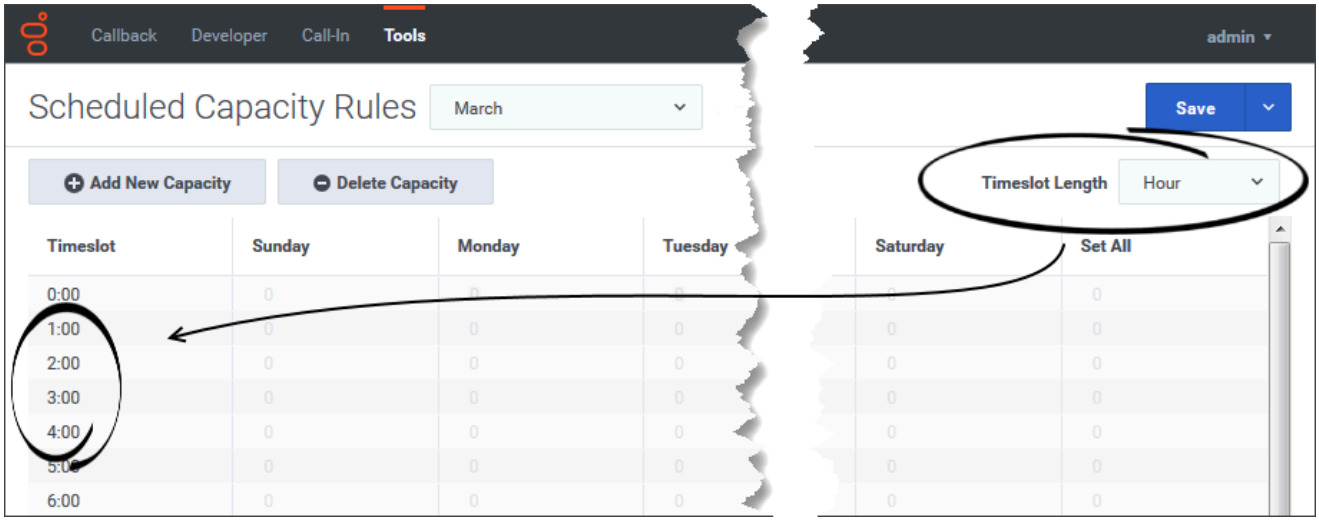
In the CALLBACK_SETTINGS data table in Designer, let's say that we have set the capacity for scheduled callbacks to 5 and specified a slot duration of 15 minutes for the callback application. In other words, on a normal day we can handle 5 scheduled callbacks every 15 minutes during open office hours. In Designer, we set this "flat rate" for capacity, but, for the month of March, we want to make some adjustments to that. We make the adjustments using a Capacity rule that we'll configure and assign to queues in Callback.

Let's say that our business hours are 8:00 a.m. to 6:00 p.m., Monday to Friday. For the month of March, we plan to have 40 agents from 8:00 a.m. to noon, 50 agents from noon to 2:00 p.m., and back to 40 agents from 2:00 to 6:00 p.m. Slot duration in the CALLBACK_SETTINGS table is, as previously mentioned, set to 15 minutes.

Configuring a sample Capacity rule



To create a Capacity rule, you must first give it a name. You cannot edit the name of a rule after you click **Create Rule**, so be sure that you have entered the name exactly as you intended. We'll call this new rule **March** because we are making adjustments to the capacity for that month. The rule can be applied to all relevant queues for the month of March to override the setting in Designer.



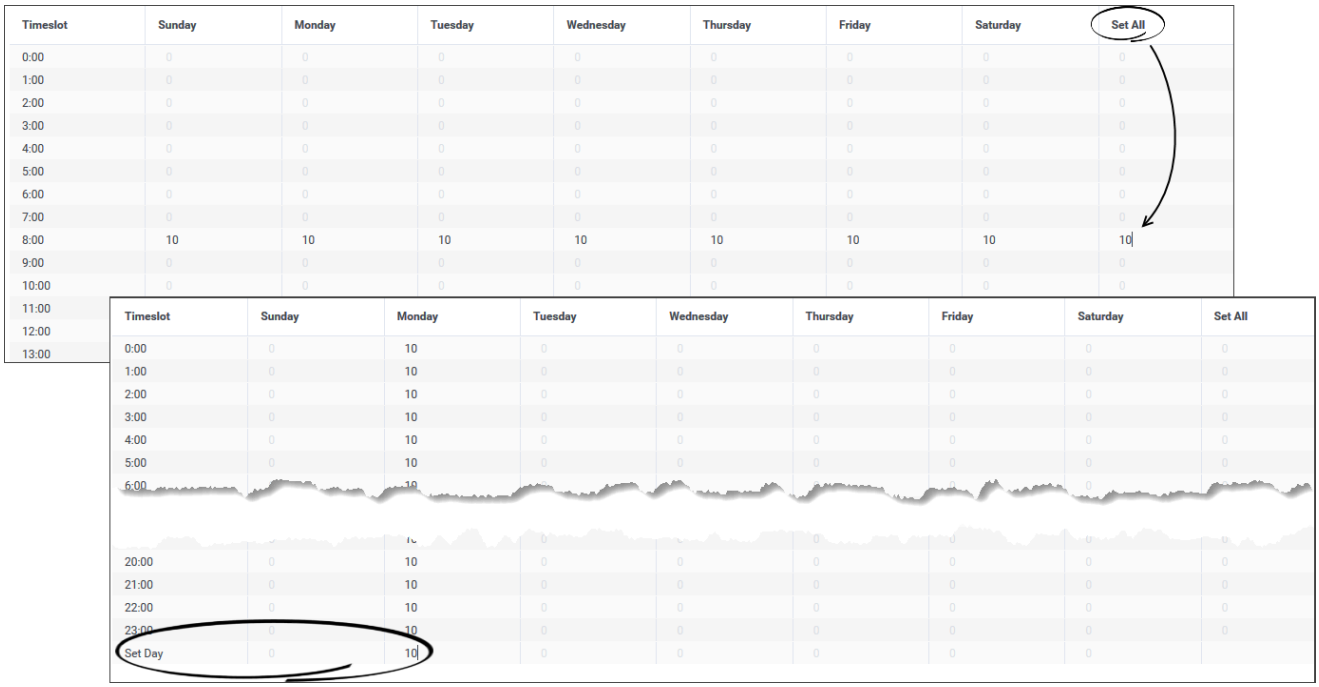
Select the **Timeslot Length**. This value sets the time slot duration on the *capacity calendar*, it does not change the **Slot Duration** that you set in Designer. This setting governs all time slots on the calendar; you cannot have some half hour-long slots and some hour-long slots.

For this example, we will set the timeslot length to Hour.

An hourly time slot begins at the top of every hour (:00). Half-hourly time slots begin at the top of every hour (:00) and at the bottom of every hour (:30). If you plan to use Capacity rules with queues, Genesys recommends that you always set the **Slot Duration (minutes)** in Designer to a value that is evenly divisible into 60. Otherwise, you might experience unpredictable results when a Capacity rule is applied.

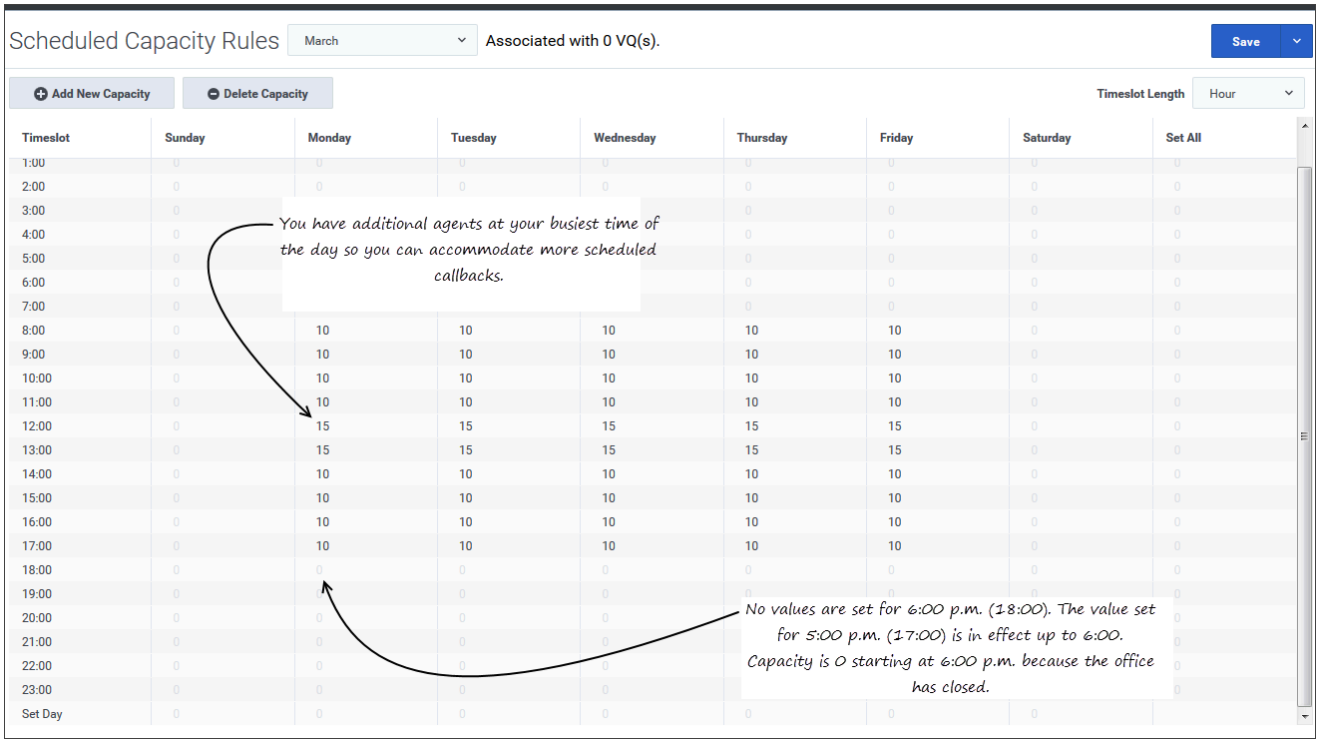
Once your capacity calendar is set up, then you can start adding the capacity values. Remember that the capacity values that you specify on the **Capacity** page indicate the maximum number of scheduled callbacks that you have the capacity to handle in each **Slot Duration** that occurs within the hour or half-hour that you're configuring. To find the **Slot Duration** that is configured for a particular queue, you must check the CALLBACK_SETTINGS data table in Designer.

If your contact center is open 24/7 and you want to set the same value for the same time slot on every day of the week, enter the value in the **Set All** cell for that time slot. To set the same value for every time slot on a particular day, enter the value in the **Set Day** cell for that day.



When creating a Capacity rule, if you do not enter a value for a particular time slot, then the system reads that time slot as having 0 or *no capacity* when determining if a scheduled callback can be booked.

Currently, Callback does not support the definition of exceptions for specific dates; for example, statutory holidays.



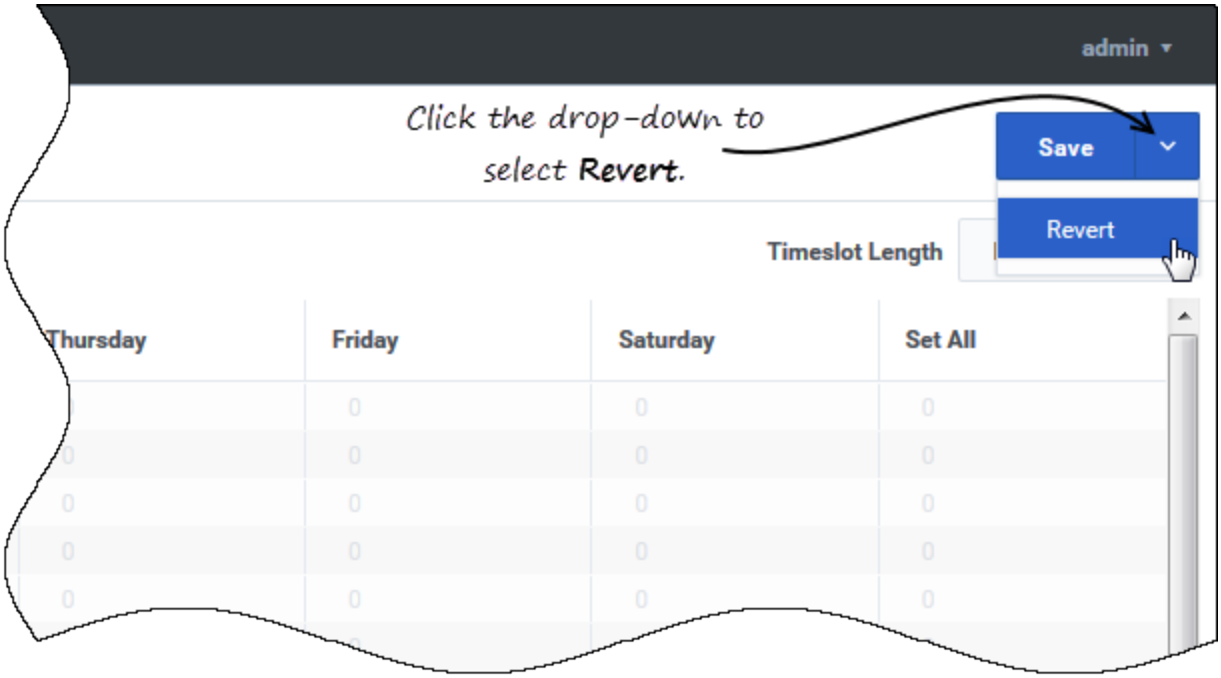
Timeslot	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Set All
1:00	0	0	0	0	0	0	0	0
2:00	0	0	0	0	0	0	0	0
3:00	0				0	0	0	0
4:00	0				0	0	0	0
5:00	0				0	0	0	0
6:00	0				0	0	0	0
7:00	0				0	0	0	0
8:00	0	10	10	10	10	10	0	0
9:00	0	10	10	10	10	10	0	0
10:00	0	10	10	10	10	10	0	0
11:00	0	10	10	10	10	10	0	0
12:00	0	15	15	15	15	15	0	0
13:00	0	15	15	15	15	15	0	0
14:00	0	10	10	10	10	10	0	0
15:00	0	10	10	10	10	10	0	0
16:00	0	10	10	10	10	10	0	0
17:00	0	10	10	10	10	10	0	0
18:00	0	0	0	0	0	0	0	0
19:00	0	0	0	0	0	0	0	0
20:00	0	0	0	0				0
21:00	0	0	0	0				0
22:00	0	0	0	0				0
23:00	0	0	0	0				0
Set Day	0	0	0	0	0	0	0	

Returning to our sample Capacity rule (**March**), we will set values for the days and hours that we're open (8:00 a.m. to 6:00 p.m., Monday to Friday) based on the additional agents that we have during the month of March. Slot duration in the CALLBACK_SETTINGS table is, as previously mentioned, set to 15 minutes. This means that we will be overriding the 5 scheduled callbacks that we normally have capacity to handle every 15 minutes with the following capacity values:

- up to 10 scheduled callbacks every 15 minutes from 8:00 a.m. to 12:00 p.m.
- up to 15 scheduled callbacks every 15 minutes from 12:00 p.m. to 2:00 p.m. (there are extra agents on phones during peak hours)
- up to 10 scheduled callbacks every 15 minutes from 2:00 p.m to 6:00 p.m.

Once this Capacity rule is assigned to the callback virtual queue, the capacity specified in the rule takes effect.

After you have set some or all of the values, remember to save the rule. You can edit the values at any time. Saved changes to rules will be applied immediately to any queue to which the rule is assigned.

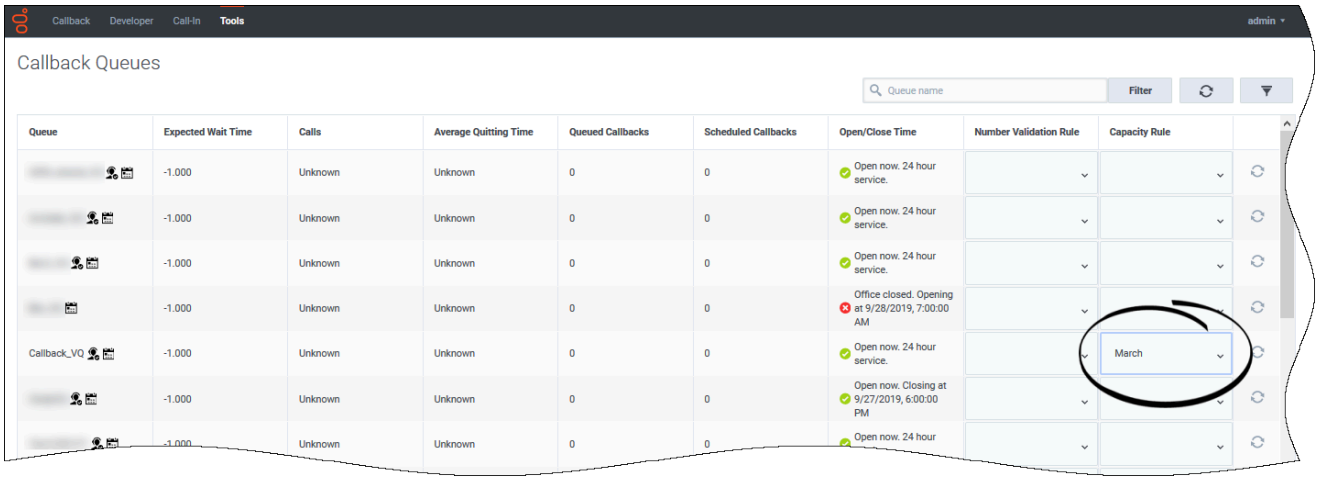


If you edit values in the calendar after it has been saved and you discover that you have made some mistakes or you decide against the changes that you've made, then click **Revert** to return to the last-saved values.

Assigning a Capacity rule to a queue

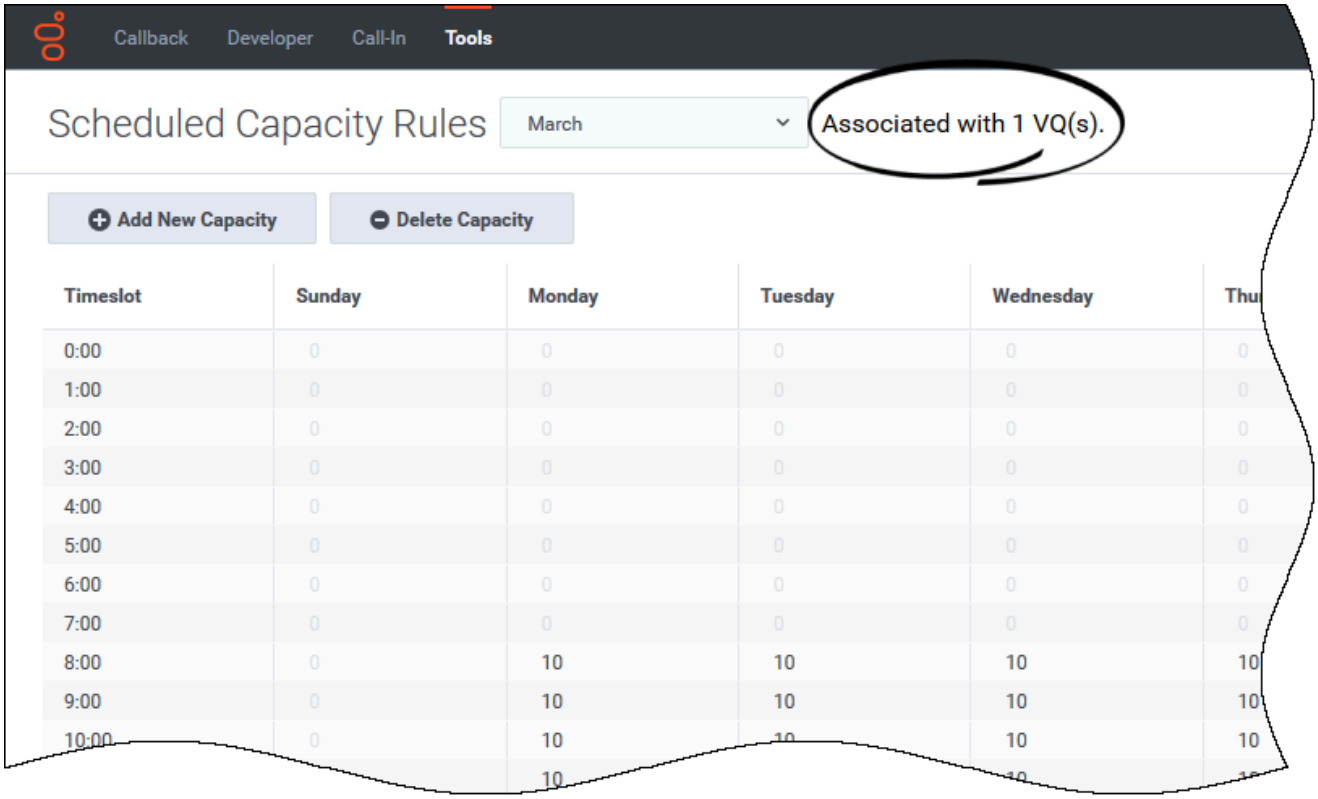
A rule is created based on one week (Sunday to Saturday), but the capacity specified in the rule for each time slot of every day remains in effect on a queue for as long as the rule is assigned to that queue. After you have configured at least one rule, you can assign the rule or rules to your queues. You can assign a rule to multiple queues, but you can assign only one rule to each queue.

To assign a rule to a queue, navigate to the **Tools** > **Queues** page in your Callback UI. Locate a queue to which you want to assign a rule. You might need to filter the list of queues to locate what you're looking for. Once you find the queue, open the drop-down menu in the **Capacity Rule** column. Select the rule that you want to assign to the queue. Continuing our example, we will assign the **March** Capacity rule to the callback virtual queue. The rule takes effect immediately on that queue.



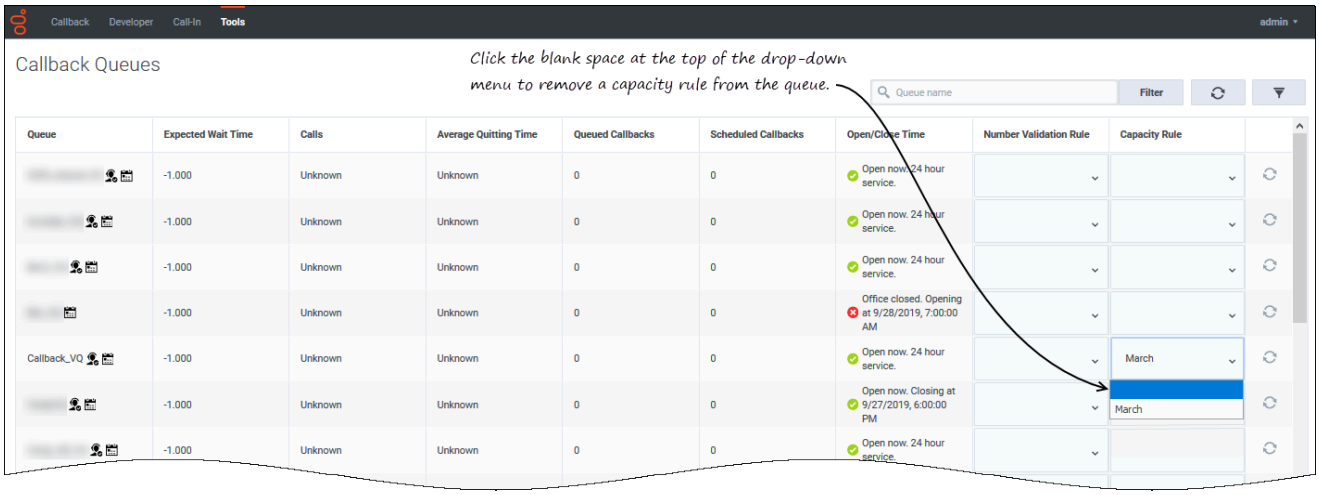
Tip

You can select a rule on the **Scheduled Capacity Rules** page to see how many queues the rule is assigned to. This is important if you want to delete a rule. As long as it is assigned to at least one queue, you won't be able to delete it.



Once March is over and we want to return to our normal level of scheduled callbacks, we simply

remove the assignment of the **March** rule from the queue.



Using JSON-format capacity statements to configure a Capacity rule

You can create a Capacity rule and manually enter your capacity for various hours and days or you can define your capacity in JSON format and then copy and paste the JSON capacity statement directly into an open and focused Capacity rule using your usual pasting shortcut; for example, Command/Ctrl+V or the contextual menu. Make sure that no cells are selected in the capacity calendar. If a cell is selected, then the capacity data will be pasted into only the selected cell.

When you paste a JSON-format capacity statement into a rule that already has data, the pasted capacity data completely overwrites the existing data. For example, let's say that we have a Capacity rule already defined and it includes a capacity of 5 on Tuesday from 9:00 a.m. to 10:00 a.m. We paste a JSON-format capacity statement into the rule, but the pasted capacity data does not include a value for the Tuesday, 9:00 a.m. to 10:00 a.m. time slot. After we paste in the JSON-format capacity statement, we notice that the capacity for that time slot has changed to 0. This happened because the pasted-in capacity statement did not merge with the existing data, it wrote over it. If nothing is defined for the 9:00 to 10:00 time slot on Tuesday morning, then the UI interprets that as a capacity of 0. If you want to maintain existing data, remember to include that data in your JSON-format capacity statement.

The following sample shows the correct format to use for JSON capacity statements:

```
{"days":{"sun":{"02:00":2},"mon":{"05:00":2},"tue":{"07:00":3,"06:00":4},"wed":{"08:00":5},"thu":{"11:00":6},"fri":
{"18:00":6},"sat":{"19:00":6}},"duration":60}
```

How capacity is evaluated for a short time interval

The callback system, when queried for availability, first checks the business hours that are configured in Designer. For more information about configuring Business Hours, see [[PEC-ROU/Current/Designer/DesBusinessHours|]] and [[PEC-ROU/Current/Designer/BusinessHrs|]].

As mentioned earlier on this page, the Callback Capacity rules affect the **Slot Duration** and **Slot Capacity** settings that are configured in Designer. Because the Capacity rules are based on callback capacity for time slots that begin at either the top of the hour or the bottom of the hour, there are situations in which a time slot might be ignored (no callbacks will be scheduled in that slot). For example, if your office hours begin at 7:50 a.m., **Slot Duration (minutes)** is set to 15 minutes in Designer, and a Capacity rule assigned to the callback queue specifies a capacity of 10 scheduled callbacks between 7:30 and 8:30 a.m., then the system will begin scheduling callbacks for 8:00 a.m. There can be up to 10 scheduled callbacks between 8:00 and 8:15, and another 10 between 8:15 and 8:30. The system will not schedule callbacks between 7:30 and 7:45 because the business is not open during that slot, and the slot 7:45 to 8:00 is also considered to be unavailable because the business is still closed at 7:45. In other words, Callback checks the conditions only at the start of a **Slot Duration** period.

Switching the timeslot length in a saved Capacity rule

If a Capacity rule was initially configured with half-hourly time slots in Callback, and you later decide that you want to change it to hourly time slots, then you can certainly do so. The capacity values that were originally configured for the bottom of each hour are "dropped" and the capacity values that were set at the top of each hour are used instead.

Original half-hourly Capacity rule settings:

Time:	9:00	9:30	10:00	10:30
Capacity value:	8	10	8	10

Half-hourly converted to hourly: When you convert a half-hourly Capacity rule to hourly, then the value that was set at the top of the hour is duplicated at the half hour.

Time:	9:00	(9:30)	10:00	(10:30)
Capacity value:	8	(8)	8	(8)

Original hourly Capacity rule settings:

Time	9:00	10:00
Capacity value	8	8

Hourly converted to half-hourly: When you convert an hourly Capacity rule to half-hourly, then the value that was set at the top of the hour is duplicated at the half hour.

Time	9:00	9:30	10:00	10:30
Capacity value	8	8	8	8

Configuring Region Affinity rules

Contents

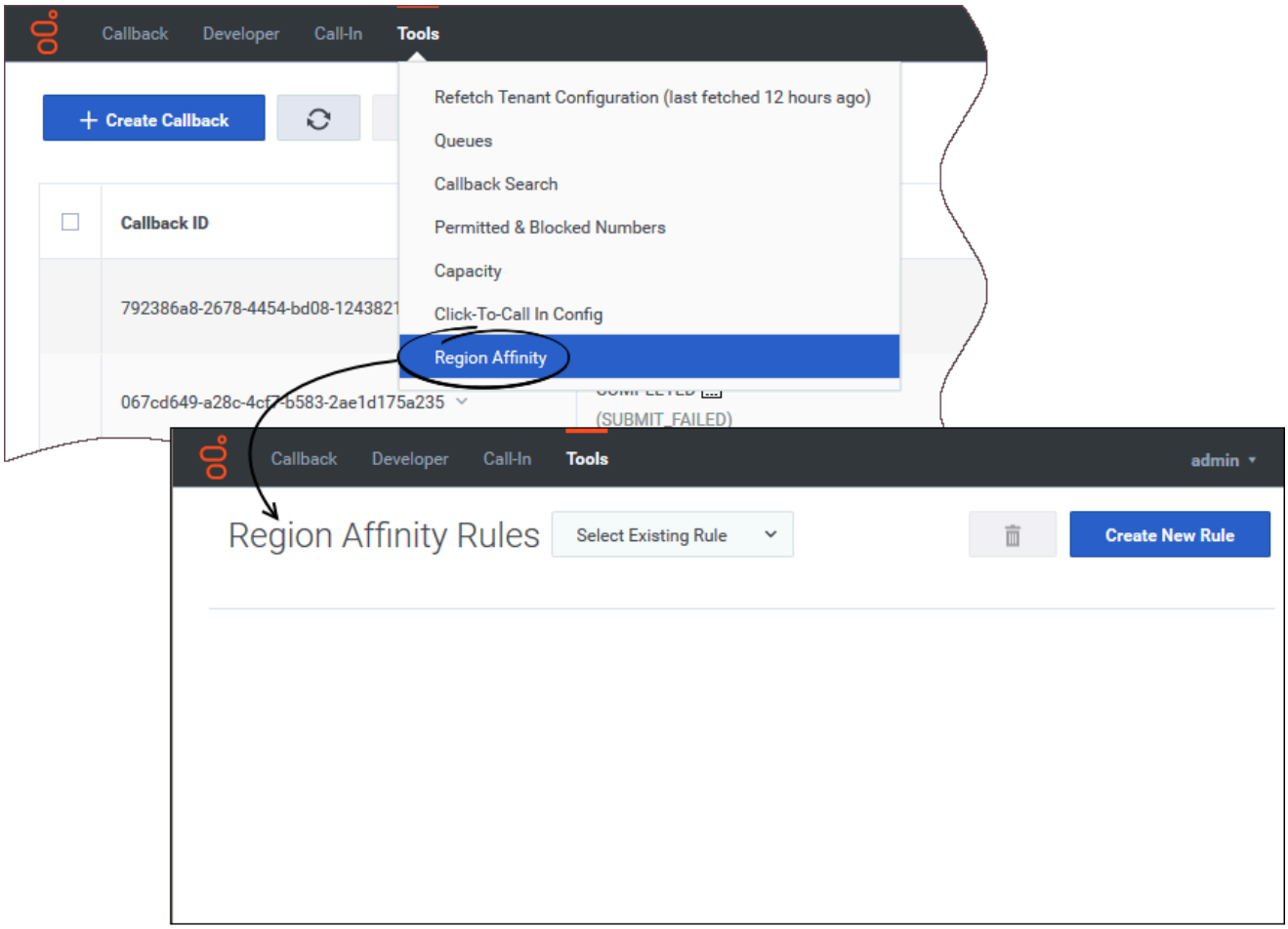
- 1 Creating a Region Affinity rule
  - 1.1 Specifying the order in which to connect to regional nodes
- 2 Assigning a Region Affinity rule to a callback queue

-
- Administrator

Genesys has regional names for the locations at which there are *nodes* (servers with relevant applications) that manage the interactions for callback virtual queues. You use a Region Affinity rule to specify the regional nodes that you prefer to use and to control the order in which Callback attempts to connect to the available nodes. Region Affinity rules configuration is an advanced feature; you must consult with your Genesys representative before using this feature.

Related documentation:

-

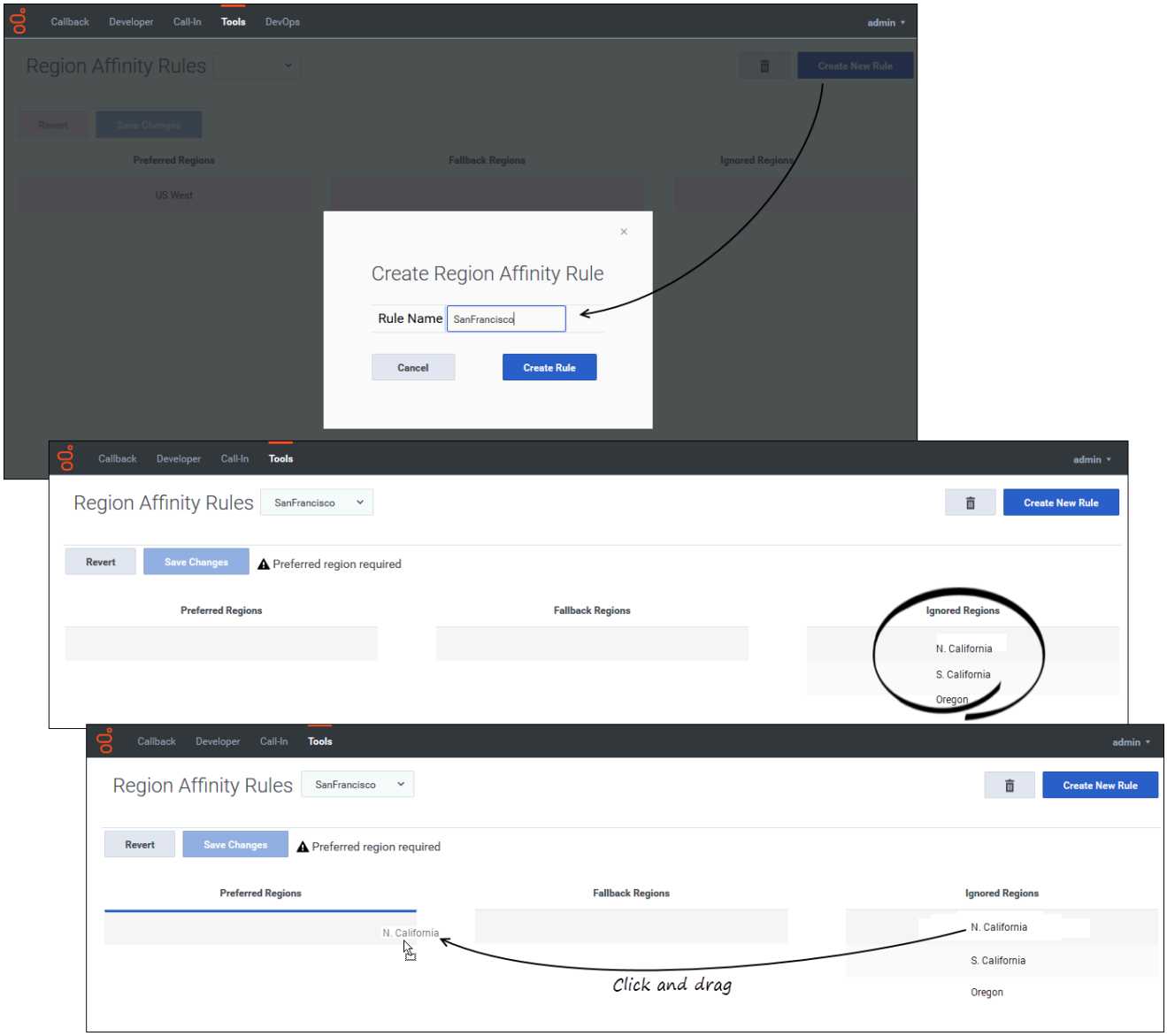


> **Warning**
> Region Affinity rules configuration is an advanced feature that requires an understanding of region configuration and processes that are typically managed internally by Genesys personnel. Do not use this feature unless you have consulted with your Genesys representative.

Use the **Tools** > **Region Affinity** page to create Region Affinity rules. To create Region Affinity rules, you must be a member of the Callback Administrator role.

Genesys has regional names for the locations at which there are *nodes* (servers with relevant applications) that manage the interactions for callback virtual queues. You use a Region Affinity rule to specify the regional nodes that you prefer to use and to control the order in which Callback attempts to connect to the available nodes. When there is activity on a queue to which a Region Affinity rule is assigned, Callback tries to connect to a node in the region designated as "preferred" first. Depending on the health status of nodes in the preferred region, Callback might fall back to other regions in the order of configuration until it finds a viable node.For information about how to configure regions in the order in which you want to attempt connection, see Specifying the order in which to connect to regional nodes.

Creating a Region Affinity rule



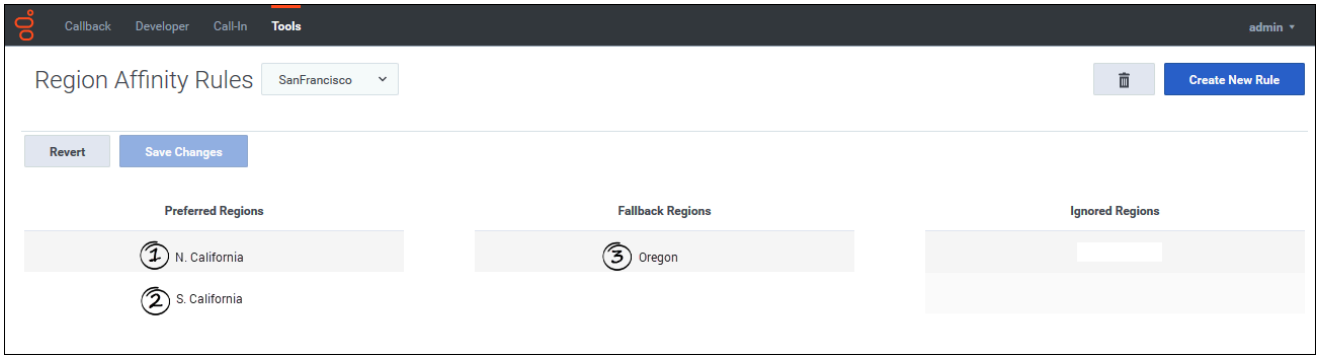
When you create a new Region Affinity rule, all of the regions that are available for your use are initially listed under the **Ignored Regions** column of the rule. To configure the rule, drag region names from the **Ignored Regions** column to the **Preferred Regions** and **Fallback Regions** columns. It is not mandatory to have a region in the **Fallback Regions** column and you can leave some regions in the **Ignored Regions** column. To save a rule, though, there must be at least one region in the **Preferred Regions** column. If there is a region that you want to use, but it does not show up on the **Region Affinity Rules** page in the UI, contact your Genesys Representative.

Specifying the order in which to connect to regional nodes

When handling queue interactions, Callback always tries to connect to a node in the region listed first in the Preferred Regions column. On occasion, there might be some reason why Callback cannot connect to that node. When this happens, Callback attempts to connect to a node in another region in the following order:

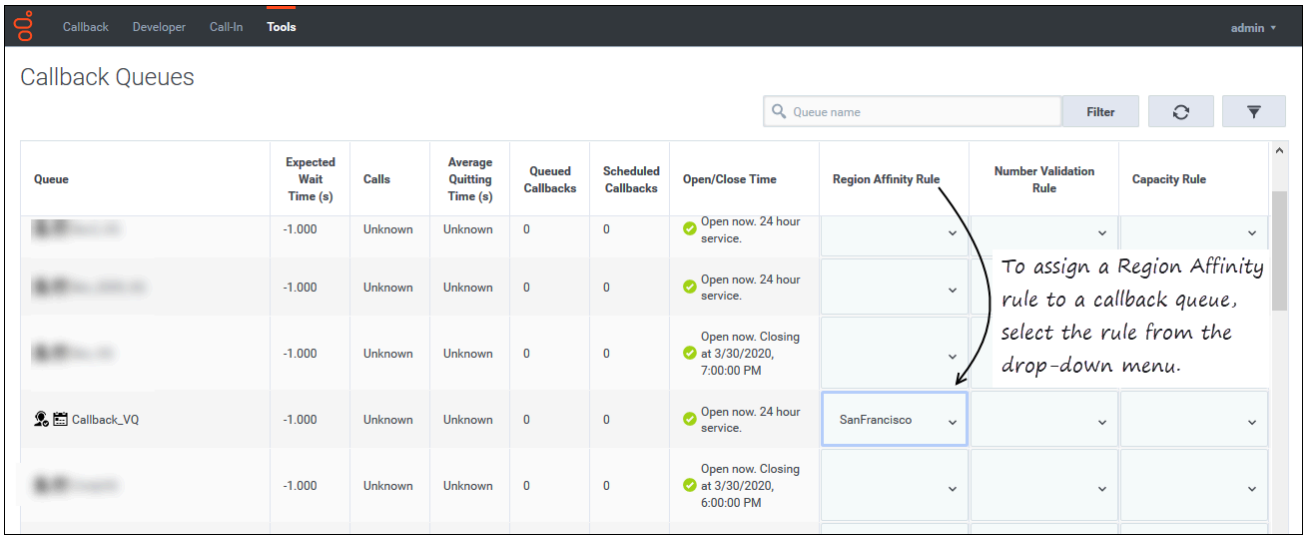
- If there are additional nodes specified in the Preferred Regions column, then Callback tries to connect to a node in the second region listed in the Preferred Regions column, and so on until it has attempted connection to every region – in descending order – in the Preferred Regions column.
- If Callback cannot connect to any node in the preferred regions, then it tries to connect to a node in the region listed first in the Fallback Regions column. If that fails, then Callback tries to connect to a node in the second region listed in the Fallback Regions column, and so on until it has attempted connection to every region – in descending order – in the Fallback Regions column.

With this in mind, you can configure your Region Affinity rule to specify the precise order in which you want Callback to attempt connection to the other available regions. For example, if the following Region Affinity rule is applied to a callback queue, then Callback will always try to connect to the N. California region first, then the S. California region, and if both of those connections fail, then it tries to connect to the Oregon region.

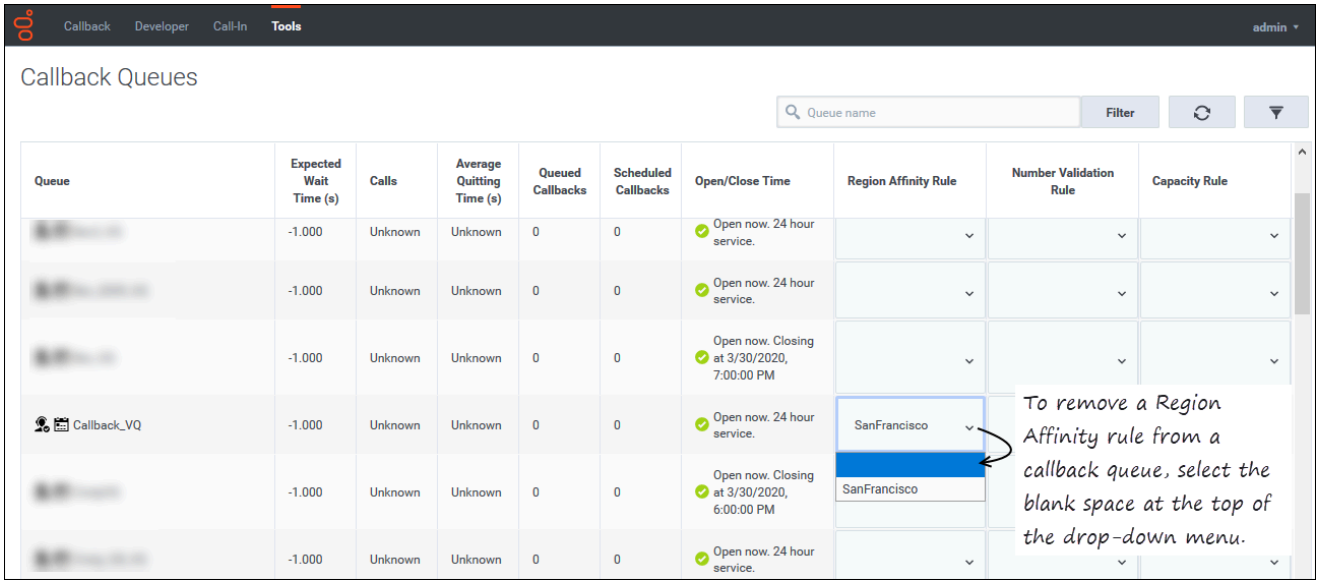


Assigning a Region Affinity rule to a callback queue

To use a Region Affinity rule to specify the regional node at which to manage interactions on a given queue, you must assign the rule to the queue on the **Callback Queues** tab in the UI. Remember that access to queues is subject to line-of-business filtering where role-based access control is enabled. A Region Affinity rule is in effect on a queue as soon as you assign the rule to the queue. You can assign a Region Affinity rule to multiple queues, but you cannot assign more than one rule to a queue.



If you want to delete a Region Affinity rule from the system, you'll first need to remove the rule from all queues to which it is assigned.



Monitor callback queues

Contents

- 1 Viewing the list of configured queues
- 2 Configuring rules for queues

-
- Administrator

Use the **Callback Queues** page in the Callback UI to find information about each configured queue and to add capacity or country and number validation rules to queues.

Related documentation:

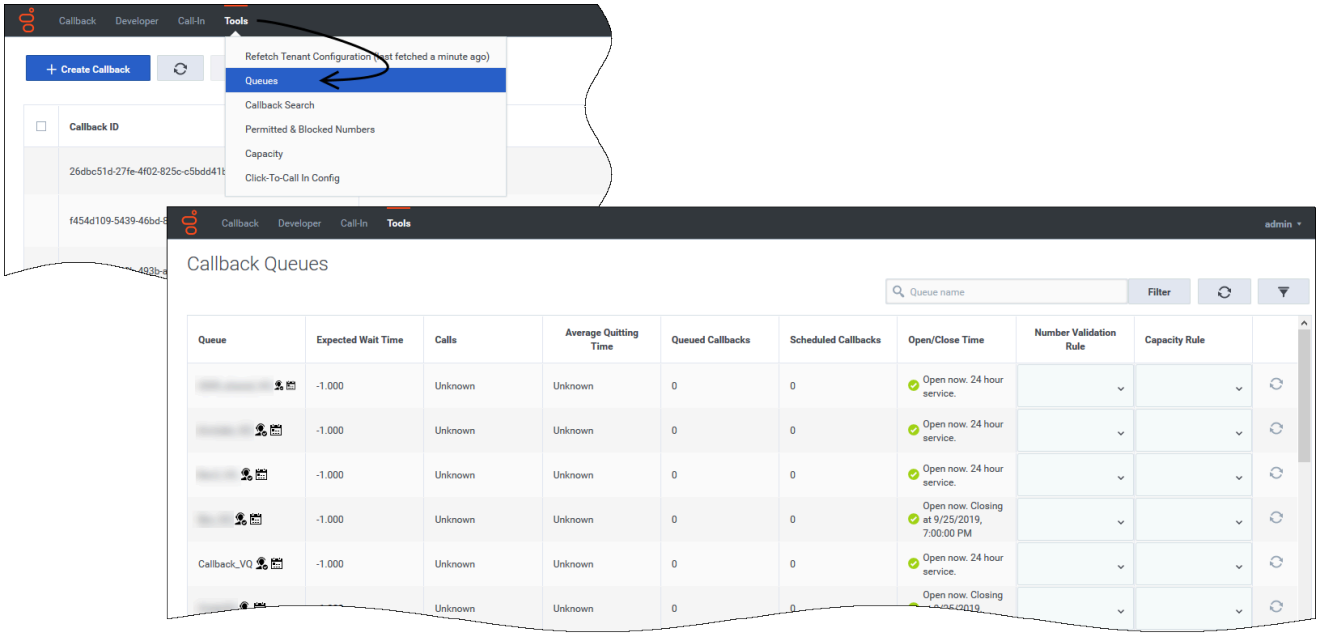
-

You open the **Callback Queues** page from the Tools tab at the top of the **Callback** UI.

For your reference, in addition to providing access to the **Callback Queues** page, the **Tools** tab includes the following menu options:

- You can search for a specific interaction within the callback records.
- You can configure rules that will be assigned to queues. For example, you might want to limit the countries to which you allow callbacks or limit the number of scheduled callbacks that you allow within a certain time interval.
- During the process of migrating from an older version of Callback, it might be helpful to immediately reflect recent changes to Tenant configuration, performed in Platform Administration or Agent Setup, in your Callback UI or to find out when the Tenant configuration data was last refreshed. Use the **Tools** > **Refetch Tenant Configuration** menu option for this purpose. Once you are fully migrated to the latest version of Callback, there is no data in the UI that will be impacted by selecting this menu option.

Viewing the list of configured queues



The **Tools** > **Queues** menu option opens a page that shows you the list of configured virtual queues. You might see the term *service* or *service name* in callback-related applications, Widgets, APIs, and in the Callback UI. A service, in this context, is a virtual queue. The service name, therefore, refers to the name of the virtual queue. For example, on the **Callback** page in the UI, the **Service Name** column identifies, by name, the virtual queue associated with each callback.

The **Callback Queues** page includes information for each queue such as Estimated Wait Time (EWT) and the number of callbacks in queue.

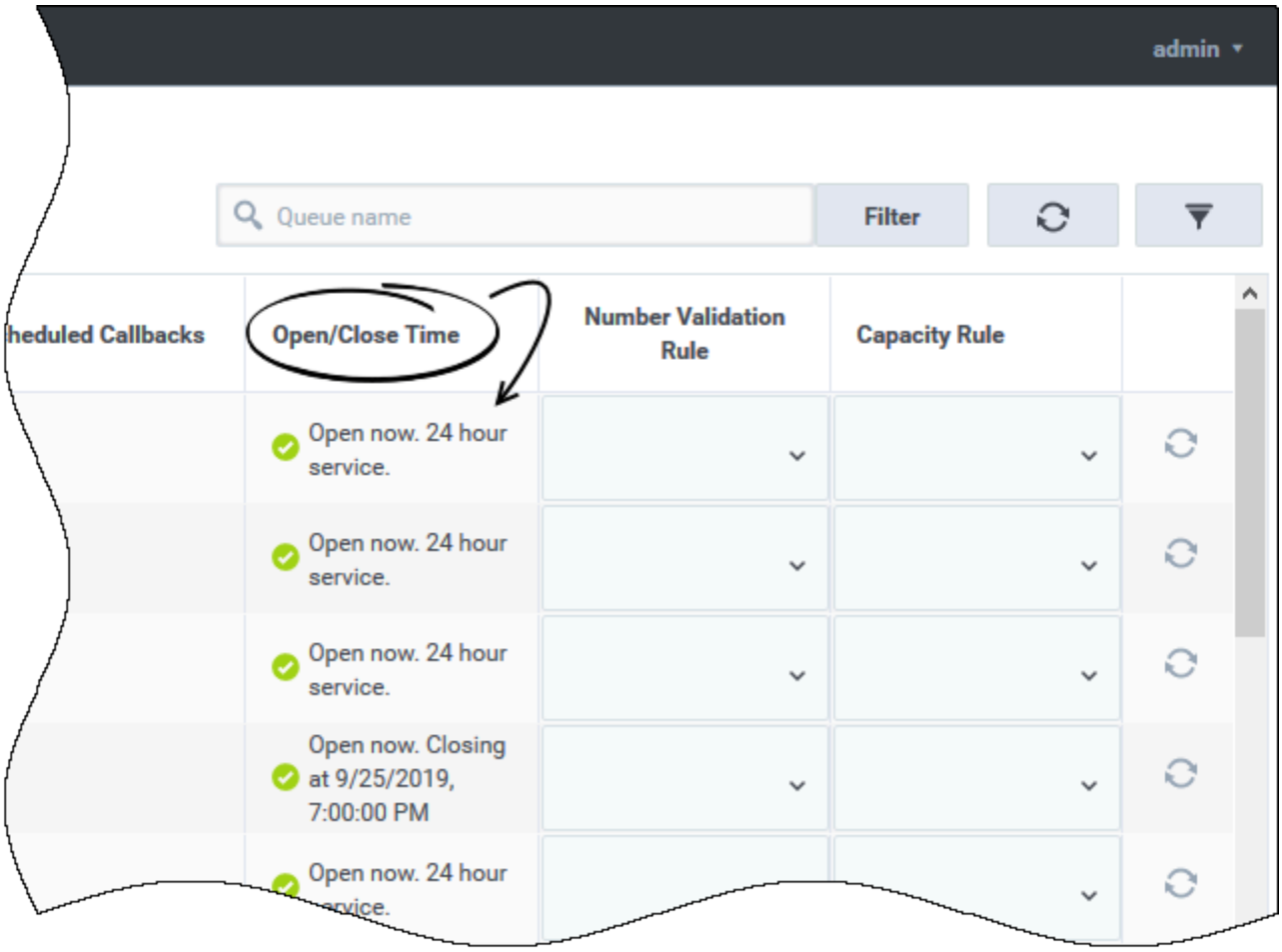
> **Tip**
>
> On the **Callback Queues** page, the Estimated Wait Time is the EWT used for callback calculation; it might not always match the EWT returned by the EWT API.

Icons beside each queue's name indicate whether the queue accepts Immediate ( ) or Scheduled ( ) callbacks or both.

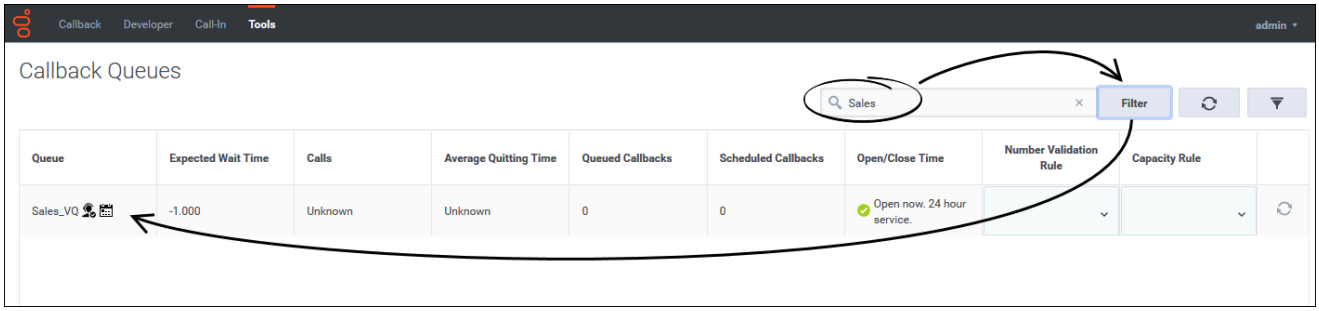
Queues are displayed in pages of 50 items. If `Unknown` values are displayed for a queue, it means that the queue cannot return data for some reason; for example, the queue might be inactive.

The **Callback Queues** page pulls data when you first open the page, but does not continually check for new data. To update the data for a specific queue, click **Refresh** for that queue. When you click **Refresh** at the top of the page, it refreshes the data for every queue.

> **Tip**
> If you provision a virtual queue for callbacks in Designer while you are logged in to the Callback UI, be aware that the new queue might not display on the Callback Queues page until you log out and log in again because Callback fetches virtual queue permissions data only at login. In other words, Callback determines which virtual queues you have access to when you are logging in, and only then.



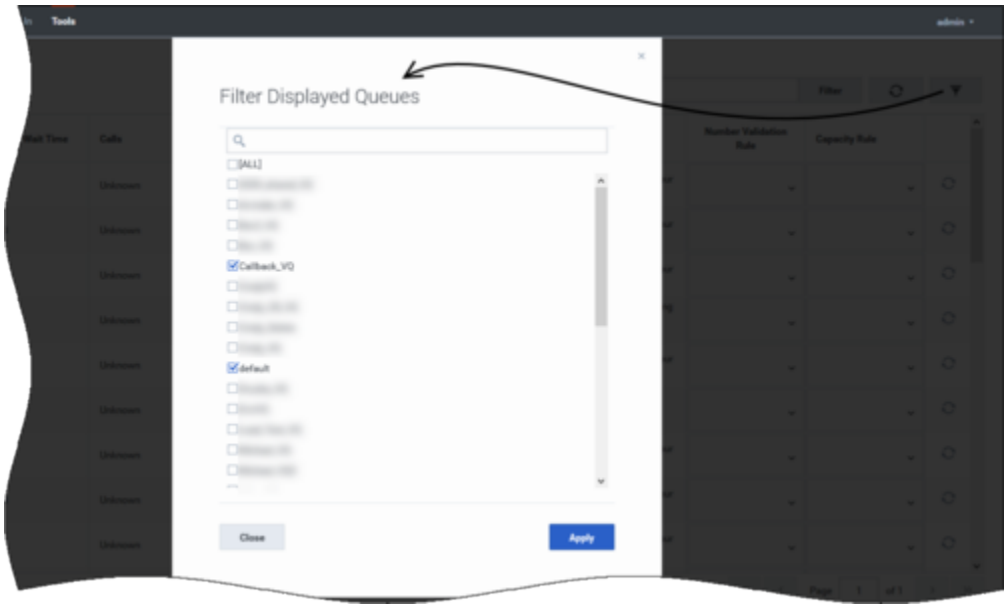
The **Callback Queues** page provides information about Open and Close times for each queue. The time displayed is based on the time zone configured for the device that you are using to access the Callback UI.



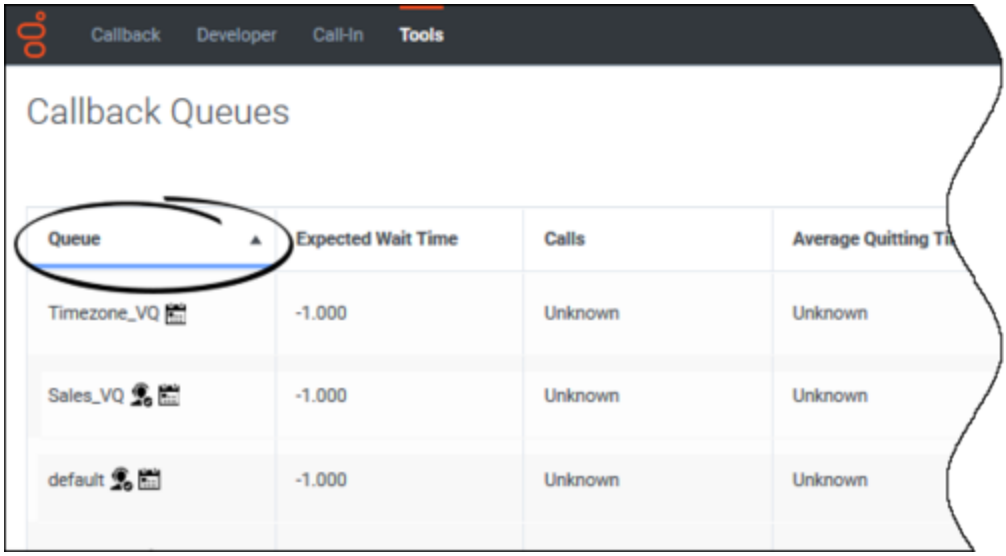
On the **Callback Queues** page, it is sometimes helpful to reduce the list to only one queue or a few queues for which you want to find information.

To filter the list, enter a queue name or part of a queue name in the field at the top of the page and click **Filter**. Using this method, you can filter the list of queues using only one criterion at a time. For example, let's say that you enter Sales in the field and click **Filter**. Any queue that includes Sales in its name will display. If you clear Sales from the field and filter on the word Service, then all queues that have Service in their name display, replacing the list of Sales-related queues.

Alternatively, when the specific queues that you want to view on the **Callback Queues** page have dissimilar names, use the **Filter Displayed Queues** dialog. From the list of virtual queues in the dialog, select those that you want to display. If you have a lot of queues, use the filter at the top of the dialog to help you find the ones that you are looking for.



In addition to filtering, you can also click the **Queue** column heading to sort data based on that column.



Configuring rules for queues

The **Tools** tab includes menu options to configure the following types of rules, which you can then assign to queues:

- Permitted and blocked numbers and countries: Number and country validation rules. These rules specify the countries to which you allow callbacks and can include a list of "blocklist" numbers (blocked number patterns) within those countries.
- Capacity: Callback capacity rules. These rules specify the maximum number of scheduled callbacks that are allowed within each time slot for a week.

Provisioning Push Notifications

Contents

- 1 Callback Push Notifications
- 2 Firebase Cloud Messaging Push Notifications
  - 2.1 Before you start
- 3 Using the Push Notification page
- 4 Register your Firebase Credentials with Callback
  - 4.1 Project name
  - 4.2 Client email
  - 4.3 Private key
- 5 Test your credentials configuration
- 6 Displaying the Firebase credentials on the Push Notification page

-
- Administrator

The **Push Notification** page in the Callback UI helps you to integrate Push Notification support into your apps. In the Callback UI, you implement Push Notifications on the **Developer** > **Credential Management** > **Push Notification** tab.

Related documentation:

-

Push Notifications are messages that are proactively sent from an app to a known, permitted client (either a mobile device or a web browser). Mobile device users are familiar with Push Notification behavior within their favorite apps.

If you have the Click-To-Call feature on your website or mobile app, the system can interact with your customers by pushing notifications about the callback to a customer's device. You can design your mobile app to offer follow-up actions such as the ability to confirm or cancel the callback. For more information about provisioning the Click-To-Call-In feature in Callback, see Provisioning the Click-to-Call-In scenario.

Only users who are members of the Callback Developer or Administrator Roles can access the **Push Notification** page in the Callback UI. For more information about controlling user access to Callback tabs and features, see Controlling User Access. As long as you have sufficient permissions to view the **Push Notification** page, then you have full access to use all of its features.

Callback Push Notifications

When Push parameters are included in a Callback Create API request on a standard callback virtual queue, the system can send Push Notifications for the following events:

- CALLBACK_UPCOMING—When the callback's Estimated Wait Time (EWT) is below the configured threshold for the queue (configured under the **Push Notification Threshold (minutes)** setting in the CALLBACK_SETTINGS data table), the CALLBACK_UPCOMING event sends a Push Notification to let the consumer know that they can expect a phone call soon. This event is sent only once. In other words, if the EWT increases above the threshold after the notification is sent and then drops below the threshold again, the CALLBACK_UPCOMING Push Notification is not sent again. The system might not send a Push Notification for the CALLBACK_UPCOMING event in the following situations:
  - When the EWT is already low (below the configured threshold) when the Callback Create API request arrives.
  - When the system determines that there is not enough time to complete the callback because the callback will be purged or the end of the business will occur before the callback is at the top of the queue.
- CALLBACK_READY—Used only with the Click-To-Call-In (Delayed) scenario. When a consumer's callback

request reaches the front of the queue, the CALLBACK_READY event sends a Push Notification to alert the consumer. If the consumer accepts the callback, the mobile app sends a Call-In Create API request. The system immediately sends another Push Notification with the information that the user needs to call in to the contact center. Note that the callback "acceptance" is a function of your mobile app's configuration. For example, your mobile app developer can configure your mobile app to make the call-in request automatically as soon as it receives the CALLBACK_READY notification. Alternatively, the app could require the consumer to make the call-in request manually by clicking a button to acknowledge receipt of the notification.

- CALLBACK_PURGED—This Push Notification is sent if the callback has been purged from the system and the callback was not rescheduled.
- CALLBACK_ATTEMPT_FAILED—This event occurs when the system makes the outbound call for a callback, but the call fails to connect to the consumer for some reason. The system makes another outbound call attempt later. In the CALLBACK_SETTINGS data table, you can configure the number of times that the system can make outbound calls for each Callback request before declaring failure. This Push Notification is not used with the Click-To-Call-In (Delayed) scenario because there are no outbound calls associated with that scenario.
- CALLBACK_FINAL_ATTEMPT_FAILED—This event occurs when the system makes the final outbound call associated with a Callback request and the call fails to connect to the consumer. In the CALLBACK_SETTINGS data table, you can configure the number of times that the system can make outbound calls for each Callback request before declaring failure. This Push Notification is not used with the Click-To-Call-In (Delayed) scenario because there are no outbound calls associated with that scenario.

Firebase Cloud Messaging Push Notifications

Genesys Callback currently supports Google Firebase Cloud Messaging (FCM) Push Notifications. Using the Callback UI, you can integrate Firebase Cloud Messaging into your web or mobile applications. For more information about FCM, see the Google Firebase Cloud Messaging documentation. You use the **Push Notification** page in the **Developer** tab to register your Firebase credentials with Callback and to test that the credentials are valid.

Before you start

Before you start working with the **Push Notification** page on the Callback **Developer** tab, ensure that you have a Google Firebase account. After you register your app with a Firebase project, you use the following Firebase data to register the app with Callback for Push Notification functionality:

- The Firebase project name that is associated with the app that you are registering with Callback.
- The client email address that is associated with the project.
- The private key.
- The application ID.
- The public API key.
- The sender ID.

> **Important**
> You might need to update your existing Firebase projects because of changes made to FCM in September 2019. For more information, see the entry for FCM in the Firebase JavaScript SDK Release Notes, under Version 7.0.0 - September 26, 2019.

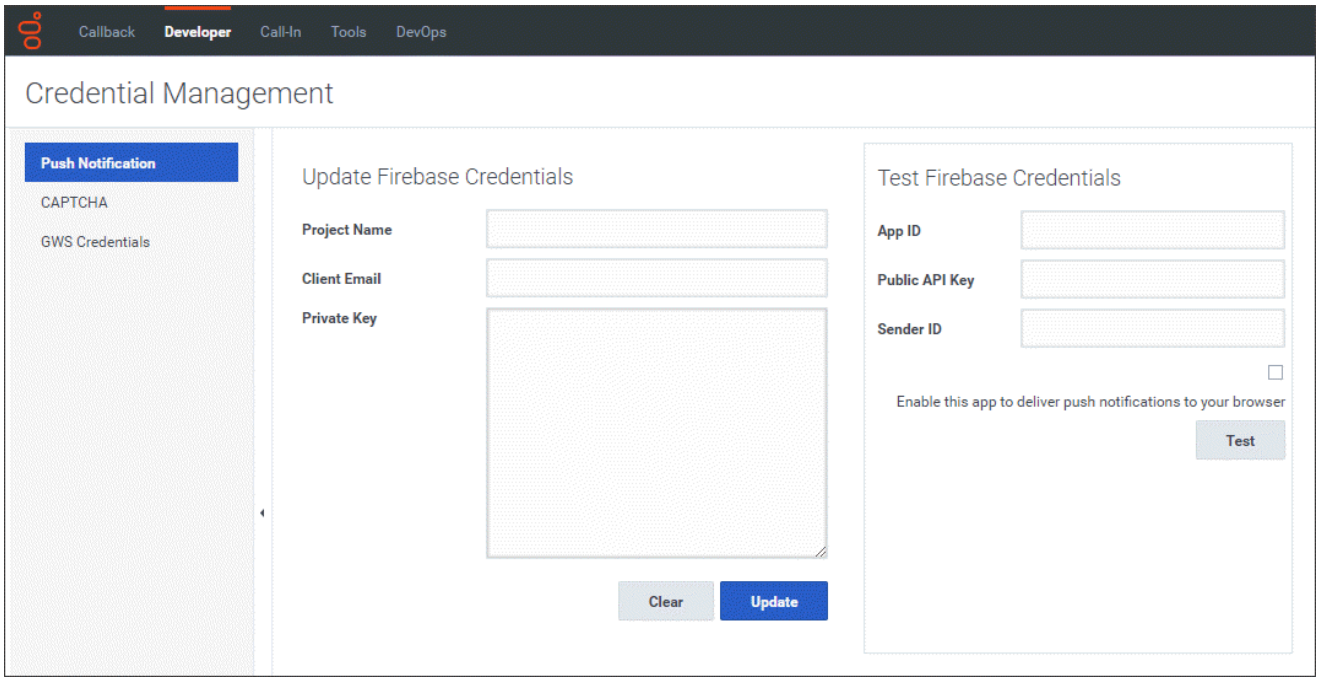
Using the Push Notification page

You can perform the following activities on the **Push Notification** page:

- Provision and update your Firebase credentials in Callback.
- Determine if your Firebase credentials for an app are already configured in Callback.
- Perform a test to verify that the Firebase credentials that you entered are correct and configured in Callback and, by extension, that the Push Notifications are working as intended.

To configure the Firebase credentials in Callback, you enter and update those credentials on the **Update Firebase Credentials** pane of the **Push Notification** page. You use the **Test Firebase Credentials** pane to verify that web Push Notifications are working in your environment. If the Firebase credentials that you provisioned on the **Update Firebase Credentials** pane are correct and valid, then – when you test the credentials – the **Push Notification** page returns a message that your Firebase credentials were successfully configured.

Register your Firebase Credentials with Callback



To implement the Firebase Cloud Messaging functionality, you must provide the following information in the **Update Firebase Credentials** pane on the **Push Notification** page:

- Project name
- Client email address
- Private key

After you click the **Update** button, a message appears in the bottom right-hand corner of the page to let you know that your credentials were successfully updated or if there was an issue.

Genesys recommends that you also test your Firebase credentials to make sure that the credentials have been successfully configured in Callback. To run the test, you require the Firebase Application ID, the Public API key, and the Sender ID associated with the project shown in the **Update Firebase Credentials** pane.

Project name

This name is a user-friendly name that you configure for the Firebase project. For example, `fcm-client-11959`. Once configured, you can find the Firebase project name (or Project ID) in the **Project settings** > **General** tab on the Firebase console. On the **Push Notification** page, you must enter the Firebase project name *exactly* as it is configured in Firebase.

Client email

The email address associated with the Firebase project. On the Firebase console, this is the service

account. In the file that contains the private key, the account is called the client email address. For example, a client email address could be `firebase-adminsdk-rkbys@fcm-client-11959.iam.gserviceaccount.com`. Once configured, you can find the service or client email account on the **Project settings** > **Service accounts** tab on the Firebase console or within the file that contains your private key.

Private key

An application uses the private key to access the Firebase API. You generate a file that contains the private key for your app on the **Project settings** > **Service accounts** tab of the Firebase console. Generating the file downloads it to your local machine. The private key is a very long character string and is only available in the downloaded file.

If, for any reason, you need to generate a new private key for an application that was previously configured in Callback, then you can do so on the Firebase console, but then you must update the credentials in Callback using the **Update Firebase Credentials** pane on the **Push Notification** page. Remember to test the new credentials as well.

Test your credentials configuration

If you are registering new Firebase credentials with Callback or updating credentials, then always test those credentials to verify that the Firebase and Callback configuration is valid. You require the Firebase App ID, Public API key, and Sender ID to test your credentials. To find that information, log into your Firebase project, open the list of apps, and click the Settings icon beside the app that you're registering with Callback. Scroll to the bottom of the page of settings until you locate the App ID, the Sender ID, and the API key.

You must check the **Enable this app to deliver push notifications to your browser** box to run the test.

The **Push Notification** page returns either a "successful configuration" message or a "failure" message.

If you receive a failure message, use the following process to recover and try again:

1. Check that the App ID, the Public API key, and the Sender ID that you entered on the **Update Firebase Credentials** pane match the values on the Firebase console for the app that you are registering with Callback. Make any necessary adjustments and then run the test again.
2. If the App ID, the Public API key, and the Sender ID match and you don't receive a Push Notification, then refresh the browser and run the test again.
3. If you don't receive a Push Notification, then navigate to the settings for your browser. Ensure that the browser allows notifications from your Genesys Engagement Service (GES)/Callback host. If you need help finding the setting, check the documentation for your browser. If your browser is blocking notifications, then alter the setting to enable notifications. Run the test again.
4. If you don't receive a Push Notification, then refresh your browser again, review your FCM credentials once more, and run the test again.

Displaying the Firebase credentials on the Push Notification page

If you refresh the **Push Notification** page or return to it some time after you successfully registered your Firebase credentials, you see only a few characters of the client email address and the private key. The obfuscation is for security purposes. Once configured, it is impossible to see the complete entries again on the **Push Notification** page. The Client email address and Private key sections on this page explain where you can find the full credentials again, if required.

Provisioning CAPTCHA

Contents

- 1 Important information about enabling reCAPTCHA on Callback requests
- 2 Provisioning Google reCAPTCHA for Callback
- 3 Testing reCAPTCHA integration with Callback

-
- Administrator
- Developer

If you have a website that includes a callback feature, then you can provision and use a CAPTCHA widget to verify that a human is requesting the callback. Currently, Callback supports integration with only one CAPTCHA widget: Google reCAPTCHA. This page describes how to enable the use of reCAPTCHA with Callback.

Related documentation:

-

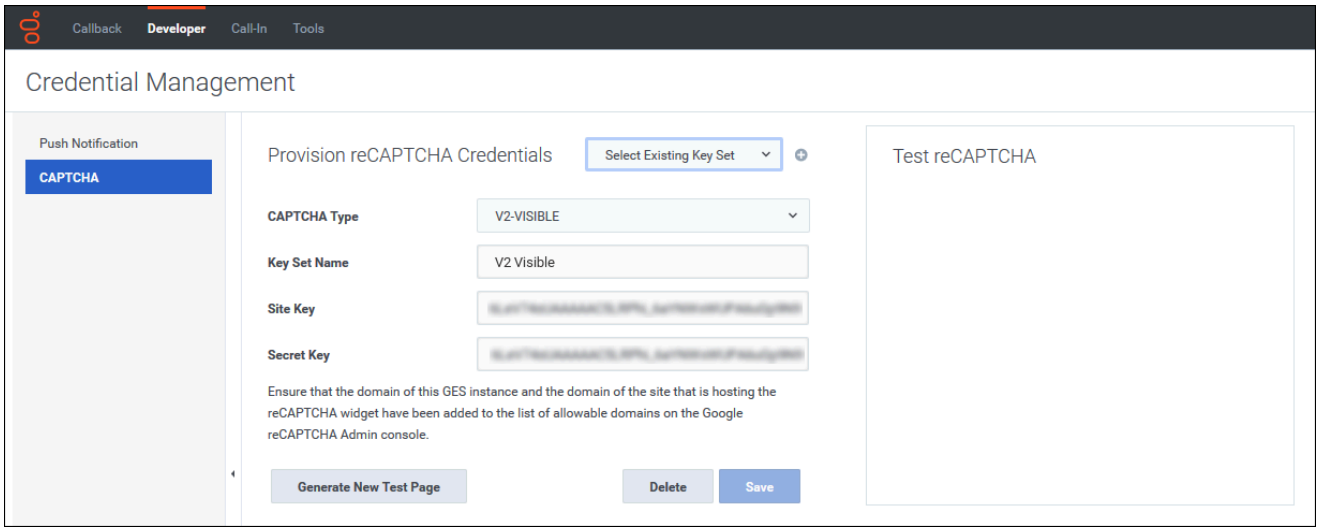
Important information about enabling reCAPTCHA on Callback requests

Integrating reCAPTCHA into your Callback API operations means that reCAPTCHA validation will be required with *all* of the Callback API operations. For example, if you have both a website and a mobile app, you cannot implement reCAPTCHA on callback requests for only your website – it must be built into your mobile app as well. Also, this implementation of reCAPTCHA is not integrated with the Genesys Callback Widget. If you are using the Callback Widget, then Genesys recommends that you avoid enabling this reCAPTCHA capability.

Provisioning Google reCAPTCHA for Callback

Callback currently supports Google reCAPTCHA. To provision reCAPTCHA for Callback, you must have a Google account. If you have not yet done so, go to the reCAPTCHA Admin console to register your site and to specify the reCAPTCHA version that you want to use. Callback supports reCAPTCHA v3, reCAPTCHA v2 Checkbox (Visible), and reCAPTCHA v2 (Invisible). Review the Google reCAPTCHA documentation; it describes the three versions of reCAPTCHA and contains information to help you to register your site on the Admin console.

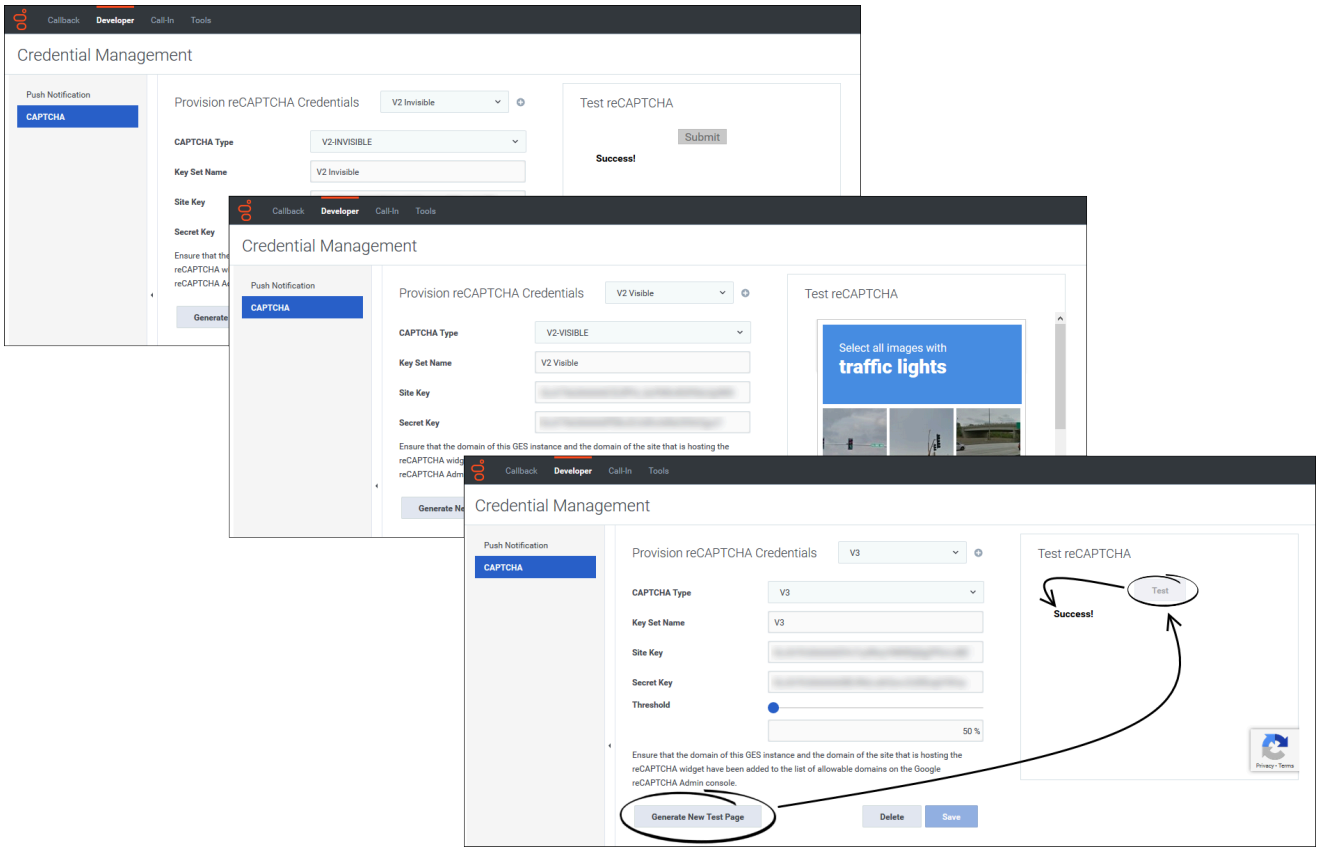
On the Google reCAPTCHA Admin console, it is important that you add the domain of your Callback instance and the domain of the site that is hosting the reCAPTCHA widget to the list of allowable domains.



After you have registered your site on the Google reCAPTCHA Admin console, you must register your reCAPTCHA credentials with Callback on the **Developer** > **Credential Management** > **CAPTCHA** tab. The CAPTCHA type, site key, and secret key that you specify must match the settings for the reCAPTCHA widget that you provisioned in the Google Admin console.

Specify a name (the **Key Set Name**) to identify the reCAPTCHA widget and its associated credentials. It is helpful to use a descriptive name. For example, you might enter a name that reflects the type of reCAPTCHA widget that you're using (`V2 Visible`). The **Key Set Name** is used only in Callback; it is not specified or used in the Google reCAPTCHA Admin console.

Testing reCAPTCHA integration with Callback



After you save your reCAPTCHA credentials on the **CAPTCHA** page in Callback, be sure to test them. When you click **Generate New Test Page**, a testing widget opens on the page. The widget will vary slightly depending on the type of reCAPTCHA that you're provisioning. You might have to check a box or otherwise interact with the widget, but you will be presented with a **Submit** or **Test** button at some point. If your credentials were correctly configured and you pass the reCAPTCHA validation, then the widget returns a Success! message. If it fails, check your credentials or – if you are using the V3-type reCAPTCHA – try adjusting the threshold, and then try again.

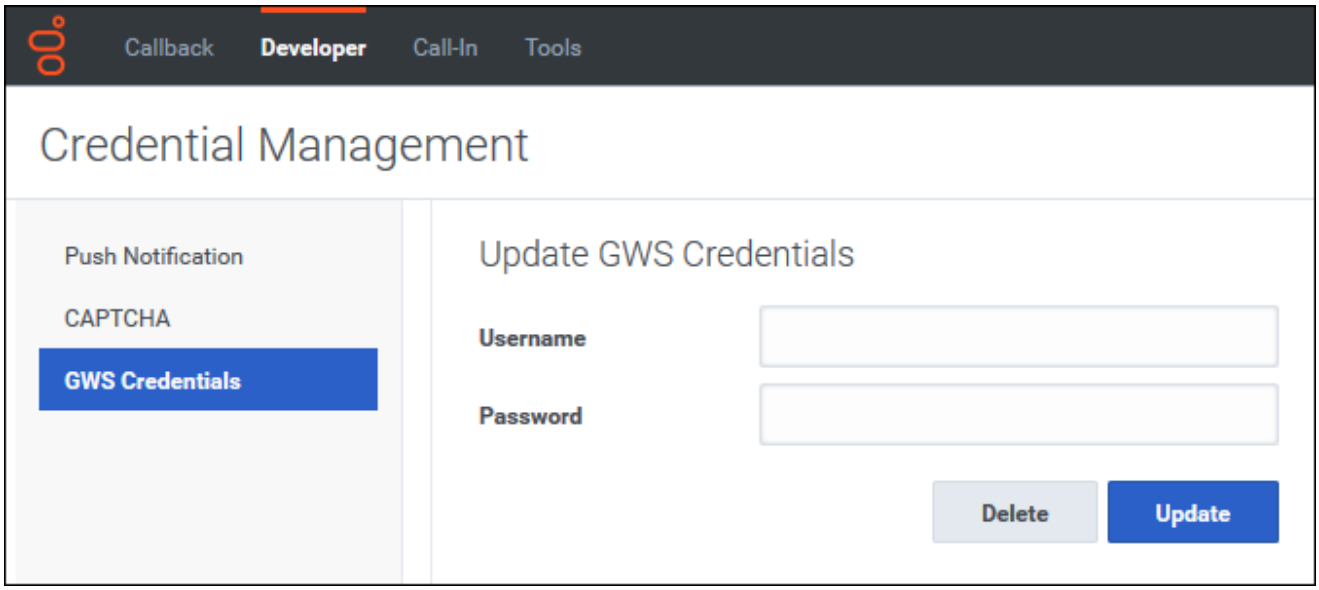
Registering for access to statistics

-
- Developer

Register your credentials to use Callback's Statistics API.

Related documentation:

-



To retrieve GWS statistics information through Callback's Statistics API, you must first register your credentials on the **Credential Management** > **GWS Credentials** tab.

For additional information about the Callback APIs, see Genesys Multicloud CX REST APIs and tutorials for Callback.

Provisioning the Click-to-Call-In scenario

Contents

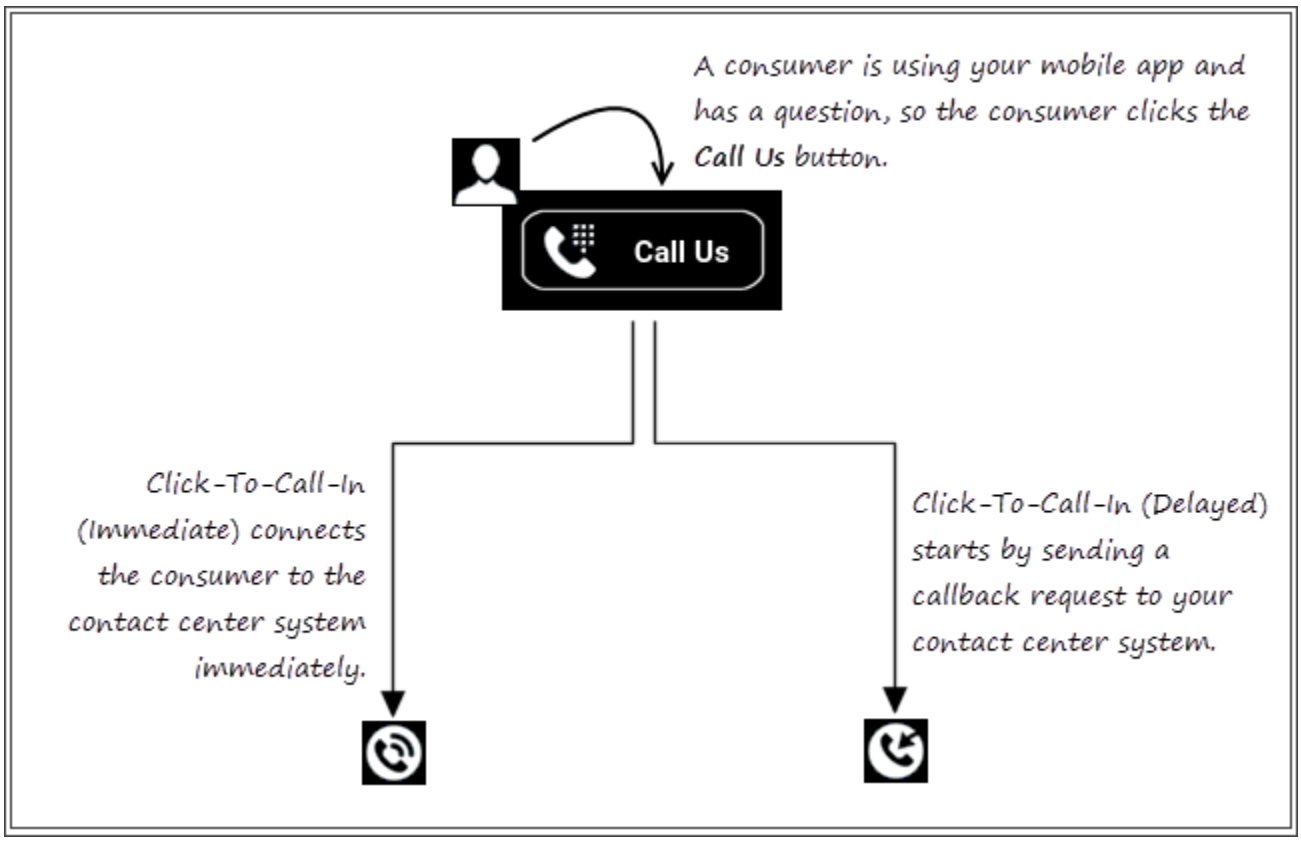
- 1 Click-To-Call-In (Delayed) Push Notifications
- 2 Provisioning the Click-To-Call-In (Immediate) scenario
  - 2.1 Overview of the provisioning steps
- 3 Provisioning the Click-To-Call-In (Delayed) scenario
  - 3.1 Before you start
  - 3.2 Overview of the provisioning steps
  - 3.3 Provision the Click-To-Call-In Delayed callback application
  - 3.4 Configure settings for the Click-To-Call-In Delayed Virtual Queue in the data table
  - 3.5 Provision the Click-to-Call-In Match application in Designer

-
- Administrator

This article describes how to provision Callback's Click-To-Call-In feature, which lets consumers initiate a call to your contact center directly from your mobile app.

Related documentation:

-



You can implement the Click-To-Call-In functionality so consumers tap a Click-To-Call button or link in your mobile app and the call is dialed immediately. In this implementation, consumers are placed in queue like any other inbound call. Alternatively, you can configure the feature to delay the call until the system is sure that there is an agent available who has the correct skills to assist the consumer. In both scenarios – immediate or delayed connection to your contact center – you must configure Click-to-Call-In groups in the Callback UI. If you

configure the delayed scenario, then you must also provision Push Notifications.

Click-To-Call-In (Delayed) Push Notifications

Push Notifications are not used with the Click-To-Call-In (Immediate) scenario.

If you use the Click-To-Call-In (Delayed) scenario with your mobile app, then you must provision Push Notifications. Your mobile app sends a callback request when the consumer taps the button or link that you have provisioned for the Click-To-Call-In (Delayed) feature. The callback request includes the Push parameters. For information about the Push Notifications supported with Callback, see Callback Push Notifications. For a description of the Click-To-Call-In (Delayed) scenario, see Click-To-Call-In (Delayed) works.

Provisioning the Click-To-Call-In (Immediate) scenario

To provision the Click-to-Call-In (Immediate) scenario, you will work in both Callback and in Designer. You can do the work in the Callback UI before you do the work in Designer or after. Either way, the configuration in both must be completed before you can test your Click-To-Call-In (Immediate) scenario.

The Click-To-Call-In (Immediate) scenario uses the Genesys Engagement Services (GES) Call-In API.

For information about the callback-related APIs, see Genesys Multicloud CX REST APIs and tutorials for Callback.

Overview of the provisioning steps

You do not require a dedicated virtual queue for the Click-To-Call-In (Immediate) scenario. The consumer is simply calling into your contact center.

You must provision the following for the Click-To-Call-In (Immediate) scenario:

- In the Callback UI, provision the following:
  - Click-To-Call-In groups: Required by the system to determine which telephone number is appropriate for a Click-To-Call-In request. The system also uses the group configuration to determine whether or not an access code is required when the consumer dials in.
- In the Designer UI, provision the following:
  - A Click-to-Call-In Match application: This application must include the Click-to-Call-In Match Block. This Designer application must be of type `default`. During this part of the process, the system attempts to match a caller to a Call-In request based on data received with the request and data that the system provides in the response, such as the number that the caller dialed (the DNIS number), the access code (if required), and the caller's telephone number. If the attempt to match a caller to a Call-In request is successful and your Designer application is configured to route the call when a match is made, then the call is queued on hold like any other call to the contact center.

Provisioning the Click-To-Call-In (Delayed) scenario

To provision the Click-to-Call-In (Delayed) scenario, you will work in both Callback and in Designer. You can do the work in the Callback UI before you do the work in Designer or after. Either way, the configuration in both must be completed before you can test your Click-To-Call-In (Delayed) scenario.

The Click-To-Call-In (Delayed) scenario uses the following Genesys Engagement Services (GES) APIs:

- Callback
- Call-In

For information about callback-related APIs, see Genesys Multicloud CX REST APIs and tutorials for Callback.

Before you start

Before you start provisioning Designer applications for Click-To-Call-In (Delayed), create a virtual queue in Platform Administration for this purpose. You need to add the queue to the CALLBACK_SETTINGS data table and then configure it, so be sure to give it a name that helps to identify it; for example, ClickToCallInDelayed_VQ.

Overview of the provisioning steps

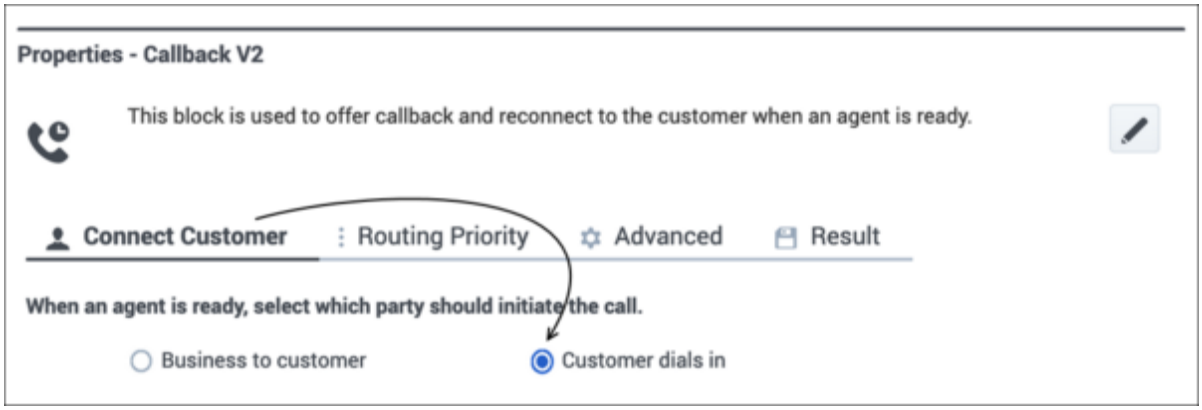
You must provision the following for the Click-To-Call-In (Delayed) scenario:

- In the Callback UI, provision the following:
  - Push Notifications: Required for communicating information about the call-in to the consumer such as when the callback is ready, what number to call and – if required – what access code to enter.
  - Click-to-Call-In groups: Required by the system to determine which telephone number is appropriate for a Click-To-Call-In request. The system also uses the group configuration to determine whether or not an access code is required when the consumer dials in.
- In the Designer UI, provision the following:
  - A Callback application specifically for Click-To-Call-In (Delayed): Used to start the callback process. This Designer application must be of type `Callback`. During this part of the process, the call is queued in the Click-To-Call-In Delayed virtual queue.
  - The Click-To-Call-In Delayed virtual queue in the CALLBACK_SETTINGS data table.
  - A Click-to-Call-In Match application: Used when the customer calls the phone number that the system provided in a Push Notification. This Designer application must be of type `default`. During this part of the process, the system attempts to match the caller to an existing callback record and, if successful, routes the call to an available agent. For more information about the Click-to-Call-In Match block, see Click-to-Call-In Match Block.

Every environment is different and requires its own specific routing logic, error handling, and so on. That means that, in addition to the configuration described in this section, you must consider what customized configuration you require in the Designer applications and in the virtual queue used with the Click-To-Call-In (Delayed) scenario. This section describes only the basic configuration that is required in all environments to provision the Click-To-Call-In (Delayed) scenario in Designer.

Provision the Click-To-Call-In Delayed callback application

1. In Designer, create a new application. Select `Callback` for the type.
2. Add a **Callback V2** block to the **Initialize** phase.
3. In the properties of the **Callback V2** block, select **Customer dials in**, rather than **Business to customer** under the **Connect Customer** tab.



4. In all other regards, configure the application as required for your contact center.
5. Save and publish your Click-To-Call-In (Delayed) Callback application.

Configure settings for the Click-To-Call-In Delayed Virtual Queue in the data table

1. In Designer, navigate to **Business Controls** > **Data Tables** and open the CALLBACK_SETTINGS data table.
2. If you have not yet done so, add the Click-To-Call-In Delayed virtual queue to the table; otherwise, find the queue in the table.
3. Make sure that you configure all of the following settings in the data table for this queue, in addition to any other parameters that you choose to customize for your environment:
   - **Pushed Callback Expiry Time (minutes)**. This setting specifies how long the callback session stays alive after the system sends the initial Push Notification to the consumer. If the callback expires, any subsequent Call-In request related to that callback session will fail.
   - **Push Notification Threshold (minutes)**. This setting specifies when to send the CALLBACK_UPCOMING Push Notification to the consumer. For example, if you specify a value of 1, then when the Estimated Wait Time (EWT) on the queue drops below 1 minute, the system sends a CALLBACK_UPCOMING Push Notification to the consumer if conditions are favorable for the callback to proceed. For detailed information about the CALLBACK_UPCOMING Push Notification, including information about cases in which the system does not send this notice, see Callback Push Notifications.
   - **Call Direction.** This must be set to `USERORIGINATED`.
   - **Callback Application**. Select the `Callback`-type application that you created for the Click-To-Call-In (Delayed) scenario.

     The **Callback Purge Time** and the **Business Hours** that you specify in the CALLBACK_SETTINGS data table for the Click-To-Call-In Delayed virtual queue are applied to this queue just as they are applied to any other queue. Those two settings are relevant when the system must determine if

the callback can be purged or not. In addition, Genesys recommends that you configure the **Skill Expression** for the virtual queue to ensure that the calls are routed to the agents who can best serve the consumers who are calling in.

The callback will be purged at the end of the day based on the specified business hours, regardless of the **Callback Purge Time** status.

4. Save and then publish your updated data table.

Provision the Click-to-Call-In Match application in Designer

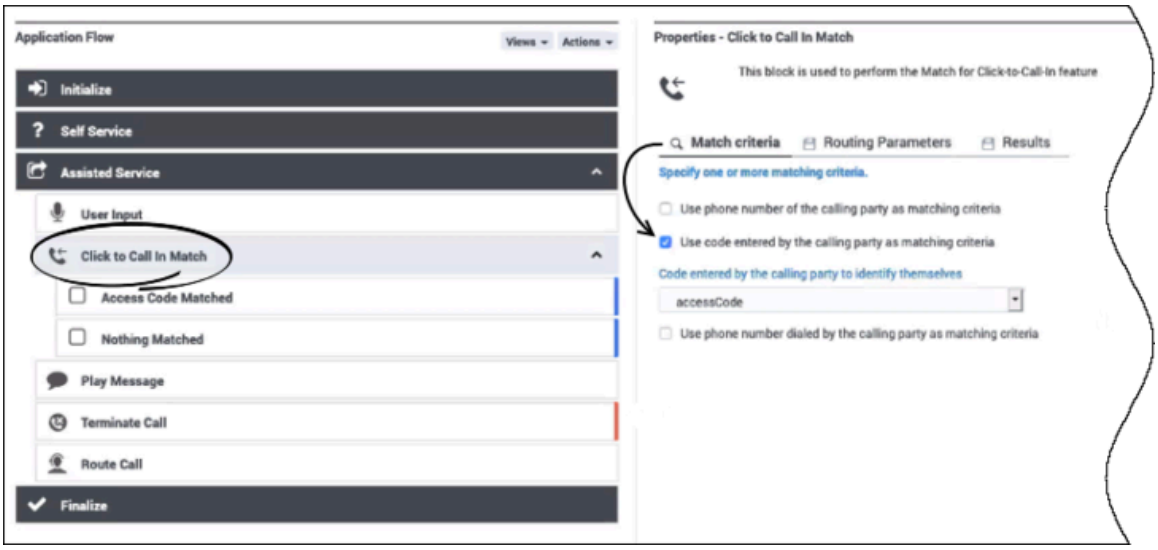
In addition to the `Callback`-type Designer application, you also require a Click-To-Call-In Match application for the Click-To-Call-In (Delayed) scenario. When a Push Notification sends the number and optional access code to the consumer, then the consumer calls in. The Match application matches the caller to the original callback request. Based on the results of the matching attempt, the application either routes the call to an agent (successful match) or the call moves to the next block (no match). The routing is performed by a Route Call block.

The Match application must be a `Default`-type application. It can be used with multiple Click-to-Call-In Delayed virtual queues and multiple Click-to-Call-In Groups.

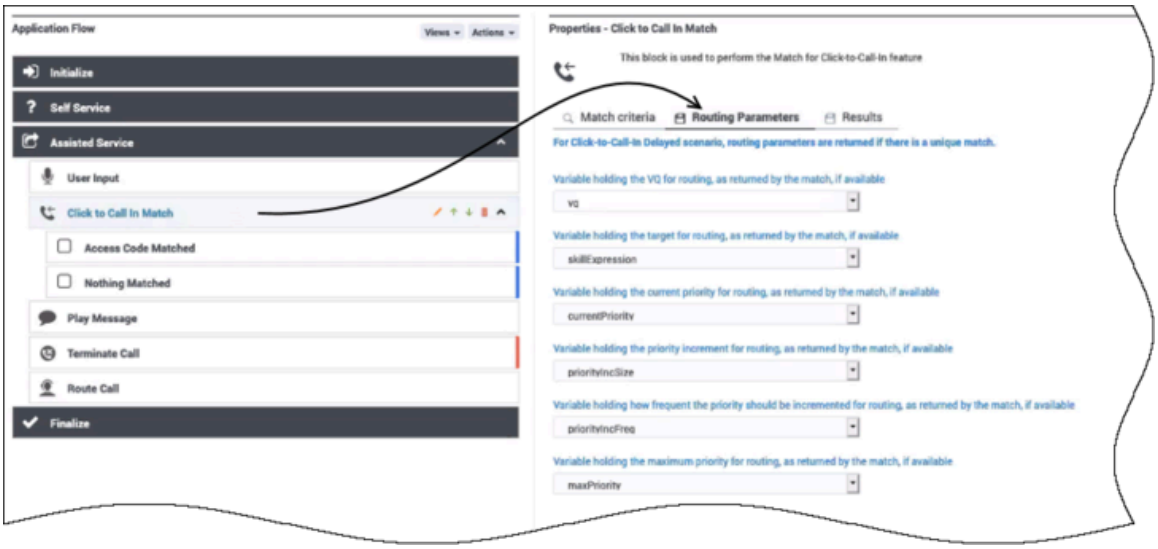
Sample Configuration

For the purposes of the following procedure, let's assume that consumers who are using the Click-To-Call-In feature are prompted to enter an access code. We’ll have the application perform the match only on the access code.

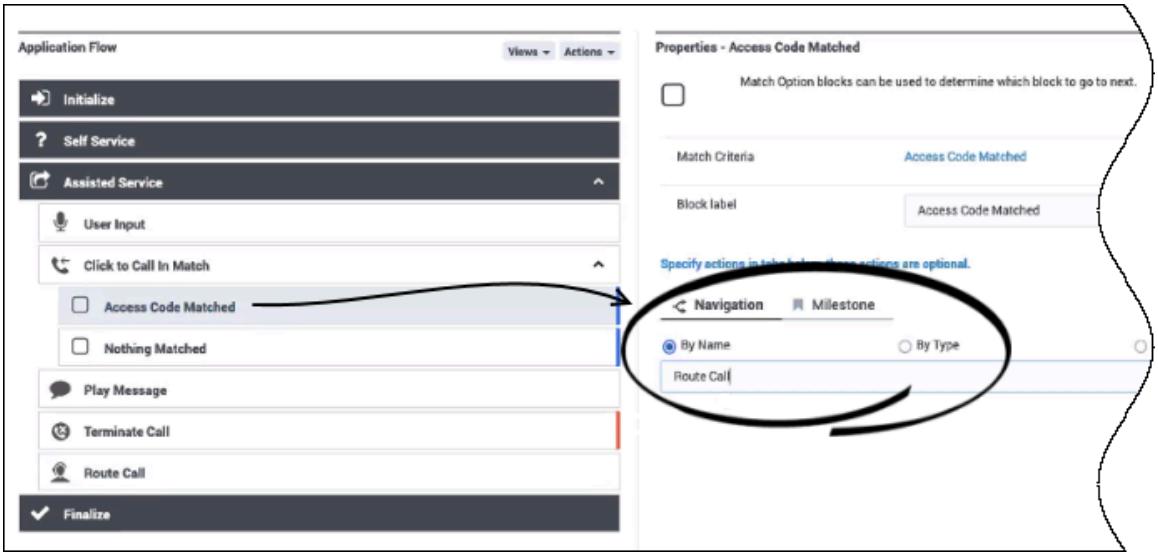
1. In Designer, create an application of type `Default`.
2. Click the **Initialize** phase and add the variables that you will use to store block outputs.
3. We will add the following blocks to the **Assisted Service** phase to handle the call flow:
   - User Input: This is required when you are using access codes with the Click-To-Call-In (Delayed) scenario. You configure this block to prompt the user to enter the access code.
   - Click-to-Call-In Match: In the Click To Call In Match block, on the **Match criteria** tab, select the option to use the access code that the consumer enters as the matching criterion.



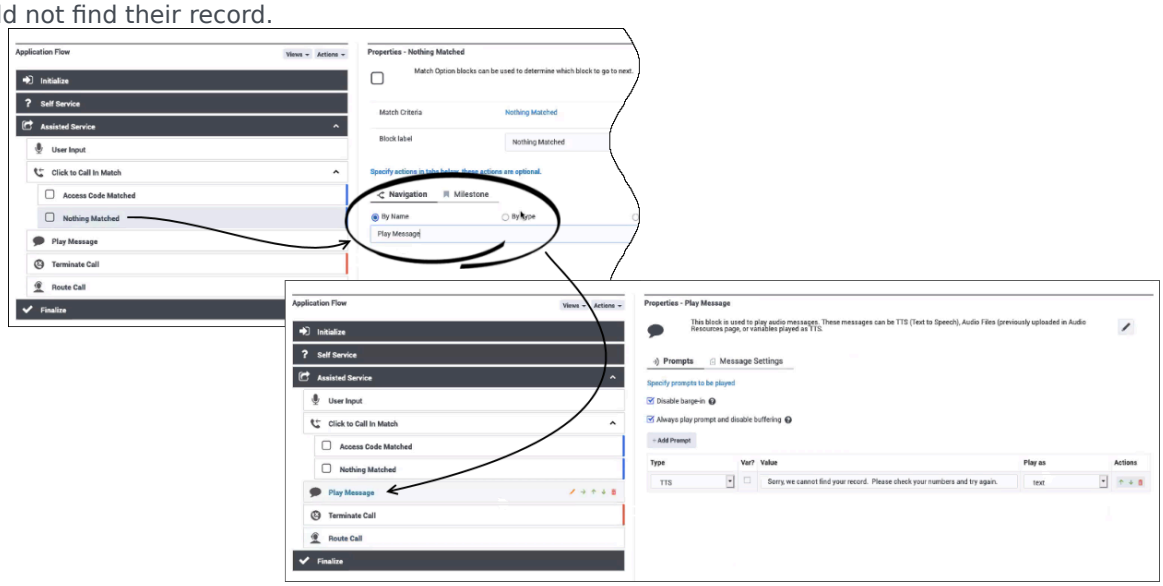
Make sure that you configure the parameters on the Routing Parameters tab. As with any application, consider the logic that needs to be configured to handle all outcomes appropriately for your contact center.



If a match is made between the caller and an existing callback record, then the call needs to move to the Route Call block.



- Play Message: When the system can find no successful match for the caller or the caller is dropped from the Match block for any reason, then the Play Message block informs the caller that the system could not find their record.



- Terminate Call: When no match can be found between a caller and existing callback records, and the message in the Play Message block has finished playing, then the call drops to the Terminate Call block and the call ends.
- Route Call: When a caller matches a callback request that exists in the system, then the system routes the call based on the information in this block. The routing priorities, virtual queue, skill expressions, and so on are included in the **Routing Parameters** tab of the Click To Call In Match block. You must assign the variables used in the **Routing Parameters** tab to the corresponding fields in the Route Call block in order for the call to retain its priority from the original callback request.

Configuring Click-To-Call-In Groups

Contents

- 1 What does a Click-To-Call-In group consist of?
- 2 A note about migration
- 3 Provisioning Click-To-Call-In Groups
  - 3.1 Additional information about the Settings
- 4 Performing a bulk import of DNIS pool numbers

-
- Administrator

As a Callback administrator, you are setting up the Click-To-Call-In groups that you require for the Click-To-Call-In scenario. We'll tell you how to add your DNIS numbers to a group and how to enable the use of an access code for consumer call-ins. If you are migrating groups from an earlier version of Callback, be sure to read the note related to this activity.

Related documentation:

-

To create or modify Click-To-Call-In groups, you must be a member of the **Callback Administrator** Role. The **Callback Developer**, **Callback Supervisor**, and **Callback Monitor** Roles have read-only access.

For information about the Callback **Call-In** tab on which you can view the Click-To-Call-In records, see Viewing Click-To-Call-In records.

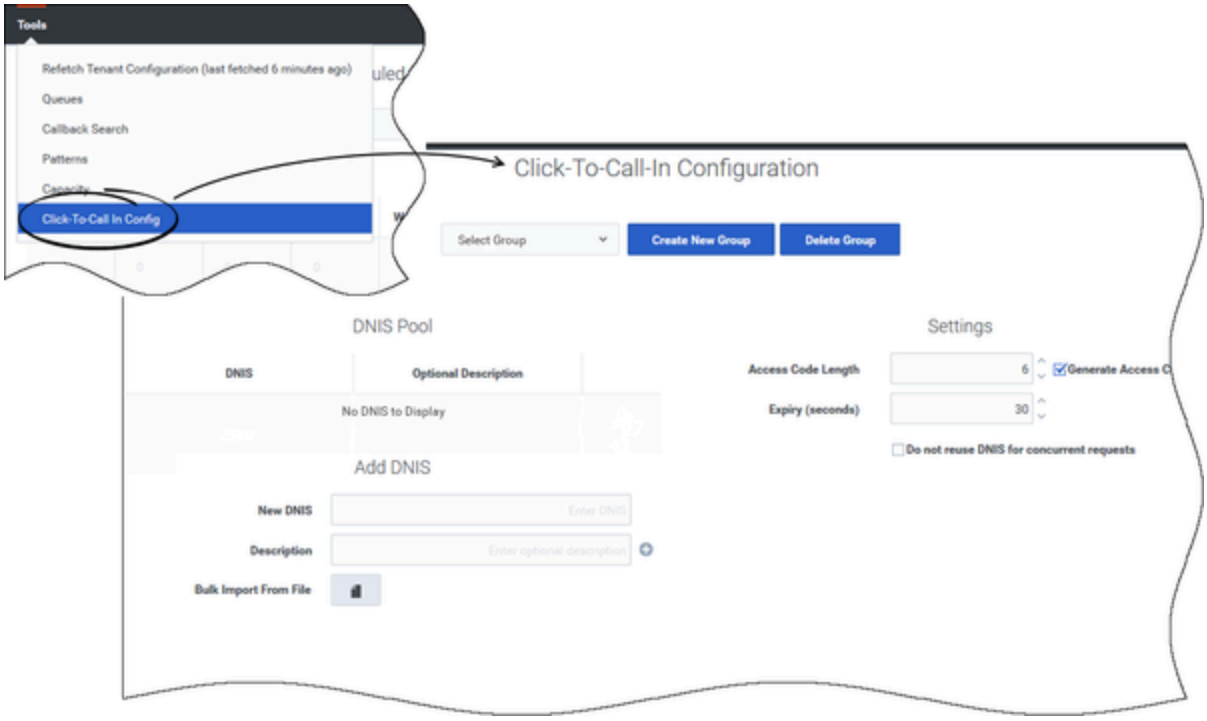
You can find information about using the Genesys Engagement Service (GES) APIs, including the Call-In API, on the Genesys Multicloud CX REST APIs and Tutorials for Callback page in this Guide. The page includes links to API documentation.

If you are using the Click-To-Call-In (Delayed) scenario, then you must do additional provisioning in Designer.

What does a Click-To-Call-In group consist of?

A Click-To-Call-In group definition consists of a Dialed Number Identification Service (DNIS) pool and access code settings, including the choice to use or not use an access code with Click-To-Call-In requests. The system uses the DNIS pool associated with a group to find the number that it will provide to the customer who is making the Click-To-Call-In request. The system also uses the group configuration to determine whether or not an access code is required when the customer dials in.

In Callback, you provision the Click-To-Call-In groups on the **Tools** > **Click-To-Call-In Config** tab.



There are two ways in which you can enter the DNIS pool of numbers for a Click-To-Call-In group:

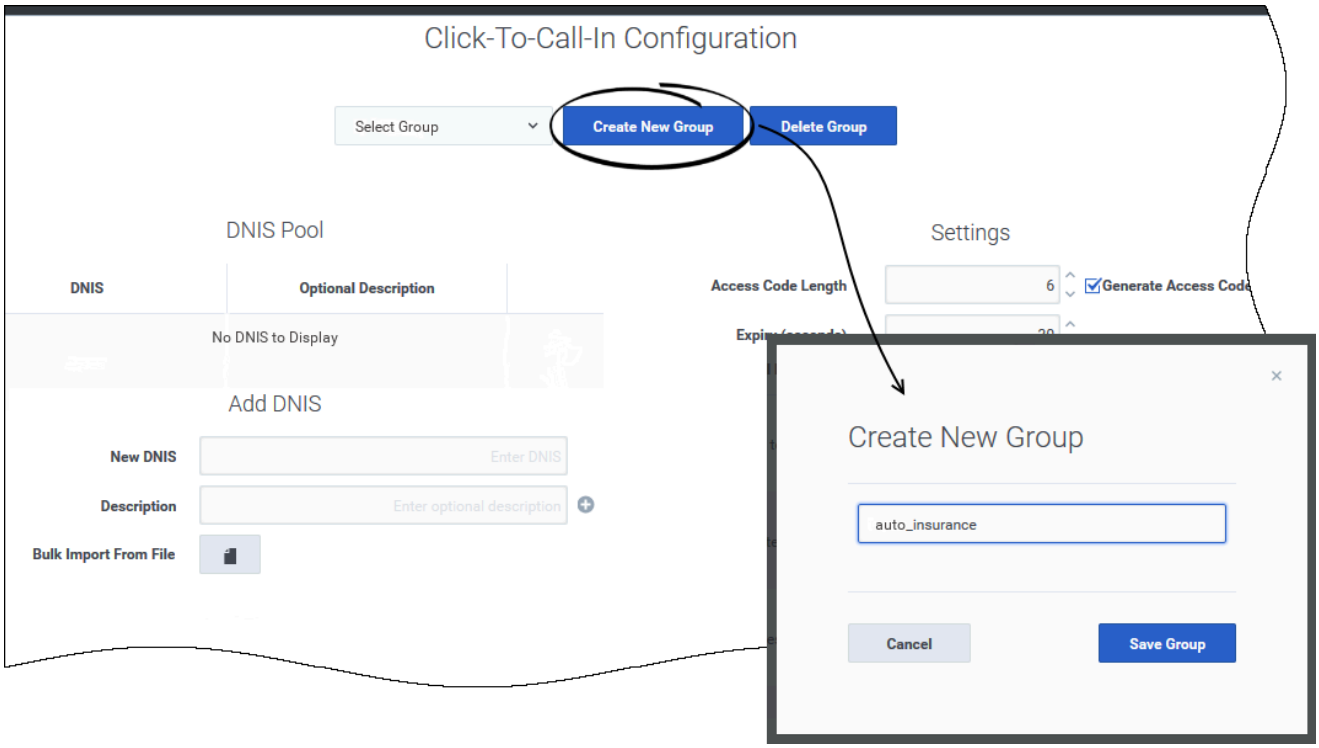
- Enter each DNIS number individually using the fields on the **Click-To-Call-In Config** page.
- Bulk import the DNIS numbers for a group.

The information on this page describes both methods. For general information, including how to enter a DNIS pool manually, see Provisioning Click-To-Call-In Groups. For information about the bulk import of a DNIS pool, see Performing a bulk import of DNIS pool numbers below.

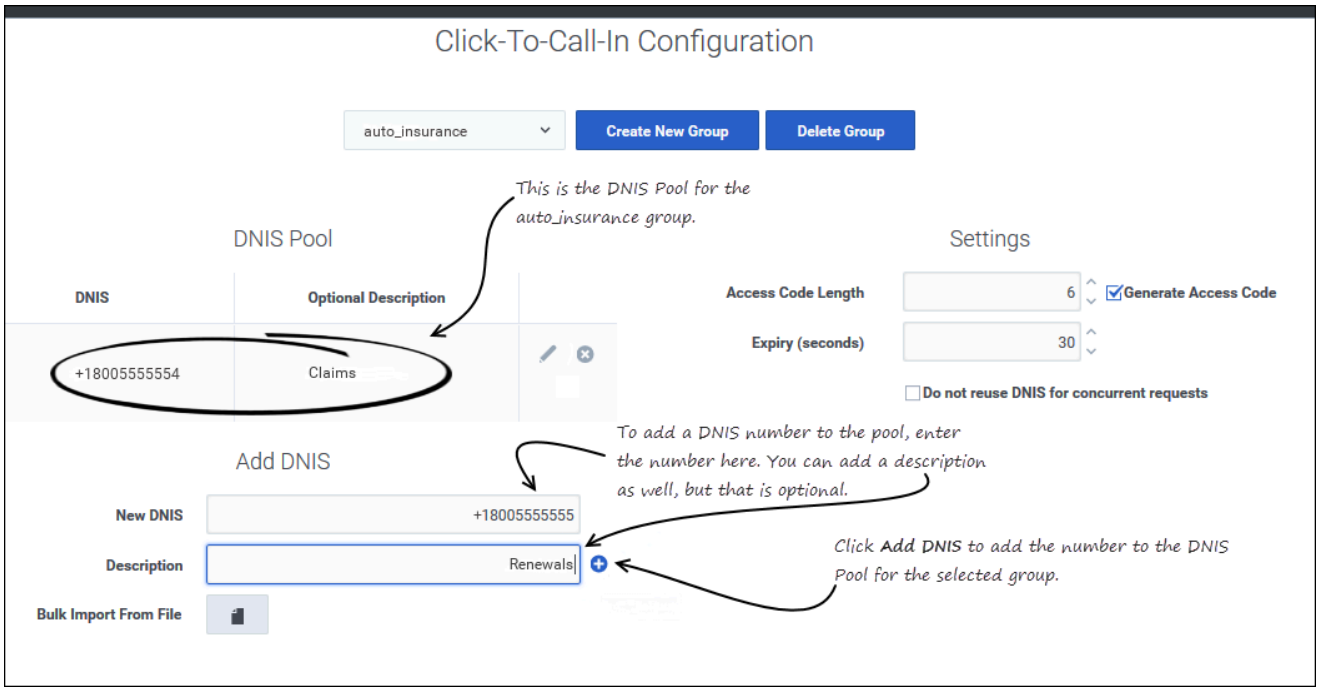
A note about migration

If you are *migrating* to the latest version of Callback, rather than starting fresh, make sure that you enter any existing Click-To-Call-In group information on the **Click-To-Call-In Config** page in the Callback UI. Previously, the Click-To-Call-In group information was configured in Platform Administration.

Provisioning Click-To-Call-In Groups



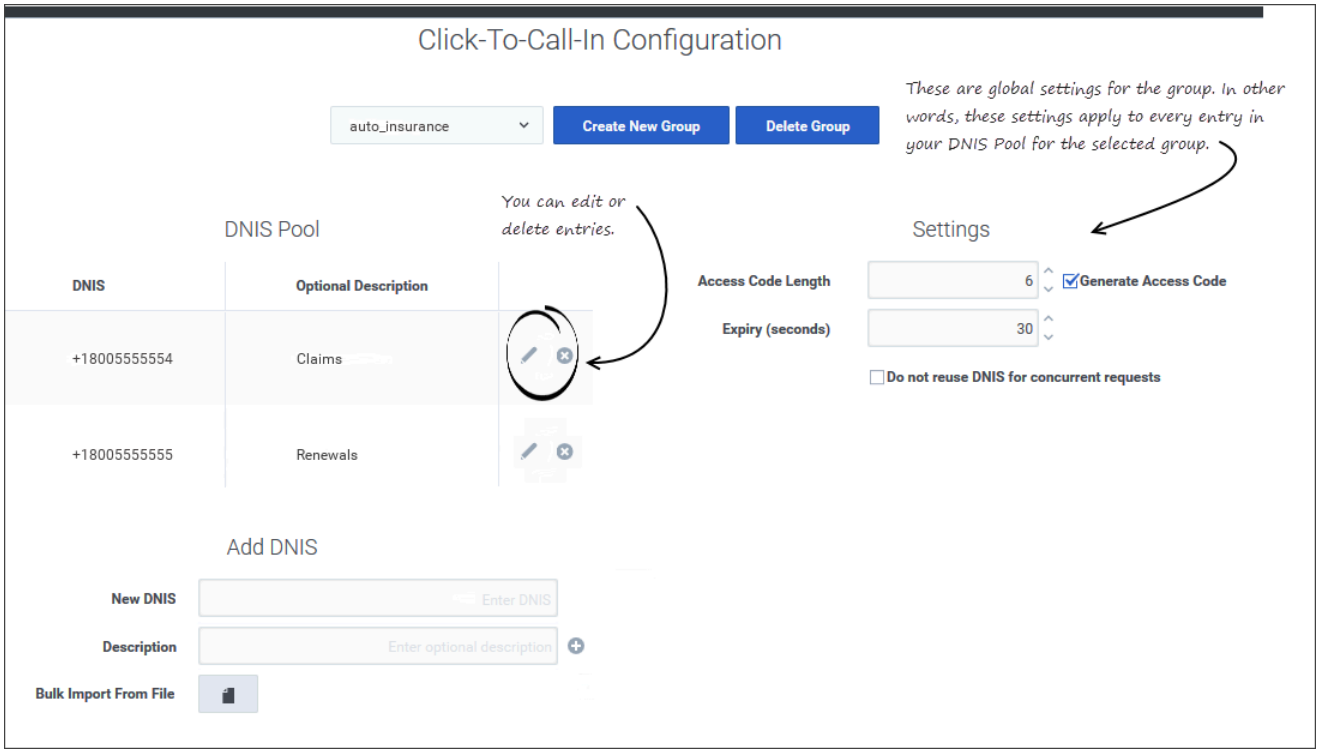
To create a new Click-To-Call-In group, navigate to the **Tools** > **Click-To-Call-In Config** page and click **Create New Group**. The UI accepts alphanumeric characters and symbols in the group name. You cannot edit the name of a group in the Callback UI after you click **Save Group**, so be sure that you have entered the name exactly as you intended before saving.



In the pane on the left side of the **Click-To-Call-In Configuration** page, you enter your DNIS pool of numbers. The group must contain at least one valid DNIS number (that is, it must conform to the E.164 format for telephone numbers). You can enter a maximum of 10,000 DNIS numbers for each group. A DNIS number cannot exceed 16 characters, and can include the "plus" sign (+) prefix.

You can include a description for a DNIS number, but that is optional. The UI accepts alphanumeric characters and symbols in the description. The description cannot exceed 32 characters.

If you have a large DNIS pool for a group, you can bulk import the numbers instead of entering each one individually. For more information about the bulk import method, see Performing a bulk import of DNIS pool numbers below.

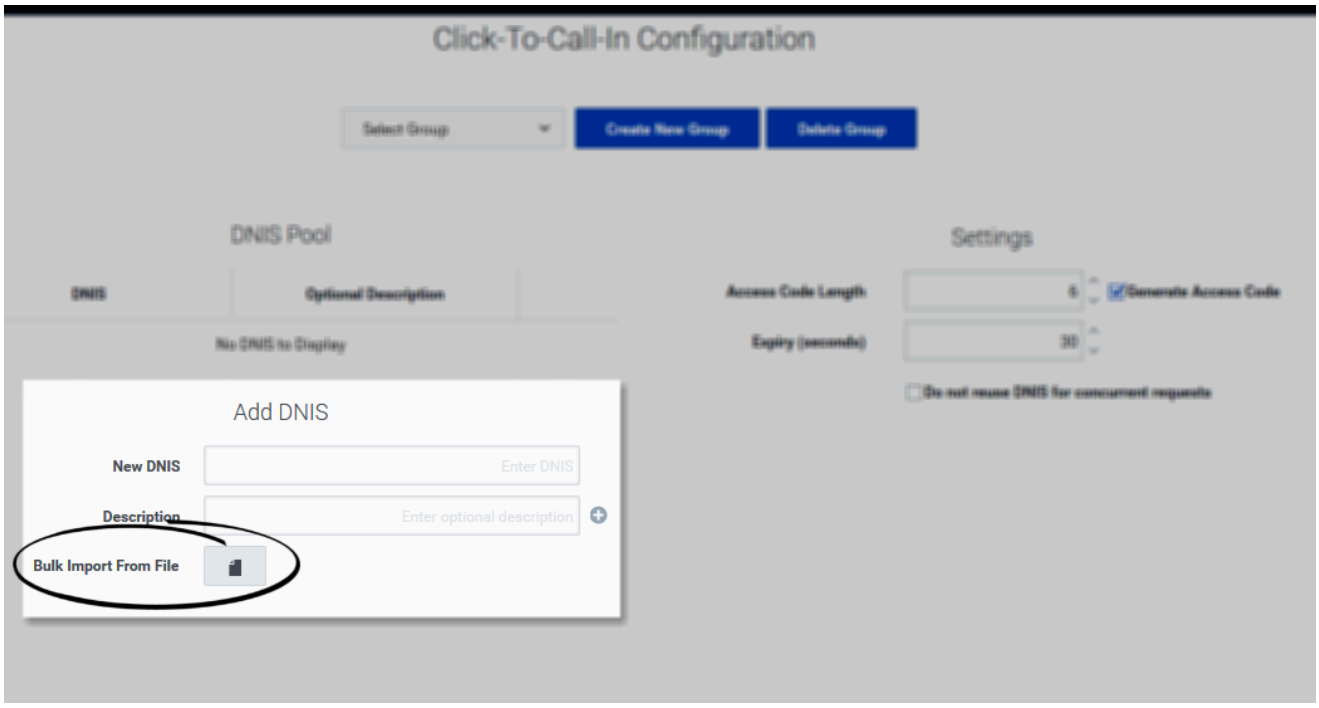


In the pane on the right side of the page, you enable or disable the use of an access code. In other words, the access code is optional. If you enable the use of the access code, then you can specify settings for the code. If enabled, the system generates an access code for each Click-To-Call-In request and provides it to the caller at the same time that it provides the phone number that the caller must use to dial in. The caller must enter the access code correctly for the call to be directed to an agent.

Additional information about the Settings

Setting	Description
Generate Access Code	Enable or disable the use of access codes with Click-To-Call-In requests.
Length	The length of an access code. Valid values are 4 to 12, inclusive. The default value is 6.
Expiry (seconds)	The lifetime, in seconds, of the Click-To-Call-In request. Valid values are 10 to 1800, inclusive. The default value is 30.
Unique DNIS	Specifies if a DNIS number is unique or not. "Unique" means that it cannot be reused while it is in use. Select the check box if the DNIS is unique. Otherwise, leave the check box clear. The default value is false, or "not unique" (the check box is not selected).

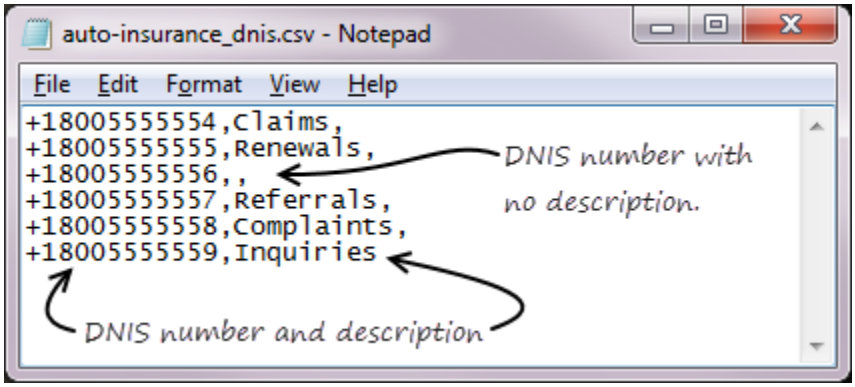
Performing a bulk import of DNIS pool numbers



If you have a large DNIS pool associated with a Click-To-Call-In group, you can import the DNIS pool as a bulk upload. After you create the group on the **Tools** > **Click-To-Call-In Config** page, then use the **Bulk Import From File** button to import the DNIS pool from a .csv file.

If you use the bulk import method, you will require a separate .csv file for each group. You can upload the DNIS pool for a single group from a file; you cannot upload the DNIS pool for many groups from one file.

The bulk import method supports only .csv files (values must be separated by a comma). You can include a description, but it is optional. In the sample file shown here, there is no description for one entry, although you still require the commas.



The DNIS numbers must be entered in the .csv file using the E.164 format for telephone numbers. The system will check the values for compliance and prompt you to fix any errors. All rules that apply when entering the DNIS numbers individually also apply when bulk uploading the pool of numbers.

Viewing Click-To-Call-In records

Contents

- 1 Using the Call-In tab
- 2 Filtering and sorting the records list

-
- Administrator

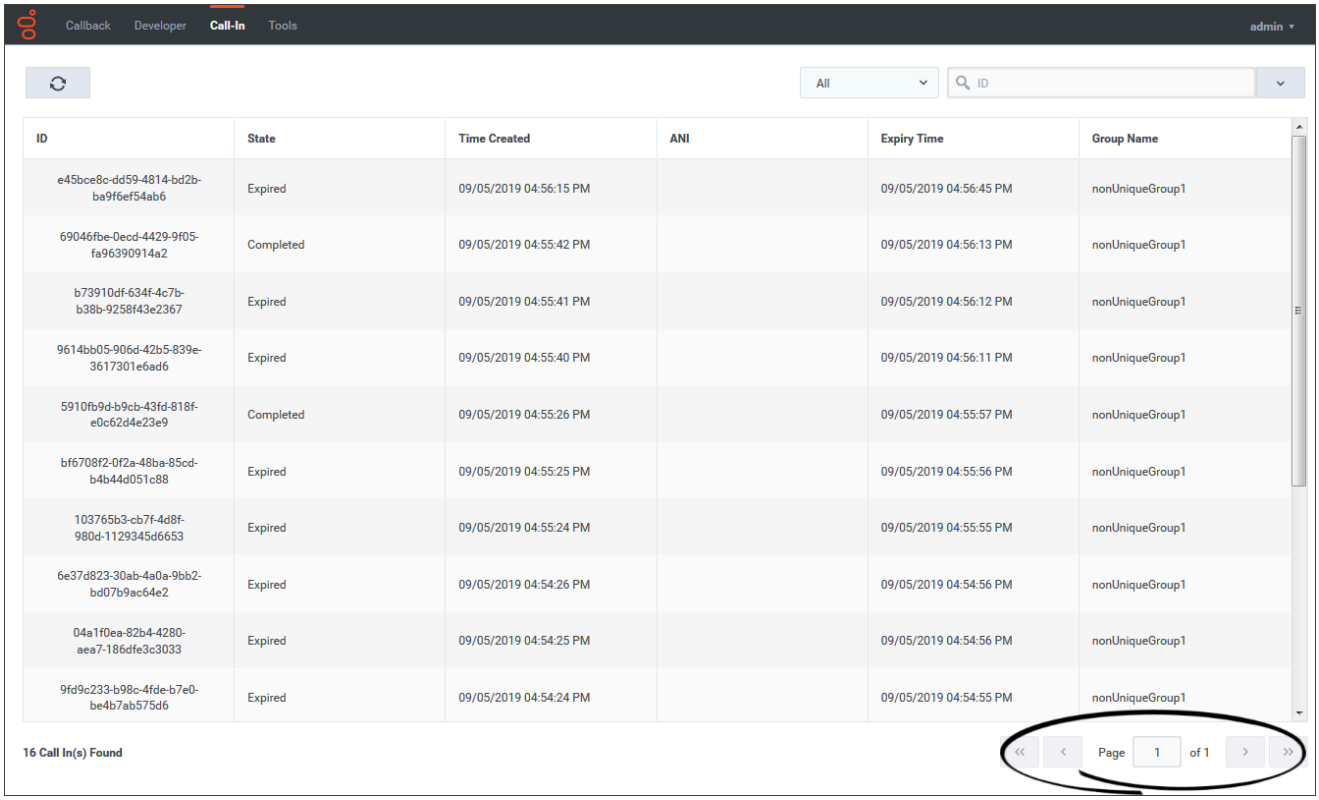
If you use the Click-To-Call-In feature with Callback, use the **Call-In** tab in the Callback UI to view the Click-To-Call-In records.

Related documentation:

-

To view the **Call-In** tab, you must be assigned the **Callback Administrator** or **Callback Developer** Role.

Using the Call-In tab



ID	State	Time Created	ANI	Expiry Time	Group Name
e45bce8c-dd59-4814-bd2b-ba9f6ef54ab6	Expired	09/05/2019 04:56:15 PM		09/05/2019 04:56:45 PM	nonUniqueGroup1
69046fbe-0ecd-4429-9f05-fa96390914a2	Completed	09/05/2019 04:55:42 PM		09/05/2019 04:56:13 PM	nonUniqueGroup1
b73910df-634f-4c7b-b38b-9258f43e2367	Expired	09/05/2019 04:55:41 PM		09/05/2019 04:56:12 PM	nonUniqueGroup1
9614bb05-906d-42b5-839e-3617301e6ad6	Expired	09/05/2019 04:55:40 PM		09/05/2019 04:56:11 PM	nonUniqueGroup1
5910fb9d-b9cb-43fd-818f-e0c62d4e23e9	Completed	09/05/2019 04:55:26 PM		09/05/2019 04:55:57 PM	nonUniqueGroup1
bf6708f2-0f2a-48ba-85cd-b4b44d051c88	Expired	09/05/2019 04:55:25 PM		09/05/2019 04:55:56 PM	nonUniqueGroup1
103765b3-cb7f-4d8f-980d-1129345d6653	Expired	09/05/2019 04:55:24 PM		09/05/2019 04:55:55 PM	nonUniqueGroup1
6e37d823-30ab-4a0a-9bb2-bd07b9ac64e2	Expired	09/05/2019 04:54:26 PM		09/05/2019 04:54:56 PM	nonUniqueGroup1
04a1f0ea-82b4-4280-aea7-186dfe3c3033	Expired	09/05/2019 04:54:25 PM		09/05/2019 04:54:56 PM	nonUniqueGroup1
9fd9c233-b98c-4fde-b7e0-be4b7ab575d6	Expired	09/05/2019 04:54:24 PM		09/05/2019 04:54:55 PM	nonUniqueGroup1

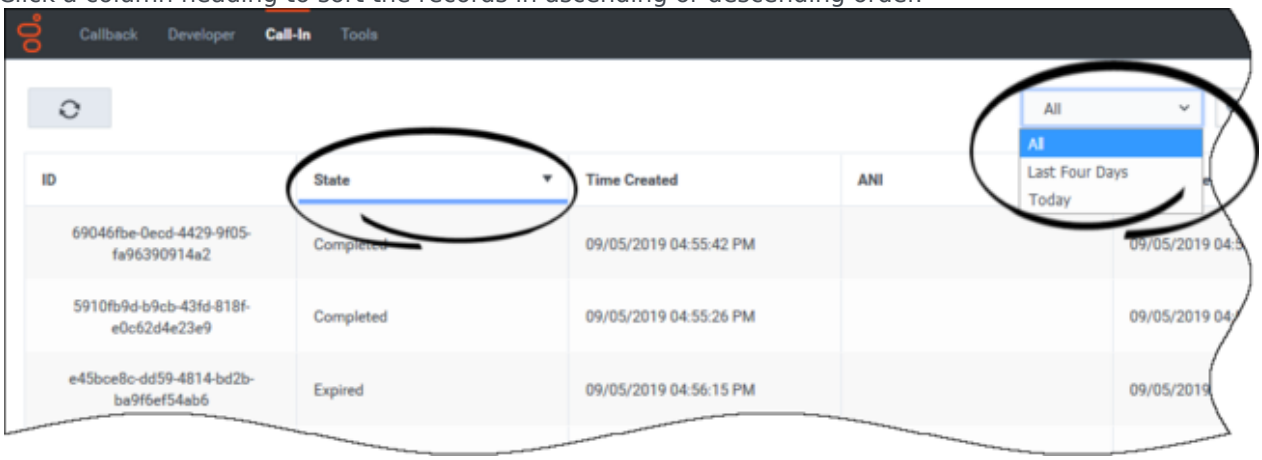
The **Call-In** tab lists up to 5000 Click-To-Call-In records. Once you have more than 5000 Click-To-Call-In records, the oldest records are dropped from the list in the UI.

One page on the **Call-In** tab displays up to 500 records. Once you have more than 500 records, you can use the pagination tools at the bottom of the page to navigate through the records.

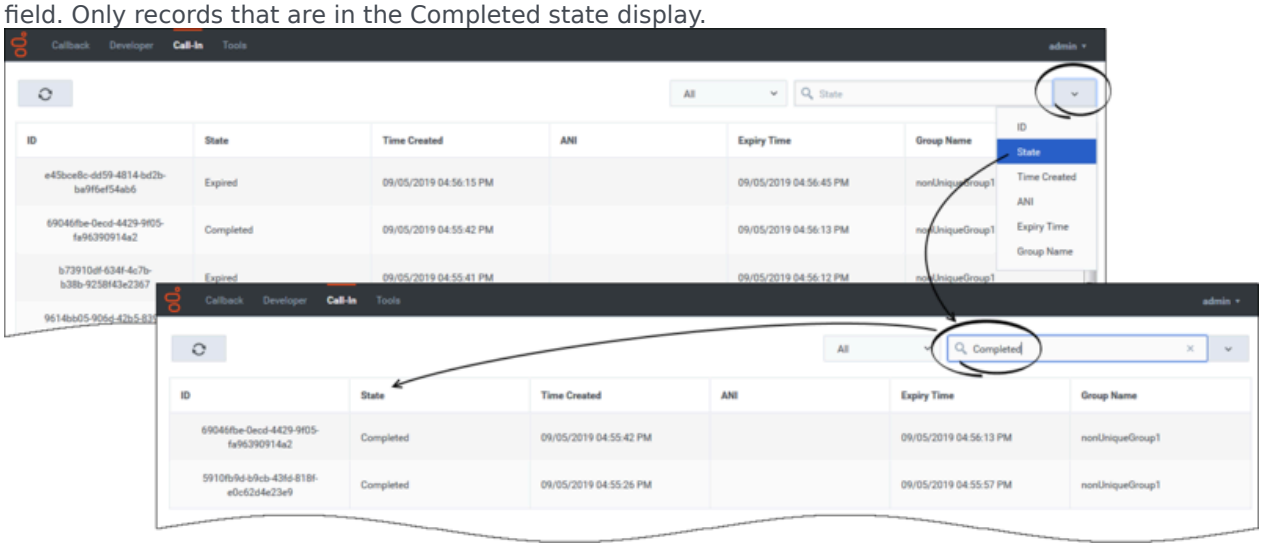
Filtering and sorting the records list

To help you find specific records more easily, the **Call-In** tab offers the following filtering tools:

- Filter the list using the predefined time-range filters available at the top of the page.
- Click a column heading to sort the records in ascending or descending order.



- Search based on a specific record attribute. In this example, we want to find all of the records that are in the Completed state. We select `State` from the drop-down menu and enter `Completed` in the **Search** field. Only records that are in the Completed state display.



Troubleshooting and validating functionality

Contents

- 1 Displaying errors
- 2 Video: Purging callbacks
- 3 Viewing CORS domains
- 4 Validating your API key
- 5 Testing credentials and feature functionality
- 6 Booking a callback using the callbacks create API
- 7 Refreshing your Tenant configuration data

-
- Administrator

The Callback UI includes a number of troubleshooting and validation tools, available as menu options under the **Developer** and **Tools** tabs. This page describes those tools and also describes some features for which you might need to investigate the existing configuration as part of your troubleshooting effort.

Related documentation:

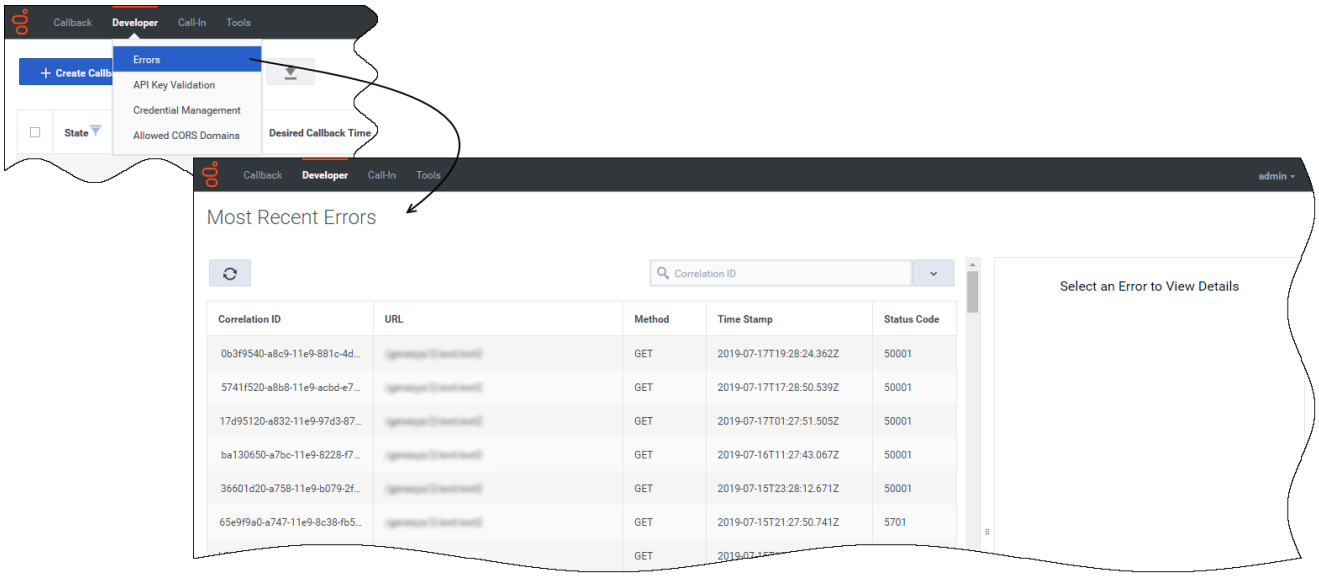
-

To view the **Developer** tab in the Callback UI, you must be assigned to the **Callback Administrator** or **Callback Developer** role.

Callback Administrators, Supervisors, or Developers have access to the **Tools** tab.

> **Important**
> Designer supports Callback for voice calls only; digital interactions are not supported.

Displaying errors



The **Developer** > **Errors** page displays the list of errors related to your Callback API queries. It is designed to help you solve API issues when developing an application using the REST APIs. When an API call returns an error, the response includes a correlation ID. Use this correlation ID as input for the **Search** field to find more information about a specific API call.

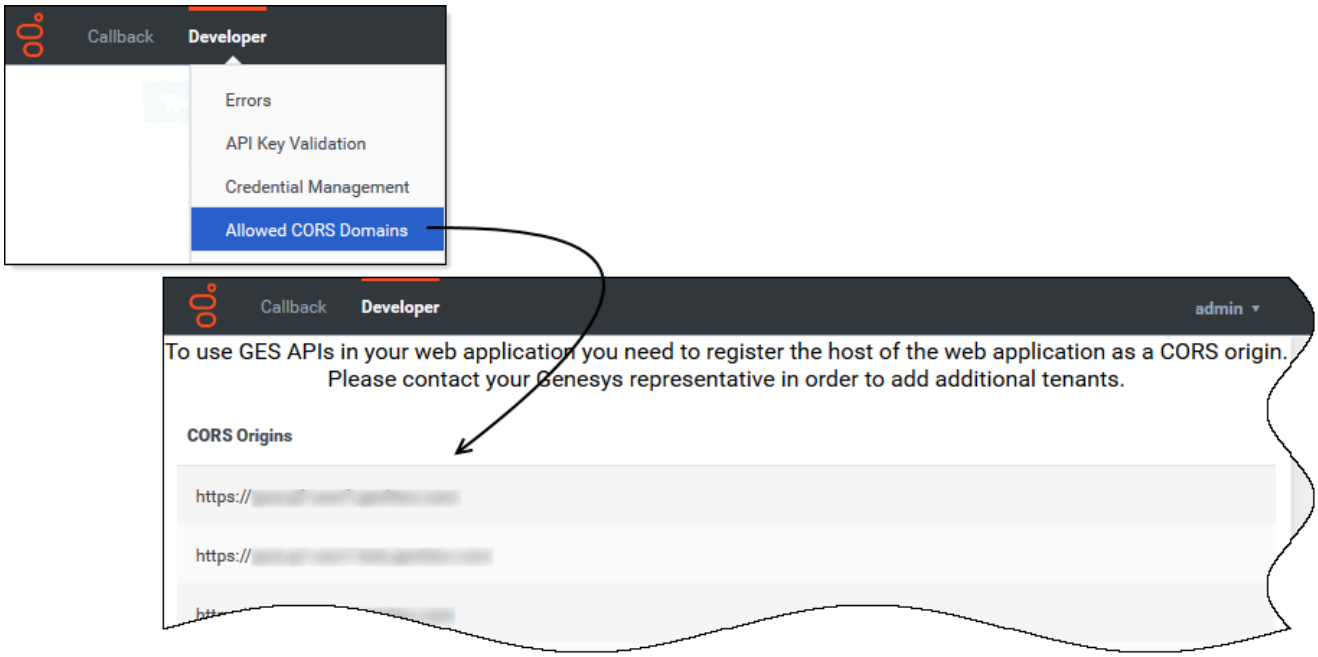
Video: Purging callbacks

Link to video

This video describes the built-in callback purging feature. If you have regular business hours configured in Designer, then Callback automatically purges callbacks at the end of your business day. For global businesses, this helps to prevent such things as overnight callbacks to customers. You also have the option to purge callbacks after a specific duration of time has elapsed. This is helpful if your contact centers are open 24/7.

The video explains these options, provides some tips about how to use the purging feature, and describes how the purging feature works in some common scenarios.

Viewing CORS domains

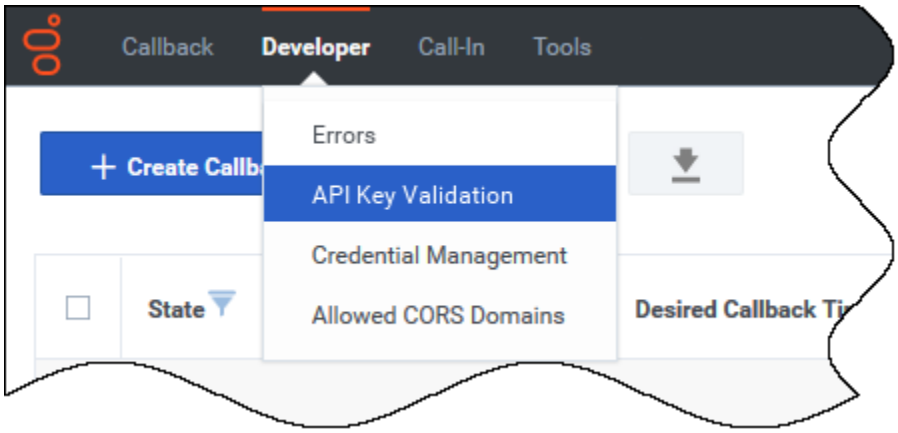


To access Genesys Engagement Services (GES) APIs in a Web application, your

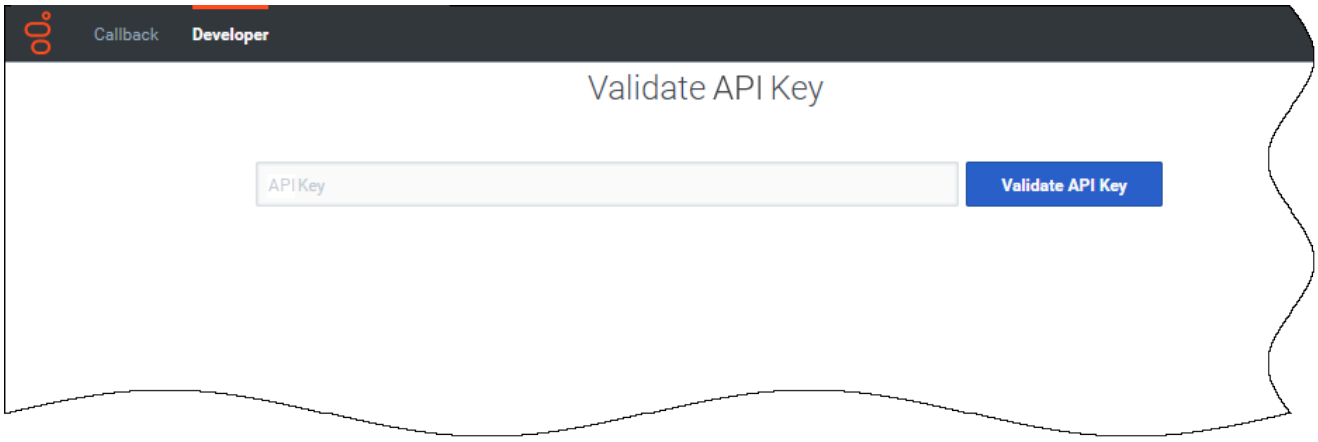
Web application's host must be registered as a "safe" domain or origin. The **Developer** tab includes an **Allowed CORS Domains** page, which you can access from the **Developer** tab menu. On the **Allowed CORS Domains** page, you can view the list of domains or origins that have been provisioned for your use. Cross-origin resource sharing (CORS) requests to the origins in the list are allowed. The list of origins on the **Allowed CORS Domains** page is read-only.

Updating the list of CORS domains	If you use Callback from the Genesys Multicloud CX hosted service, contact your Genesys representative to add cross-region resources to the list. If you don't have the **Allowed CORS Domains** page in your Callback UI, contact Genesys Customer Care to request it. If you are a Genesys Multicloud CX private edition customer, see Update CORS settings.

Validating your API key



If your permissions allow it, the **Developer** tab menu includes **API Key Validation**.

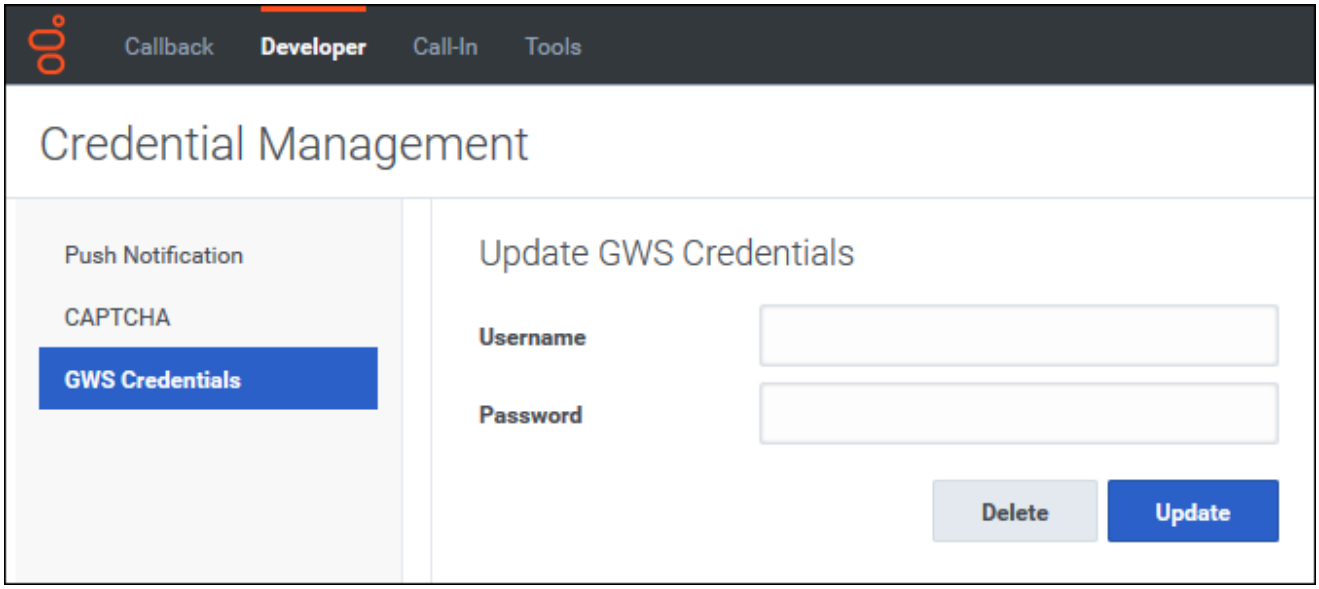


Copy/paste your API Key into the text box and click **Validate API Key**.

- If your API Key is incorrect, the UI displays **Invalid API Key**.
- If your API Key is valid, the UI displays **Valid API Key**.

Testing credentials and feature functionality

The **Developer** tab menu includes **Credential Management,** if you have sufficient permissions to view this part of the interface. Use **Credential Management** to register, manage, and test credentials for features within your Callback environment and features that work within other supported delivery networks.



To view GWS statistics information through Callback's Statistics API, you must register your credentials on the **Credential Management** > **GWS Credentials** tab before you can retrieve the statistics.

For additional information about the Callback APIs, see Genesys Multicloud CX REST APIs and tutorials for Callback.

If you use the following features with callbacks, then register, manage, and test the credentials associated with these features on the **Credential Management** page:

- Push Notification
- CAPTCHA

Click the feature name to find information about how to provision these features for use with Callback.

Booking a callback using the callbacks create API

When you book a callback using the Callback UI or the /callbacks/create API, the callback is booked regardless of the specified office hours or whether the virtual queue is enabled for immediate or scheduled callbacks.

If you are booking callbacks directly with the API, not all settings in the CALLBACK_SETTINGS data table are taken into account. The parameters that are included with API callbacks are marked with a checkmark (✓) in the **Web Callback** column of the CALLBACK_SETTINGS parameters table. For some settings, the API reads them, but only when an associated parameter is set to `true`. For information about the settings and relevant parameters, see Web Callbacks (API) in the *Designer User's Guide*.

When you use the Callback UI to book a Scheduled callback, only available timeslots are returned for the hours configured. However, when you use the callbacks/create API to book a Scheduled callback, the slot availability is *not* checked. Genesys strongly recommends that you use the other Genesys Engagement Services (GES) APIs to check parameters before booking the callback. The following APIs are relevant in this situation:

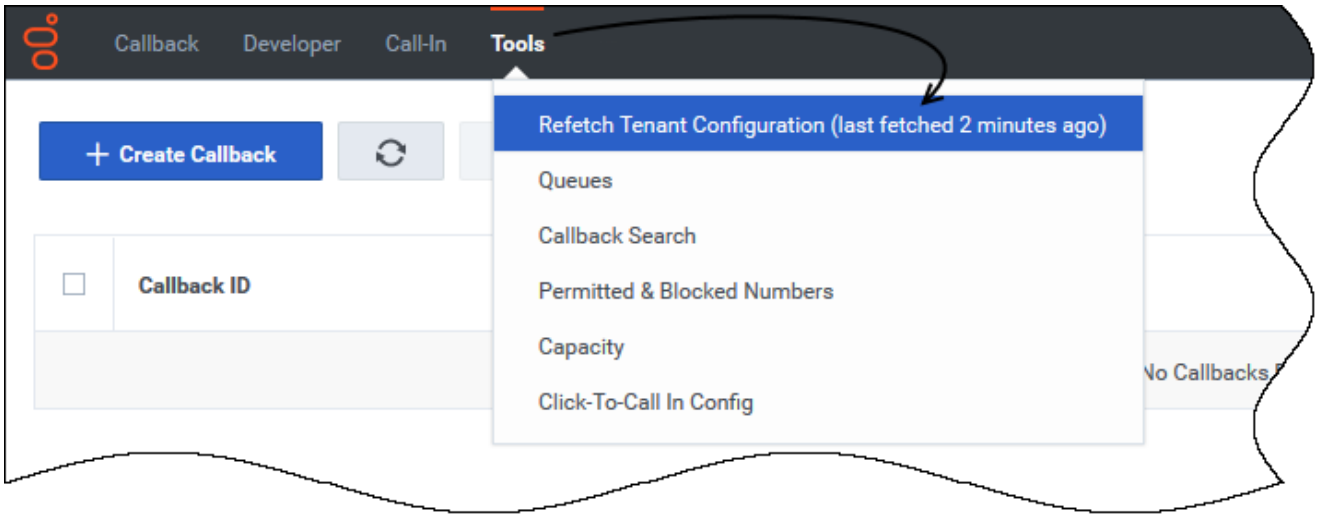
- To check if the office is open (configured in the **Business Hours** column in the CALLBACK_SETTINGS data table), use the /callbacks/open-for API.
- To check which Scheduled callback timeslots are available (configured in the **Business Hours**, **Slot Capacity**, and **Slot Duration** columns in the CALLBACK_SETTINGS data table), use the /callbacks/availability API.
- To check the values of **Immediate Enabled** and **Scheduled Enabled** for a callback virtual queue, use the /callbacks/queue-status API.

Starting with GES version 100.0.003.0156, you can use the open-for API to find the following additional information, but only when certain input parameters are set to `true`. The following table provides additional information.

open-for API input parameter = `true`	Associated output parameter and values
queryImmediateOfferHours	**immediateOfferHoursOpenFor** The API returns the following values: • If **Immediate Offer Hours** is not configured for

open-for API input parameter = `true`	Associated output parameter and values
	the callback virtual queue, the API returns `-1`. • If **Immediate Offer Hours** is configured, but the current time is outside of the configured offer hours, the API returns `0`. • If **Immediate Offer Hours** is configured and the current time is within the offer hours, the API returns the number of seconds until the end of the business day associated with **Immediate Offer Hours**. **NOTE:** If you specify a value for the **start** parameter, then that value is used as the current time for any open-for API calculation. However, because some of the open-for API output parameters depend on estimated wait time, and estimated wait time is a real-time calculation, Genesys recommends that you avoid specifying a **start** time when using **queryImmediateOfferHours** or **returnWithinImmediateOfferTime**.
returnWithinImmediateOffertime	**isWithinImmediateOfferTime** Indicates whether Immediate callbacks can be offered based on: • The current time. • The current estimated wait time. • **Immediate Offer Hours**, if configured. • **Business Hours** and **Immediate Blackout (minutes)** when **Immediate Offer Hours** is not configured.
checkImmediateEnabled	**isImmediateEnabled** The API returns the **Immediate Enabled** value that's set for the callback virtual queue.

Refreshing your Tenant configuration data



During the process of migrating from an older version of Callback, you can use the Tools > Refetch Tenant Configuration menu option to immediately reflect recent changes to Tenant configuration in your Callback UI or to find out when the Tenant configuration data was last refreshed. The actual Tenant configuration is performed in Platform Administration or Agent Setup.

By default, Callback fetches the Tenant configuration every 30 minutes. When you click **Refetch Tenant Configuration**, the system refreshes Tenant configuration data wherever it exists throughout the Callback UI.

Once you are fully migrated to the latest version of Callback, there is no data in the UI that will be impacted by selecting this menu option.