



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.

Genesys Callback Private Edition Guide

Provision Genesys Engagement Service

7/27/2024

Contents

- 1 Create routes for GES to use
 - 1.1 Create an external route
 - 1.2 Create an internal route
- 2 Validate DesignerEnv in Agent Setup
- 3 Provision a User for GES
- 4 Provision GES in Designer
 - 4.1 Provision the callback services (virtual queues)
 - 4.2 Publishing data tables
- 5 Configuring callback features
- 6 Additional Roles and Access Groups

-
- Administrator

Learn how to provision Genesys Engagement Service.

Related documentation:

-
-
-
-

RSS:

- [For private edition](#)

Create routes for GES to use

Create routes for Genesys Engagement Service (GES) only if routes were not created using the Helm chart while deploying the service.

Create an external route

Create an external route for GES. If you use the following sample as a guide, remember to replace with the value for your environment.

external-service.yaml:

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: ges-ext-ingress
  namespace: ges
  annotations:
    cert-manager.io/cluster-issuer:
    kubernetes.io/ingress.class: nginx
    nginx.ingress.kubernetes.io/ssl-redirect: 'false'
    nginx.ingress.kubernetes.io/use-regexp: 'true'
spec:
  tls:
    - hosts:
      - ges.
      secretName: ges-ingress-ext
  rules:
    - host: ges.
      http:
        paths:
          - path: /*
            pathType: ImplementationSpecific
            backend:
              service:
```

```
name: ges
port:
  number: 8080
```

Create the External Service spec for the external route:

```
kubectl create -f external-ingress.yaml
```

Create an internal route

Create an internal route for GES. If you use the following sample as a guide, remember to replace with the value for your environment.

internal-route.yaml:

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: ges-int-ingress
  namespace: ges
  annotations:
    cert-manager.io/cluster-issuer:
    kubernetes.io/ingress.class: nginx
    nginx.ingress.kubernetes.io/ssl-redirect: 'false'
    nginx.ingress.kubernetes.io/use-regexp: 'true'
spec:
  tls:
    - hosts:
        - ges-int.
      secretName: ges-secret-int
  rules:
    - host: ges-int.
      http:
        paths:
          - path: /*
            pathType: ImplementationSpecific
            backend:
              service:
                name: ges
                port:
                  number: 80
```

Create the External Service spec for the internal route:

```
kubectl create -f internal-ingress.yaml
```

Validate DesignerEnv in Agent Setup

Validate that the DesignerEnv transaction list is specified and correct for GES functionality. In Agent Setup, navigate to the DesignerEnv transaction on the **Transactions** page and review the **Annex** settings. The following settings must be specified (you might need to specify the flowsettings and gms sections and options):

- flowsettings > useSharedGms=true
- gms > _url=http://ges.ges.svc.cluster.local/genesys/

-
- reporting > ReportingURL=true

Currently, the ges section with key gesurl is not a required setting; it is not used in the system.

Annex tab options for DesignerEnv:

```
flowsettings
  customerfolder : Environment
  enablePTE : False
  httpProxy:"
  outboundTimeout : 20
  primarySwitch : SIP_Switch
  recordingType : FCR
  redundantHttpProxy:"
  useSharedGms : true

gms
  _url: http://ges.ges.svc.cluster.local/genesy/

reporting
  ReportingURL : true
```

Provision a User for GES

Use Agent Setup to provision an administrative User for GES. Be sure to configure appropriate Role-Based Access Control (RBAC) for the User. In other words, be sure to add the Admin User to the correct Access Groups for administrative-level access to the system or to GES:

- If you are configuring a general Admin User, use Access Group “Administrators”. In our example, it is “t100 Administrators”.
- If you are configuring an Admin User specifically for GES/callbacks, use Access Group “Callback Developers”.

Edit User

Ges Admin

User ^

- General Info
- Caller IDs
- Access Group
- Annex
- Favorites Pool
- User Favorites
- Switches
- External URLs

Desktop Options v

Digital Management v

Gplus Salesforce v

Recording v

General Info

Folder

First Name *

Last Name *

Username *

Email

Password * Reset Password

Password confirm *

Employee ID

External ID

State Enabled

Multimedia Agent

Call Recording

Agent Desktop Agent

Agent Desktop Supervisor

Agent Setup Admin

Contact Center Supervisor/Administrator

Console developer

Console support

Phone Number(s)

Number	Description	Default	Place Name	Use
<input type="text" value="Phone Number"/>	<input type="text" value="Description"/>	<input type="checkbox"/>	<input type="text"/>	<input type="checkbox"/>
<input type="text" value="Phone Number"/>	<input type="text" value="Description"/>	<input type="checkbox"/>	<input type="text"/>	<input type="checkbox"/>
<input type="text" value="Phone Number"/>	<input type="text" value="Description"/>	<input type="checkbox"/>	<input type="text"/>	<input type="checkbox"/>

Cancel
Delete User
Save
Save and close

Access Group

Add
Remove

<input type="checkbox"/>	Access Group	⬇
<input type="checkbox"/>	t100 Administrators	
<input type="checkbox"/>	Callback Developers	

Provision GES in Designer

You provision GES/Callback in Designer. Log into Designer as a User who is a member of the **Designer Developers Group**. For a description of basic GES provisioning in Designer to support

callbacks, see Provisioning Callback in Designer.

Tip

When working with Voice Microservices, Genesys recommends that you create builds for your Designer applications.

For your Default-type application, in the main Designer Application list view:

- Assign the build that you created to the **Development** stream.
- Associate a working DNIS to the **Development** stream.

For your Callback-type application, in the main Designer Application list view:

- Assign the build that you created to the **Development** stream so it can be used in the CALLBACK_SETTINGS data table.

For more information about builds, see Manage Builds and Application workflow.

Provision the callback services (virtual queues)

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.

Configuration notes

When configuring settings in the CALLBACK_SETTINGS data table, use the following guidelines:

- **Call Display Number** and **Callback Display Name** are optional settings.
- In the row in which you are configure the inbound virtual queue, select your Designer Callback-type application in the **Callback Application** column of the table.
- For the **Application Stream**, use the **Development** stream, as specified in the preceding Tip.
- The **Routing Point** setting is a free-form text box, but you must enter a valid routing point that is configured on the SIP switch.
- Whether or not you enter a **Dial Prefix** is dependent on the phone number format, dial plan set up, and so on. The actual outbound number used by Callback - before any transformations in the dial plan - is

in the format “dial prefix>callback phone number>”.

- If you deploy in AWS or Azure, use the “+” dial prefix, unless there is a very good reason not to. When used, the callback phone number> must be in E.164 format without the leading “+” (for example, 1xxxnnnnnnn format for North American numbers).
- For IVR-based callbacks, assuming the default Callback templates are used and number validation configurations are not used (the NUMBER_VALIDATION_CONFIGURATIONS data table):
 - If the ANI is properly detected (at the SIP level) and provided to the Designer strategy, and if it is in full E.164 format (for example, +1xxxnnnnnnn for North American numbers), the system states the ANI to the caller without the leading “+”; for example, “1xxxnnnnnnn”.
 - If the ANI is available and the caller accepts the spoken phone number, then that is the phone number that the system uses to book the callback. For example, if the ANI is +12223334444 and the caller accepts that detected number, then the callback is booked with phone number 12223334444.
 - If the ANI is not detected, or if the caller chooses to enter a different phone number, the phone number that the caller enters is used to book the callback. For example, if the caller entered 2223334444 as the phone number, the callback is booked with the phone number 2223334444. In this example, note the difference between the detected ANI (12223334444) and the entered phone number (2223334444).
- Continuing with IVR callbacks, if you use the default Callback templates with the virtual queue that is configured in the NUMBER_VALIDATION_CONFIGURATIONS data table, then phone numbers are normalized into the E.164 format without the leading “+” sign, regardless if it is a detected ANI or one that is manually entered.
- For web callbacks, number validation configurations (in the NUMBER_VALIDATION_CONFIGURATIONS table) do not apply. Instead, the phone number that the caller provides when booking the callback is used as is.

Publishing data tables

After you complete configuration in the relevant data tables in Designer – for example, the CALLBACK_SETTINGS and NUMBER_VALIDATION_CONFIGURATIONS data tables – make sure you save and publish the tables. If you see tables in the list view that do not have a **Last Published** date and time, then publish them now.

Name ↕	Tags	Last Modified ↕	Last Published ↕
<input type="checkbox"/> CALLBACK_SETTINGS		Today at 2:05 PM	Today at 2:05 PM
<input type="checkbox"/> NUMBER_VALIDATION_CONFIGURATIONS		Last Monday at 4:01 PM	Last Monday at 4:01 PM
<input type="checkbox"/> CALLBACK_ROUTING_ALLOCATIONS		Last Monday at 4:01 PM	Last Monday at 4:01 PM

Configuring callback features

For information about how to configure Click-to-Call-In, CAPTCHA, Push Notifications, and other advanced use case scenarios, see the *Callback Administrator's Guide*.

Additional Roles and Access Groups

In addition to administrator-type users, you must assign users to specific Roles and Access Groups for access to the Callback UI. Genesys Callback uses Role-based-access-control (RBAC) to allow and restrict users' activities within the Callback UI.

For more information about Callback-related Roles and Access Groups, see [Controlling user access](#).