



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 Task Routing Administrator's Guide

8/2/2025

Table of Contents

| | |
|---|----|
| Set up Genesys Task Routing | |
| Set up Genesys Task Routing | 4 |
| Integrate third-party applications | 7 |
| Explore the sample Designer application | 10 |

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

Related documentation:

-

Set up Genesys Task Routing

Learn how to set up Genesys Task Routing so you can submit and manage workitems from an external source like Salesforce.

Related documentation:

-

Contents

- [1 Define agents and agent groups](#)
- [2 Configure Agent Workspace](#)
- [3 Create a Capture Point](#)
- [4 Create a routing application](#)
- [5 Integrate third-party applications](#)
- [6 View dashboards in CX Insights](#)

Genesys Task Routing allows you to submit workitems to Genesys from an external source, such as Salesforce or NetSuite. Once these external workitems are entered in the Genesys system, you can treat them like other Genesys interactions. For example, you can route workitems to the best-skilled agent using a Designer application. You can also get interactions that were processed by an agent in Agent Workspace, including automated third-party application screen pop-ups. Reporting is available through Genesys CX Insights.

Genesys Task Routing is integrated with other Genesys solutions and it can support blended agents handling chat, email, voice and so on, at the same time they manage workitems.

Important

You should be able to complete most of the steps on your own, but sometimes you'll need help from Genesys. We'll let you know when a task requires extra help from a Genesys representative.

Define agents and agent groups

Use Agent Setup to define agents and agent groups you want to be able to handle workitems from your external system.

- Create agents
- Create agent groups
- Contact your Genesys representative to enable an Open Media channel.
- Define a capacity rule.
- Assign the Open Media channel and capacity rule to agents or agent groups.

Configure Agent Workspace

Configure Agent Workspace for the agents in your contact center.

- Configure Agent Workspace
- Contact your Genesys representative to set desktop options for the Open Media channel.

Create a Capture Point

Create a capture point to capture the workitems submitted from your external source.

- Contact your Genesys representative to create a Capture Point.
-

Create a routing application

Create a routing application in Designer to send workitems to agents based on skill, agent group, and so on. Genesys Task Routing includes a sample Designer application to show one of the possible scenarios. We recommend you use this as a starting point to create your own routing application.

- Explore the sample Designer application
 - Create a new Designer application
 - Link your application to your previously created capture point. **Note:** You can associate more than one Capture Point with the same routing application.
-

Integrate third-party applications

Create a third-party system adapter that can handle workitems that come from your external source and submit them to Genesys through the Genesys Task Routing API.

- Integrate third-party applications
 - Genesys Task Routing API
-

View dashboards in CX Insights

View historical reporting for Genesys Task Routing in CX Insights.

- View Genesys Task Routing dashboards
-

Integrate third-party applications

Contents

- [1 Configure screen pop-up for all media types](#)
- [2 Configure screen pop-up for only workitems](#)

Learn to submit workitems to Genesys from a third-party application using the Genesys Task Routing API and the Service Client API.

Related documentation:

-

A third-party application can interact with Genesys Task Routing through the Genesys Task Routing API and the Service Client API.

You'll need to develop a third-party system adapter that can create and stop workitems using the Genesys Task Routing API. It should also update and retrieve information about a workitem, if needed.

The typical workflow is as follows:

1. The third-party application creates a workitem through the Genesys Task Routing API.
2. The third-party application updates and/or retrieves the interaction through the Genesys Task Routing API (if needed).
3. The third-party application stops the interaction.

There are two ways for the third-party application to stop an interaction:

- Implicit (recommended) — stop by invoking Mark Done using the Service Client API. For this method, complete the following steps:
 1. Set up Agent Workspace to use the API.
 2. Disable the Mark Done button in Agent Workspace and instead invoke Mark Done through the Service Client API.
 3. You can also use the Service Client API to set the Disposition Code.
- Explicit — stop by using the Genesys Task Routing API. This requires additional configuration from your Genesys representative.

A sample application that demonstrates using the Genesys Task Routing API and the Service Client API is also available through your Genesys representative.

Configure screen pop-up for all media types

You can configure Genesys Task Routing to enable a third-party application screen pop-up inside Agent Workspace when an agent accepts a task. This approach works well when an agent is handling only workitems. See the Agent Setup documentation for details.

Configure screen pop-up for only workitems

You can configure Genesys Task Routing to enable a third-party application screen pop-up inside Agent Workspace when an agent accepts a task. This approach works well when an agent handles multiple media types and the third-party pop-up should only happen for workitems. You'll need to work with your Genesys representative to configure this approach.

Explore the sample Designer application

Contents

- [1 Prerequisites](#)
- [2 Import the application](#)
- [3 Configure the routing table](#)
- [4 Input values](#)
- [5 Application flow](#)
 - [5.1 Initialize phase](#)
 - [5.2 Assisted Service phase](#)

Explore the sample Designer application for Genesys Task Routing and learn how to modify it to work with your solution.

Related documentation:

-

Important

Contact your Genesys representative for access to the sample application.

Genesys Task Routing (GTR) includes a sample Designer application that you can use as a starting point to help meet your custom requirements for task routing. The application also includes a simplified Open Media Routing Data Table (called `OPEN_MEDIA_ROUTING_CONFIG`) to map the routing tasks based on agent, skill, agent group, and so on.

The sample application routes tasks to agents based on attributes you specify on the workitem interaction and information included in the Open Media Routing Data Table. It has four routing cascades that run in the following order:

1. GTR routes the task to the requested agent.
2. If not, GTR routes the task to an agent with the requested skill.
3. If not, GTR routes the task to the requested agent group.
4. If not, GTR routes the task to the default agent group.

The application works based on the assumption that all workitems are pre-classified in the third-party system before integrating with GTR. You can read more about this in [Input values](#).

Using the Open Media Routing Table, you can configure specific values for agents, skills, and agent groups, or set a default mapping instead. If the interactions have more than one row, you can define a default row as routing precedence. You can also override the default routing table by specifying the requested agent, skill, or agent group directly in the interaction user data.

Prerequisites

Complete the following steps before you get started:

1. Create a default agent group named **workitems**. The sample application uses **workitems** as the default agent group, so if you choose another name, you must update the application to use the new name.
2. Configure the third-party pop-up as described in [Integrate third-party applications](#).

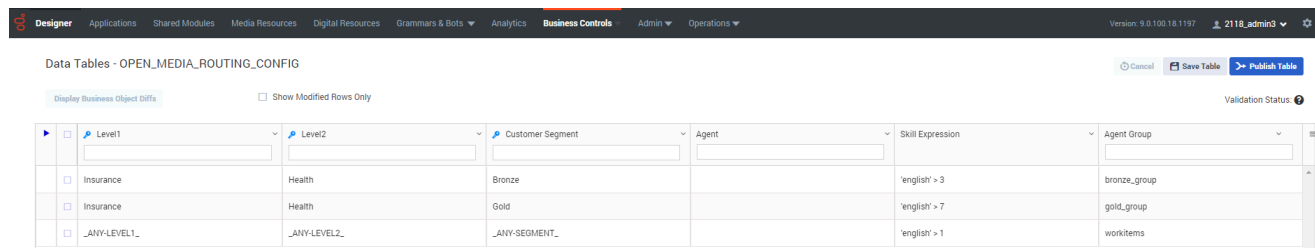
Import the application

Contact your Genesys representative to get the sample Designer application and the Open Media Routing Data Table. Once you have the application, follow the steps in the Designer documentation to import the application.

To complete the setup, edit the sample application and click the **Initialize** header. Set the default value for the **overrideKeyName** variable — you can get this value from the Transactions Override Attached Data Key in Agent Setup.

Configure the routing table

Once the application is imported, you can find the OPEN_MEDIA_ROUTING_CONFIG table under **Business Controls > Data Tables**.



The screenshot shows the Genesys Designer interface with the 'Business Controls' tab selected. Under 'Data Tables', the 'OPEN_MEDIA_ROUTING_CONFIG' table is displayed. The table has columns for Level1, Level2, Customer Segment, Agent, Skill Expression, and Agent Group. The first row is a header row with dropdown menus for each column. The second row shows 'Insurance' for Level1, 'Health' for Level2, 'Bronze' for Customer Segment, and 'English' for Skill Expression, with 'bronze_group' as the Agent Group. The third row shows 'Insurance' for Level1, 'Health' for Level2, 'Gold' for Customer Segment, and 'English' for Skill Expression, with 'gold_group' as the Agent Group. The fourth row shows '_ANY-LEVEL1_' for Level1, '_ANY-LEVEL2_' for Level2, '_ANY-SEGMENT_' for Customer Segment, and 'English' for Skill Expression, with 'workitems' as the Agent Group. There are buttons for 'Cancel', 'Save Table', and 'Publish Table' at the top right of the table view.

| Level1 | Level2 | Customer Segment | Agent | Skill Expression | Agent Group |
|--------------|--------------|------------------|-------|------------------|--------------|
| Insurance | Health | Bronze | | 'english' > 3 | bronze_group |
| Insurance | Health | Gold | | 'english' > 7 | gold_group |
| _ANY-LEVEL1_ | _ANY-LEVEL2_ | _ANY-SEGMENT_ | | 'english' > 1 | workitems |

In the image above:

- **Level1**, **Level2**, and **Customer Segment** form a compound key.
- **Agent**, **Skill Expression**, and **Agent Group** are searchable values in the table.
- The row with **_ANY-LEVEL1_**, **_ANY-LEVEL2_**, and **_ANY-SEGMENT_** enables the default mapping. If you change the values in this compound key, the application won't be able to use the default mapping. If you don't need the default mapping, you can leave the **Agent**, **Skill Expression**, and **Agent Group** fields blank in this row.

Input values

To work with the sample application, make sure your third-party system adapter adds the following user data keys on the workitem interaction:

- GTR_level1 — The value for **Level1**. Use this field to add any kind of classification that makes sense for your business. For example, this could represent a particular category in your business structure or a severity/escalation level.
- GTR_level2 — The value for **Level2**. This field provides another level of classification.
- CustomerSegment — The value for **Customer Segment**. This is a third classification level that represents how important the customer is for your business. For example, you could have values such as Bronze, Silver, and Gold.

Together these three attributes form a compound key that is used to get the **Agent**, **Skill Expression**, and **Agent Group** values from the routing table. If you don't provide values for these attributes, the sample application uses the default mapping (see Configure the routing table above).

You can also override the default routing table values for **Agent**, **Skill Expression**, and **Agent Group** by specifying values for these fields with the following user data keys:

- GTR_requestedAgent
- GTR_skillExpression
- GTR_requestedAgentGroup

Application flow

The sample application goes through two phases when routing tasks: Initialize and Assisted Service.

Initialize phase

The screenshot displays the 'Application Flow' designer interface. On the left, the 'Initialize' phase is expanded, showing a sequence of steps: 'Read variables from interaction userdata', 'Segmentation' (with a conditional 'If no Agent, Skill and Agent Group provided with interaction'), 'Read by Level1/Level2/Segment from Data Table', 'Segmentation' (with a conditional 'If search by Level1/Level2/Segment is failed'), 'Read defaults from Data Table', 'Segmentation' (with a conditional 'no agent requested in interaction'), 'Assign Agent as configured in Data Table', 'Segmentation' (with a conditional 'no Skill Expression requested in interaction'), 'Assign Skill Expression as configured in Data Table', 'Segmentation' (with a conditional 'no Agent Group requested in interaction'), and 'Assign Agent Group as configured in Data Table'. The 'Assisted Service' phase is also visible at the bottom.

On the right, the 'Properties - Initialize' panel is shown. It includes a description: 'This block or phase is typically used to setup variables for the application and initialize them. Assign blocks can be used to calculate expressions and assign their results to variables in this phase.' Below this, there are tabs for 'User Variables' and 'System Variables'. The 'User Variables' tab is active, showing a table of variables to be specified.

| Name | Default Value | Description | Secure | Trace | Delete |
|-------------------------|---------------------------|----------------------------|--------------------------|--------------------------|--------------------------|
| level1 | | Level 1 Name | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| level2 | | Level 2 Name | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| segment | | Customer Segment | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| skillExpression | | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| requestedAgent | | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| requestedAgentGroup | | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| overrideKeyName | 'overrideKey' | User must set value of Clo | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| thirdPartyIntegrationLi | 'GTR_3rdPartyIntegration' | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| isSearchValid | false | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| searchRowCount | 1 | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| defaultSkillExpression | | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| defaultRequestedAgen | | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| defaultRequestedAgen | | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

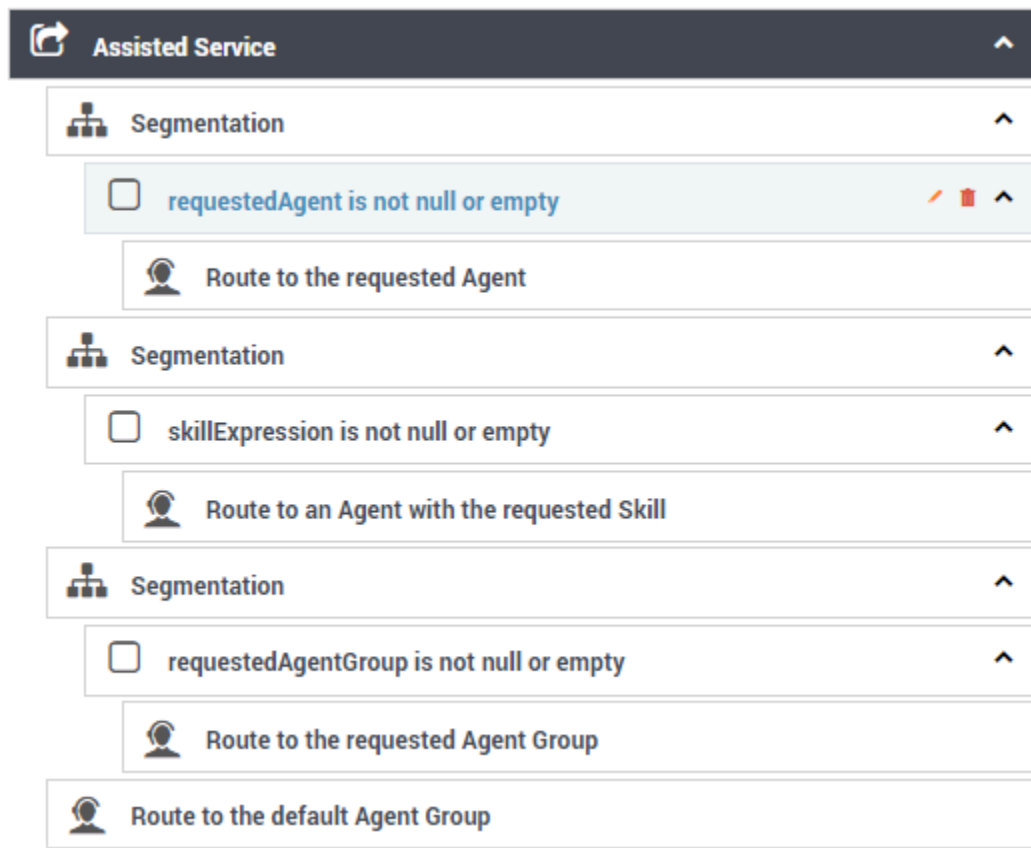
In the Initialize phase, the sample application goes through the following setup:

1. The **Read variables from interaction userdata** block reads GTR_level1, GTR_level2, and CustomerSegment from the user data on the interaction. Next, the application writes a key/value pair to the interaction user data to support the media type for third-party application pop-up in Agent Workspace:
 - overrideKeyName = thirdPartyIntegrationListObject (the default is overrideKey =

GTR_3rdPartyIntegration).

2. The **Segmentation** block checks to see **If no Agent, Skill or Agent Group provided with an interaction**.
 - If none exist, the application tries to read the missing parameter from the OPEN_MEDIA_ROUTING_CONFIG data table in the **Read by Level1/Level2/Customer Segment from Data Table** block.
 - If the key is not available or more than one row is present, the application sets the **isSearchValid** and **searchRowCount** variables.
3. If the previous iteration is not successful, the application tries to **Read defaults from Data Table** by getting the values from the **_ANY-LEVEL1_**, **_ANY-LEVEL2_**, and **_ANY-SEGMENT_** default row.
4. After going through the parameters, the application runs three independent **Segmentation** blocks that check if there was an agent, skill, or agent group requested in the interaction user data (see Input values). If not, the application sets the values configured in the data table.

Assisted Service phase



In the Assisted Service phase, the application follows four routing cascades in this order:

1. Route to the requested Agent

2. Route to an agent with the requested Skill
3. Route to the requested Agent Group
4. Route to the default Agent Group

The first three routing cascades are a sequence of **Segmentation** blocks that perform null checks and route to the appropriate agent or agent group. The last cascade, **Route to the default Agent Group**, uses the **workitems** agent group you created in Prerequisites. As mentioned previously, you can also set another default group name here to suit your custom needs.

All **Route** blocks use a 120-second timeout by default. You can adjust this value and enhance the routing data table and application to specify timeouts for the **Level1**, **Level2**, and **Customer Segment** fields.

Warning

Genesys does not recommend values below 120 seconds.