

GENESYS[®]

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 Mobile Services Deployment Guide

Basic Configuration

5/4/2025

Contents

- 1 Basic Configuration
 - 1.1 Basic Configuration Overview
 - 1.2 Creating and Configuring a Resource List
 - 1.3 Configuring Services for Genesys Mobile Services
 - 1.4 Deploy DFM Files for Orchestration Server-based Services
 - 1.5 Next Steps

Basic Configuration

This page details the basic configuration steps required before your can use your Genesys Mobile Services installation. For a more general look at the configuration options available, refer to Configuration Options. Additional documentation is also available that describes considerations and configuration for:

- security and access control for your Genesys Mobile Services solution.
- load balancing suggestions for your Apache server.

Basic Configuration Overview

Genesys Mobile Services provides a set of APIs or services that require configuration before the product can be used. Depending on the volatility of their configuration, these services can either be configured in Configuration Server (when the volatility is high) or have their configuration embedded directly into a .WAR file.

This page is limited to managing configuration options using Configuration Server.

Working in Configuration Manager

Genesys Mobile Services is represented by an Application object in the Configuration Server database. This Application object is based on the "Genesys Generic Server" template and contains typical settings for a Genesys application including Server Info, Start Info, and Connections to other servers. It also includes Options that correspond to configuration details for sub-services of Genesys Mobile Services.

• Note: By design, settings in Configuration Manager for any configured service override the matching request parameters. This means that if *provide_code* is set to *true* then this service will always respond with an access code - even if the *_provide_code* parameter received with the request is set to *false*.

Configuration settings are grouped for different service types, and stored in Option sections described below:

- push Section—Notification service parameters. Not monitored at run-time, so the Genesys Mobile Server instance must be restarted for changes to take effect
- resource Section—Details about how resource groups are handled. Run-time configuration changes are supported, so changes take effect immediately.
- server Section—Cluster sub-service configuration details. Includes URL representation of this node of the cluster, consisting host, port and application name formatted in the following way: http://web_host:web_port/app_name. (Example: http://yourHostName:8080/gms). Run-time configuration changes are supported, but due to tight logical connection to the web-container configuration, a restart is needed in most cases.
- service.servicename Section—Additional configuration options for customized services.

Some services also rely on configuration details from a Transaction object in Configuration Manager that must be created and configured in your Genesys environment. **Note:** If setting up multiple Genesys Mobile Services nodes, the configuration options specified in the Application object must be the same for each instance. If you are familiar with working in a Genesys environment, this type of configuration should be second nature. If you require additional information about how to work with Configuration Manager to edit these configuration options, refer to the Help file included with that product.

Additional Considerations

Some parts of Genesys Mobile Services are configured outside of Genesys Configuration Server. One example is the *cmd.properties* file inside of the web application archive, which contains some global configuration objects such as connection parameters for accessing Configuration Server. Another would be load balancing, which relies on web server configuration.

Creating and Configuring a Resource List

Some services included with Genesys Mobile Services require a list of resources, such as a list of access numbers that can be managed. Such lists are held in a Transactions object, which is then referenced by Options set in the Genesys Mobile Services Application object. Steps required to create and populate this resource list are provided below.

Note: Be sure that you have the *Show Annex tab in object properties* option selected from the *View* > *Options* menu before starting this procedure.

Start of Procedure

- 1. Start Configuration Manager.
- 2. Under the tenant you are working with, open the Transactions folder.
- 3. Right-click and select **New > Transaction**.
- 4. On the General tab, configure the following fields:
 - Name—This name must match the *resources > resource_list_name* option value from your Genesys Mobile Services Application object. The default value is *GMS_Resources*.
 - Type—Select List from the drop down box.
 - Alias—Enter an alias of your choice.
- 5. On the *Annex* tab, create a new section. The section name used here must match the value of the *resource_group* option, located in the *service.servicename* section of your Genesys Mobile Services Application object.
- 6. Add options to the newly created section to create your resource list.
- 7. Add and set an allocation strategy option for this group.

End of Procedure

A sample resource list configuration is shown below.

GMS_Resources [199.79.227.1	44:2020] Properties	×
General Format Annex Security		1
	J 🔨 🖾 🔍 🚱	
Enter text here	Tenter text here	
abs dnis2 abs dnis1		
bc_allocation_strategy	"LOCAL"	
,		
Cano	cel <u>A</u> pply Hel	p

Configuring Services for Genesys Mobile Services

To complete your deployment, the following services need to be configured in your Genesys Mobile Services Application object:

- Genesys Mobile Services-Based Services
- Orchestration Server-Based Services
- Native Push (Notification)

Genesys Mobile Services-Based Services

The following services need to be configured for this category:

- 1. request-interaction
- 2. match-interaction
- 3. request-access

Required options are outlined below, with some sample values and screenshots provided to help you get started. For more information about configuring these services, see the Overview of Services Provided by Genesys Mobile Services.

request-interaction

GMS_810_CDREFRESH [199	0.79.227.144:	2020] Properties	
Annex	Security	Depende	Oction
General Tenants Serv	er Into Star	tinto Connections	Opeon
service.request-interaction	• 🕫 🗅	X 🗔 🖉 🖗 🕸	
Name *		Value	
Enter text here	7	Enter text here	7
ss type		"builtin"	
see ti		*30*	
service		Trequest-interaction"	
seresource_group		"DINIS"	
aspionde_code		100	
	Cancel	L Annia L	Help

Sample request-interaction options

Required request-interaction Options

Option Name	Option Value
type	builtin
ttl	30
service	request-interaction
resource_group	(Use the section name created earlier under your GMS_Resources Transactions object.)
provide_code	true

match-interaction

Annex	Sec	urity	Depend	dency
General Tenants	Server Info	Start	Info Connections	Option
		_		
📚 service.match-intera	action 💌	· 🦻 🗋	🗙 📴 🕸 🕼 🕸	
Name *			Value	
Enter text here		7	Enter text here	7
sype			"builtin"	
as service			"match-interaction"	
🚵 delete_match			"true"	

Sample match-interaction options

Required match-interaction Options

Option Name	Option Value
type	builtin
service	match-interaction
delete_match	true

request-access

Annex	Security	/	Dependen	icy
General Tenants	Server Info	Start Info	Connections	Option
service.request-acce	55 💌 (3 🗋 🗙 🗖	۵ 🙀 🖉	
Name *		Value		
Enter text here		P Enterte	xthere .	7
😹 type		"builtin"		
ses ti		*30*		
service		"reque:	t-access"	
esource_group		*DNIS*		
access_code_length	1	-4-		

Sample request-access options

Required request-access Options

Option Name	Option Value
type	builtin
ttl	30
service	request-access
resource_group	(Use the section name created earlier under your GMS_Resources Transactions object.)
access_code_length	4

Orchestration Server-Based Services

This page shows five sample services that can be configured:

- request-inbound-immediate
- request-inbound-delay
- request-inbound-poll
- request-outbound-immediate
- request-outbound-delay

These services are based off of sample SCXML applications that are distributed with Genesys Mobile Services. These sample files can be downloaded from the Sample Resources page. Applications should be customized to fit the business logic for applicable use cases, and then made available on

an application server. The URL is then specified as the *service* parameter in the applicable Options section of your Genesys Mobile Services Application object.

Other services can be configured by creating similar sections with the service parameter using a different URL that points to the SCXML application hosted on the application server of your choice.

Dependency: Genesys Mobile Services-based services should be configured first since Orchestration Server-based services are dependent on the match-interaction and request-access services.

request-inbound-immediate

 zervce redness sports anneget 	• <u>•</u> • • • • • • • • • • • • • • • • •
Name T Enter ten baro Suppe Su St Su service Su service, group Su service, group	Value

Sample request-inbound-immediate options

Required request-inbound-immediate Options

Option Name	Option Value
type	ors
ttl	30
service	http:// <your server="">/gms_samples/request- inbound-immediate.scxml</your>
resource_group	(Use the section name created earlier under your GMS_Resources Transactions object.)
provide_code	true

request-inbound-delay

Velae Enter too hoo "ge" "too "too food only enter the service of the service of the service "too "too" "too"	X

Sample request-inbound-delay options

Required request-inbound-delay Options

Option Name	Option Value
type	ors
ttl	30
service	http:// <your server="">/gms_samples/request- inbound-delay.scxml</your>
resource_group	(Use the section name created earlier under your GMS_Resources Transactions object.)
provide_code	true

request-inbound-poll



Sample request-inbound-poll options

Required request-inbound-poll Options

Option Name	Option Value
type	ors
ttl	30

Option Name	Option Value
service	http:// <your server="">/gms_samples/scxml/request- inbound-poll.scxml</your>
resource_group	(Use the section name created earlier under your GMS_Resources Transactions object.)
provide_code	true

request-outbound-immediate

	num Leonacione internet menult ratemental	
o service request outbound-imm	edate 💌 🔮 🗅 🗙 💭 🚱 🕼 🖉	
Name T	Value	T
Enter text here	P Enterheid here	4
- 30+	,03,	
and the	"We decalculture cancies in white we to do not impediate score"	
A RESOLUTION DECIMIN	CMC.	
provide_code	"tut"	

Sample request-outbound-immediate options

Option Name	Option Value	
type	ors	
ttl	30	
service	http:// <your server="">/gms_samples/scxml/request- outbound-immediate.scxml</your>	
resource_group	(Use the section name created earlier under your GMS_Resources Transactions object.)	
provide_code	true	

Required request-outbound-immediate Options

request-outbound-delay

lo service request outbound-deley 💽 🕈 🗋 🗙 🗔 🕐 😭 🕐		
Nene " Toros tará hao Im 500 Im 50 Im 50 Im 50 Im seoloca, golado Im seoloca, golado Im seoloca, golado	Value Ser	N
1		

Sample request-outbound-delay options

Required request-outbound-delay Options

Option Name	Option Value
type	ors
ttl	30
service	http:// <your server="">/gms_samples/scxml/request- outbound-delay.scxml</your>
resource_group	(Use the section name created earlier under your GMS_Resources Transactions object.)
provide_code	true

Native push (notification service)

Some services, such as request-inbound-delay or request-outbound-delay, send native push messages to the mobile device. For this to work, both general and device-specific settings need to be configured correctly in the *push* section of your Genesys Mobile Services Application object.

 \bigcirc Options set in the *push* section determine how all push notifications are handled by Genesys Mobile Services, regardless of which service is sending the notification.

Note that it is possible to configure this native push notification service for more than one type of device by using a comma-delimited string in the *pushEnabled* option. In this case, be sure to configure the mandatory options for all available device types.







GMS iOS notification.png

Common Device Settings:

- pushEnabled Device operating system.
- · defaultSubscriptionExpiration -

Mandatory iOS Device Settings:

- debug.apple.keystore Location of the debug keystore holding the certificates for push notification.
- debug.apple.keystorePassword Password for the debug keystore.
- apple.keystore Location of the production keystore holding the certificates for push notification.
- apple.keystorePassword Password for the production keystore.

Note: The specified location of the Apple iOS push keystore is environment specific, and must be configured based on your environment for iOS push notification to work. **Mandatory Android C2DM Device Settings:**

- android.senderEmail Name of a valid mail account. (Notifications are sent on behalf of this account.)
- android.senderPassword Password of mail account specified in android.senderEmail.
- android.senderAccountType Specified when initializing C2DM push service.
- android.source Specified when sending push notifications.
- android.collapseKey An arbitrary string used to collapse a group of like messages when the device is offline, so that only the last message gets sent to the client.

Mandatory Android GCM Device Settings:

- android.gcm.apiKey A valid Google API key value (Notifications are sent on behalf of this API key, see http://developer.android.com/guide/google/gcm/gs.html).
- android.gcm.retryNumber- Number of retries in case of service unavailability errors.

For additional detail about these options and the allowed values, see the push Section documentation. For general information about the push notification service, refer to Push Notification Service.

Deploy DFM Files for Orchestration Server-based Services

Included with your installation are special configuration files for Orchestration Server called DFM. These files define Genesys Mobile Services-specific SCXML constructs that are required for the execution of SCXML applications used within Orchestration Server-based Services. For the Orchestration Server-based Services to function correctly, the following DFM files need to be configured in your Orchestration Server Application object:

- Storage
- Notification
- Genesys Mobile-Based Services
- Geo-coding (for location based services)

The latest DFM definition files are available to download from the Genesys Mobile Services Sample Resources page. Details about deploying these DFM in your environment are provided in Deploy DFM Files in Orchestration Server. After deploying these DFM, you can use either an actual device with the demo application or an HTTP client (such as RestClient) to send API requests to Orchestation Serverbased services. Please refer to the Genesys Mobile Services API Reference for syntax of the requests.

Note: You must restart Orchestration Server and Genesys Mobile Services after deploying DFM files for the changes to take effect.

Start of Procedure

1. Download and extract the latest DFM sample files from the Genesys Mobile Services Sample Resources

page.

- 2. Start Configuration Manager.
- 3. In Configuration Manager, select *Environment* > *Applications*.
- 4. Locate and open the Application object for your Orchestation Server. (**Note:** This should be the same Application object you created a connection to when configuring your Genesys Mobile Services Application object.)
- 5. Select the *Options* tab.
- 6. Open the *dfm* section.
- 7. Create and configure one option for each DFM, using the option value to specify the file path. Details are provided in the table below.
- 8. Click OK to save your changes.
- 9. Restart Orchestration Server.
- 10. Restart Genesys Mobile Services.

End of Procedure

List of DFM Options for Orchestration Server

Service	Option Name	Option Value
Storage	gsgStorage	file://< <i>extraction path></i> \ors\drm\ cfg_GSG_storage.txt
Notification	gsgNotification	file:// <extraction path="">\ors\drm\ cfg_GSG_notification.txt</extraction>
Genesys Mobile-Based Services	gsgBasedServices	file://< <i>extraction path</i> >\ors\drm\ cfg_GSG_base_services.txt
Geo-coding	geocoding	file://< <i>extraction path</i> >\ors\drm\ cfg_YahooPlaceFinder.txt

Next Steps

With basic configuration complete, you are ready to consider more advanced configuration details.

- Understand and implement additional configuration steps to provide security and access control for your Genesys Mobile Services solution.
- Configure load balancing for your server.
- Review the sample applications included with this installation to help understand how you can leverage Genesys Mobile Services in your own development.
- Read the API Reference to get a better grasp on what type of services are available with Genesys Mobile Services.