Predictive Routing Deployment and Operations Guide

Configuration Options

5/9/2025

Contents

Configuration Options

Genesys Predictive Routing uses configuration options to enable you to specify certain behaviors. Options relating to the AI Core Services (AICS) and the strategy subroutines are configured in a Transaction List object, configured in Genesys Administrator in the following location: **PROVISIONING > Routing/eServices > Transactions > List Objects > DEV > AgentScoring**.

> **Important**
>
> Before release 9.0.009.01, AI Core Services (AICS) was known as Journey Optimization Platform (JOP).

Agent State Connector (ASC) has its own `Application` object, where you configure options relating to specifically to ASC functionality.

- Predictive_Route_DataCfg Transaction List Object Options. Some functionality has multiple options controlling the desired behavior:
  - Agent Occupancy Options
  - Agent Holdout Options
  - Dynamic Interaction Priority Options
- Agent State Connector Configuration Options
  - ASC Log Options

Predictive_Route_DataCfg Transaction List Object Options

orig-connid-key
overload-control-timeout
prr-mode
scoring-token-expiration
send-user-event
udata-keys-to-exclude
use-action-filters
use-crm-query
use-double-selection
use-login-status
use-setreadycondition
vq-for-reporting

Agent Occupancy Options

Agent occupancy is the percentage of time that an agent is working while logged in, a service objective that can be specified when building a staffing forecast. Agent occupancy data is taken from Stat Server by URS using the SData function. Stat Server collects agent occupancy data using the StatAgentOccupancy statistic. The routing strategy filters agents by occupancy in the ScoreIdealAgent callback subroutine. The agent occupancy results are used to sort the agents in the target agent group; over-occupied agents drop down lower in the sorted list.

agent-occupancy-factor
agent-occupancy-threshold
use-agent-occupancy

Agent Holdout Options

Agent hold-out enables you to have an interaction wait a specified time, even when an agent has become available, if the available agent is has a low score for the interaction and there is a chance a better-matched agent might become available within the configured time window.

initial-threshold-timeout
score-base-threshold
threshold-relaxation-step
threshold-relaxation-timeout

Dynamic Interaction Priority Options

If an interaction has a low score for all targeted agents, it can stay in a queue for a long time. To avoid such situations, you can configure a schedule for incremental priority increases. The schedule is set once for each interaction processed by GPR. The following options control interaction priority increments.

> **Important**

> If you already use priority increments for the strategy into which you are inserting the GPR subroutines, you doo not need to configure these options. If you are using priority increments only for predictive routing, use the following options to configure it.

priority-increment
priority-init-interval
priority-interval
set-dynamic-priority

ab-test-time-slice

Specifies the length, in seconds, of the periods of time when Predictive Routing and skill-based routing are alternately turned on when you have set the **prr-mode** configuration option to `ab-test-time-sliced`. Genesys recommends that you do not set the value of this option to less than `600` seconds in a production environment.

- Configured in: Predictive_Route_DataCfg List object, **[default-predictor]** or **[<predictor_name>]** section
- Default value: 1741
- Valid values: Any positive integer
- Changes take effect: Immediately

context-id-key

Specifies the name of the user data key containing an ID for the current interaction, using which the Predictive Routing scoring engine can retrieve a record from an internal database of customer profiles (CRM database) and use features from the record to compute agents scores for the interaction.

To incorporate customer profile data into models for matching the agents, a copy of the CRM database must be uploaded into JOP before you train a predictor model. The URS ActivatePredictiveRouting subroutine attaches a `context_id` key to the scoring request body and provides the value of the user data key defined by this option as the `context_id` value.

If the returned customer ID is empty or you set the option value to `ANI`, the interaction ANI is used.

- Configured in: Predictive_Route_DataCfg List object, **[default-predictor]** or **[<predictor_name>]** section
- Default value: `ANI`
- Valid values:
  - `ANI`
  - A valid user data key name holding a customer ID

- Changes take effect: Immediately

default-agent-score

The option specifies the value the ScoreIdealAgent and isAgentScoreGood subroutines should use as the agent score for an interaction for those agents who belong to the target Agent Group but that GPR did not score. For example, an agent might be logged out, or in another status configured as unavailable, until after the scoring request it sent. If such an agent then becomes available before the interaction is routed, GPR assigns that agent the default score.

> ### Important
>
> This option functions differently depending on the release of URS Strategy Subroutines you have deployed:
>
> - In release 9.0.015.00 and higher, gpmAgentScore records the default score assigned to agents GPR did not score. The ScoreIdealAgent subroutine uses this value to sort the scores and the isAgentScoreGood subroutine compares it against any threshold you have configured to determine whether the agent is acceptable.
> - In release 9.0.014.04 and lower, the gpmAgentScore user data KVP always contains the value `0` for such agents. The score specified in this option is used only when URS is sorting the agents in the target group according to their scores.

- Configured in: Predictive_Route_DataCfg List object, **[default-predictor]** or **[<predictor_name>]** section
- Default value: The agent is assigned a score of `0`, which means that the agent is unlikely to receive an interaction from the queue.
- Valid values:
  - `max` - Use the maximum score calculated for an agent in the target agent group.
  - `median` - Use the median score calculated for the target agent group.
  - `global` - Use the average global score for the agents in the target group.
  - `min` - Use the minimum score calculated for an agent in the target agent group.
  - `0` - Use the value `0` as the score.
- Changes take effect: On the next interaction

emergency-scoring-token

Provides an emergency token in the event of continued authentication errors. It is intended for use only in scenarios where the strategy is unable to automatically update the token required to access

the Predictive Routing API.

> **Warning**
> This option should only be used in an emergency situation.

- Configured in: Predictive_Route_DataCfg List object, **[default]** section
- Default value: empty string
- Valid values: Any valid security token string
- Changes take effect: Immediately

format-as-map

The IRD subroutine ActivatePredictiveRouting_v<*version_number*> and the Composer subroutine ActivatePredictiveMatching now support two types of responses to, and score requests to, the Predictive Routing API, either containing both **list** and **list_ranks** fields or just the **list** field.

If set to `true`, the response and the score request to the Predictive Routing API contains two fields, **list** and **list_ranks**. The 'list' field contains a JSON dictionary with agent employee IDs as the keys and agent scores for the current interaction as the values. The **list_ranks** field contains a JSON dictionary with agent employee IDs as the keys and agents ranked according to their scores in the target group as values.

If set to `false`, the response and the score request to the Predictive Routing API contains only the **list** field. The value of this field is a JSON list object, where the items in the list are JSON dictionary objects. Each dictionary item contains the fields: **id** (agent employee ID), **score** (the score that agent has for the current interaction), and **score_type** (the type of a model, local or global, used to compute the score). The list is sorted by agent score in decreasing order.

- Configured in: Predictive_Route_DataCfg List object, **[default]** section
- Default value: `true`
- Valid values: `true, false`
- Changes take effect: Immediately

global-map-timeout

Defines the time period, in seconds, during which supporting information about an interaction (such as the predictor name and ID, the model name and ID, the Predictive Routing operation mode, and the interaction time in queue) are stored in the Universal Routing Server (URS) global map. If option value is set to `0`, the records are stored indefinitely.

- Configured in: Predictive_Route_DataCfg List object, **[default]** section
- Default value: `7200`
- Valid values: (integer) any non-negative integer
- Changes take effect: On the next interaction

> **Important**
>
> To improve URS performance, agent scores are stored in the URS global map with a timeout value of `0` (indefinitely). To remove them, you must call the PrrIxnCleanup subroutine after the interaction has been successfully routed.

jop-api-key

Specifies an access key that is used by the Agent State Connector or the ActivatePredictiveRouting subroutine in URS—depending on where the option is configured—to access the Genesys Predictive Routing API. To obtain the value of this option, open the **Accounts** tab in the Predictive Routing user interface and open your account (or create, to add a new account). The **API key** field appears in the **Account** configuration window. For details, see Settings: Configuring Accounts in the *Genesys Predictive Routing Help*.

- Configured in:
  - Predictive_Route_DataCfg List object, **[default]** section
  - Agent State Connector, **[default]** section
- Mandatory: yes
- Default value: none
- Valid values: Any valid AICS API key
- Changes take effect: After restart

jop-auth-url

Specifies the Genesys Predictive Routing API authentication endpoint URL. This value is the host name of the server on which the AI Core Services (AICS) component is installed followed by `/api/v2.0/authenticate`.

- Configured in: Predictive_Route_DataCfg List object, **[default]** section
- Mandatory: yes
- Default value: none

- Valid values: (string) A valid AICS authentication endpoint URL, in the following format: `<aics_server_host_name>/api/v2.0/authenticate`
- Changes take effect: immediately

jop-logging-url

Defines the URL for logging the interaction routing score log and outcome results to the Predictive Routing web application REST API.

- Configured in: Predictive_Route_DataCfg List object, **[default]** section
- Default value: none
- Valid values: (string) any valid URL
- Changes take effect: On the next interaction

jop-password

Specifies a user password valid for use with the Genesys Predictive Routing.

- Configured in:
  - Predictive_Route_DataCfg List object, **[default]** section
  - Agent State Connector, **[default]** section
- Mandatory: yes
- Default value: none
- Valid values: (string) The password for any valid Predictive Routing user
- Changes take effect: After restart

jop-scoring-url

The ActivatePredictiveRouting strategy subroutine in URS uses the URL defined by this option as the HTTP address to send scoring requests to AI Core Services (AICS) Scoring REST API. This URL should be the value for the **jop-base-url** option with `<predictor_name>/score` appended.

You can locate the predictor ID in messages returned from the GPR API or in the browser URL address when you are in the GPR application with the page for the desired predictor open.

- Configured in: Predictive_Route_DataCfg List object, **[default-predictor]** or **[<predictor_name>]** section

- Default value: none
- Valid values: (string) a valid AICS scoring endpoint URL + a valid predictor ID
- Changes take effect: On the next interaction processed

jop-username

Specifies a user's username to access Genesys Predictive Routing.

- Configured in:
  - Predictive_Route_DataCfg List object, **[default]** section
  - Agent State Connector, **[default]** section
- Mandatory: yes
- Default value: none
- Valid values: (string) Any valid email address registered with Predictive Routing.
- Changes take effect: After restart

login-status-expression

If you set the value of the **use-login-status** option to `true`, the value of the **login-status-expression** option is added to the action_filters expression in the ActivatePredictiveRouting_v3 subroutine when the scoring request is created.

- Configured in: Predictive_Route_DataCfg List object, **[default-predictor]** or **[<predictor_name>]** section
- Default value: no default value
- Valid values:
  - &((loginStatus>0&loginStatus<23)|loginStatus>23) - Instructs the scoring engine to evaluate scores for those agents identified as part of the target group by a skill expression or an Agent Group name who are logged into the voice channel.
  - &(loginStatus=4|loginStatus=9) - Instructs the scoring engine to evaluate scores for those agents identified as part of the target group by a skill expression or an Agent Group name who are ready to accept an interaction, or have status AfterCallWork on the voice channel.
- Changes take effect: On the next interaction

log-to-api

Specifies whether logging is enabled to the Predictive Routing application REST API from the routing strategy. If the option value is set to `true`, the context of the interaction is submitted to Predictive Routing when the PrrIxnCompleted subroutine is called, before interaction is routed to an agent.

- Configured in: Predictive_Route_DataCfg List object, **[default]** section
- Default value: `false`
- Valid values:
  - `true`: The context of the interaction is submitted to the Predictive Routing application when the PrrIxnCompleted subroutine is called before the interaction is routed to an agent.
  - `false`: Logging is not enabled.
- Changes take effect: On the next interaction

max-score

Defines the maximum score that an agent can be assigned for an interaction. The value of this option is used by the `ScoreIdealAgent` callback function to re-scale the agent score as the distance from an ideally matched agent for the interaction (assumed by URS to be `0`).

The value you set should correspond to the largest possible value returned by this Predictor from the scoring engine. To function properly, this value must be consistent with the value configured for the Predictor **Score expression** field. Because the GPR scoring engine and URS have different scales, you might need to adjust returned scoring values using the **Score expression** field in the Predictor configuration. See the instructions for how to configure this field in Creating and Updating Predictors in the *Genesys Predictive Routing Help* for more information.

To take advantage of the most precise values, set **max-score** to `10000` and the value for Score expression in the Predictor configuration to `10000 * p_score`. (*p_score* is a term used in the GPR documentation to indicate the raw score returned from the scoring engine. It is not in any way derived from or related to the statistical term *P value*.) For example, if scores range from -4 to 10, use the following **p_score** - `((p_score + 5) / 16) * 100`.

- Configured in: Predictive_Route_DataCfg List object, **[default-predictor]** or **[<predictor_name>]** section
- Default value: `100`
- Valid values: (integer) 1 - `<max>`
- Changes take effect: On the next interaction processed

orig-connid-key

Defines a user data key that the Predictive Routing strategy must attach on initialization. It holds the original connection ID of an interaction, which is used to uniquely identify the interaction for the scoring engine. The ActivatePredictiveRouting subroutine checks for the presence of this key when it starts processing an interaction.

- Configured in: Predictive_Route_DataCfg List object, **[default]** section
- Default value: None
- Valid values: Any valid user data key holding the original interaction connection ID
- Changes take effect: Immediately
- Mandatory: Yes

ConnIDs in Consult Calls
Agents can make consult calls to the route point using GPR to identify target agents. In such cases, your T-Server/SIP Server configuration determines which ConnID is added in the scoring request and recorded in the score log. Depending on your configuration, the ConnID could be from either the main call or the consult call.

The following table explains the different scenarios possible depending on your T-Server/SIP Server option settings.

Option Name	Values	Result for GPR
consult-user-data	separate	The ConnID key should be explicitly attached in both main call and consult call user data *before* the GPR subroutines are invoked. If the keys are attached to both calls, then the GPR score request, and the score log, report the main call and the consult call as separate, each with its respective ConnID.
consult-user-data	inherited joint	The ConnID key must be explicitly attached to the main call by the strategy that invokes the GPR subroutines. The main call and consult call are treated as a unit and both requests log only the ConnID for the main call.

overload-control-timeout

Defines a timeout value that sets the maximum delay, in milliseconds, between the moment when URS receives an Event from T-Server and when URS starts to process the Event in the strategy. If the delay is greater than the value set in this option, Predictive Routing considers the URS application overloaded and temporarily turns off. Once the URS overload ends and the strategy is processing

events within the limit defined by this timeout, Predictive Routing restarts.

- Configured in: Predictive_Route_DataCfg List object, **[default]** section
- Default value: `1000`
- Valid values: Any positive integer
- Changes take effect: Immediately

prr-mode

Specifies whether an instance of Predictive Routing should run as a production instance or as a test instance.

- Configured in: Predictive_Route_DataCfg List object, **[default-predictor]** or **[<predictor_name>]** section
- Default value: `off`
- Valid values:
  - `prod` - All the interactions that pass through the ActivatePredictiveRouting strategy subroutine are processed using Predictive Routing.
  - `off` - No interactions use Predictive Routing.
  - `ab-test-time-sliced` - The periods of time when Predictive Routing and skill-based routing are alternately turned on. The duration of each period is configured in the [<*predictor_name*>].**ab-test-time-slice** configuration option in the Predictive_Route_DataCfg `Transactions List` object.
  - `dry-run` - Predictive Routing scores agents for your interactions, but does not use the scores for routing.
- Changes take effect: Immediately

scoring-token-expiration

If configured, overrides the default token expiration time of 43200 seconds. For example, if set to `3600`, the token expires in the URS memory map in one hour, and a new token is requested from the JOP Scoring Engine.

- Configured in: Predictive_Route_DataCfg List object, **[default]** section
- Default value: `43200`
- Valid values: Any positive integer
- Changes take effect: Immediately

send-user-event

When set to `true`, the routing strategy used with Predictive Routing sends the EventUserEvent TEvent, which includes the following attributes:

- AttributeThisDN with a value indicating the virtual queue where the strategy is executed. This is set in the **vq-for-reporting** option.
- AttributeUserData containing the Predictive Routing-specific key-value pairs which provide the foundation for reports on routing outcomes presented in Genesys Interactive Insights/GCXI.

The KVP data is stored in Genesys Info Mart, and is then available to the Genesys reporting suite and to Predictive Routing, which can use this KVP data to refine predictor and model performance.

For more information on creating reports based on Predictive Routing data, see Deploying: Integrating with Genesys Reporting.

- Configured in: Predictive_Route_DataCfg List object, **[default]** section
- Default value: `false`
- Valid values: `true, false`
- Changes take effect: Immediately

udata-keys-to-exclude

Use this option to exclude unnecessary user data keys from the scoring context.

- Configured in: Predictive_Route_DataCfg List object, **[default]** section
- Default value: no default value
- Valid values: a list of KVP names to be excluded, separated by commas and no spaces
- Changes take effect: On the next interaction

use-action-filters

- Configured in: Predictive_Route_DataCfg List object, **[default-predictor]** or **[<predictor_name>]** section
- Default value: `true`
- Valid values:
  - `true` - URS uses a skill expression or Agent Group names taken from the **action_filters** field in the scoring request.
  - `false` - URS checks with the Stat Server for the target list of agents, as specified in the **login-**

status-expression option, and adds the target Agent IDs to the scoring request.

- Changes take effect: Immediately

> **Important**
>
> If **login-status-expression** is set to `&(loginStatus=4|loginStatus=9)`, indicating that the agents who are in the Ready state or ACW state (for voice calls) are the designated target agents, then the GetActionFilters subroutine uses a custom statistic called RStatGPRAgentsReadyOrACWvoice. This custom statistic is provided in the **object.kvlt** file in the URS Strategy Subroutines IP.

use-crm-query

Option name reserved for future use.

- Configured in: Predictive_Route_DataCfg List object, **[default-predictor]** or **[<predictor_name>]** section
- Default value: `true`
- Valid values:
  - `true`
  - `false`
- Changes take effect: Immediately

use-double-selection

Specifies whether URS uses a double selection mechanism, applying a custom statistic when agents have the same score to select the target agent for an interaction.

If the Predictive Routing routing solution is configured to use the agent hold-out feature (the **use-setreadycondition** option is set to `true`) and the **use-double-selection** option is set to `false`, when two or more agents are in ready state and have the same score for an interaction, the target agent for an interaction is selected at random. If the **use-double-selection** option is set to `true`, URS selects a target agent from a group of agents with equal scores based on a predefined statistic. This is a statistic passed as an argument to the SelectDN function by the routing strategy or one defined in an IRD routing block.

- Configured in: Predictive_Route_DataCfg List object, **[default]** section
- Default value: `false`
- Valid values:

  - `true`: Predictive Routing uses the double selection method.
  - `false`: The double selection method is turned off.
- Changes take effect: Immediately

use-login-status

Set the value of this option to `true` to have the value of the **login-status-expression** option added to the action_filters expression in the ActivatePredictiveRouting_v3 subroutine when the scoring request is created.

> **Important**
>
> Genesys recommends that you set this option to `true` and provide a valid value for **login-status-expression** to reduce the number of agents for whom scores are evaluated. The value `false` should be used only for debug and troubleshooting purposes in a staging environment.

- Configured in: Predictive_Route_DataCfg List object, **[default-predictor]** or **[<predictor_name>]** section
- Default value: `false`
- Valid values: `false, true`
- Changes take effect: On the next interaction

use-setreadycondition

If option is set to `true`, the strategy executes calls to the isAgentScoreGood subroutine, which temporarily removes low-scoring agents from consideration for routing. If option is set to `false`, the strategy does not execute calls to the isAgentScoreGood subroutine.

> **Important**
>
> This option takes effect only when the **prr-mode** option is set to `prod` for the same predictor.

- Configured in: Predictive_Route_DataCfg List object, **[default-predictor]** or **[<predictor_name>]** section
- Default value: false

- Valid values: true, false
- Changes take effect: On the next interaction

vq-for-reporting

Indicates the virtual queue or DN where URS sends the Genesys Predictive Routing (GPR) user event data describing the routing decision made for the interaction. The user event data, in the form of key-value pairs, is attached to EventUserEvent in the AttributeUserData attribute. This should be the same value as AttributeThis DN in the EventUserEvent event.

For more information on creating reports based on Predictive Routing data, see Deploying: Integrating with Genesys Reporting.

- Configured in: Predictive_Route_DataCfg List object, **[default]** section
- Default value: no default value
- Valid values: Any valid virtual queue or DN name
- Changes take effect: Immediately

agent-occupancy-factor

If you set the value of the **use-agent-occupancy** option to `true`, and the value of the agent occupancy statistic is higher than the threshold specified in the **agent-occupancy-threshold** option, the ScoreIdealAgent subroutine multiplies the score received for an agent for the current interaction by a coefficient defined by this option.

- Configured in: Predictive_Route_DataCfg List object, **[default]** section
- Default value: `0.5`
- Valid values: Float number between `0.0` and `1.0`
- Changes take effect: On the next interaction

agent-occupancy-threshold

If you set the value of the **use-agent-occupancy** option to `true`, the isAgentScoreGood subroutine compares the value of the occupancy statistic with the value you set in this option. If the occupancy value is higher than the specified threshold, the subroutine multiplies the score received for an agent for the current interaction by a coefficient defined in the **agent-occupancy-factor** option.

- Configured in: Predictive_Route_DataCfg List object, **[default]** section

- Default value: `0`
- Valid values: Any non-negative integer
- Changes take effect: On the next interaction

use-agent-occupancy

The value you set for this option determines whether the isAgentScoreGood subroutine checks for agent occupancy. If you set the value to `true`, the subroutine compares the value of the occupancy statistic with the value you set in the **agent-occupancy-threshold** option. If the occupancy value is higher than the specified threshold, the subroutine multiplies the score received for an agent for the current interaction by a coefficient defined in the **agent-occupancy-factor** value.

- Configured in: Predictive_Route_DataCfg List object, **[default]** section
- Default value: `false`
- Valid values: `true, false`
- Changes take effect: On the next interaction

initial-threshold-timeout

Defines a timeout, in seconds, during which the `isAgentScoreGood` URS callback function uses an initial minimum agent score, defined by the **score-base-threshold** option, to match agents to an interaction. After this timeout expires, the minimum score required to allow an agent to handle the interaction is gradually decreased.

- Configured in: Predictive_Route_DataCfg List object, **[default-predictor]** or **[<predictor_name>]** section
- Default value: `0`
- Valid values: (integer) `0` - `<max>`
- Changes take effect: On the next interaction processed

score-base-threshold

This option defines the initial minimum agent score required for an agent to be considered a match for an interaction. After the timeout defined by the **initial-threshold-timeout** option expires, the minimum score required to handle the interaction is gradually decreased. If you set the value to `0`, no initial minimum score is required and agents with any score are considered for an interaction.

- Configured in: Predictive_Route_DataCfg List object, **[default-predictor]** or **[<predictor_name>]**

section

- Default value: `0`
- Valid values: (integer) `0` - `<max>`
- Changes take effect: On the next interaction processed

threshold-relaxation-step

Defines an increment by which, while an interaction remains queued, the minimum agent score required to match the interaction is decreased after each period defined by the value of the **threshold-relaxation-timeout** option, following the initial period defined by the **initial-threshold--timeout** option.

- Configured in: Predictive_Route_DataCfg List object, **[default-predictor]** or **[<predictor_name>]** section
- Default value: 1
- Valid values: (integer) 1 - `<value of the **max-score** option>`
- Changes take effect: On the next interaction processed

threshold-relaxation-timeout

This option defines a timeout, in seconds, after which the minimum agent score required for matching an interaction is decreased by the amount defined by the value of the **threshold--relaxation-step** option.

- Configured in: Predictive_Route_DataCfg List object, **[default-predictor]** or **[<predictor_name>]** section
- Default value: 1
- Valid values: (integer) 1 - `<max>`
- Changes take effect: On the next interaction processed

priority-increment

Specifies the increment by which priority is increased each time.

- Configured in: Predictive_Route_DataCfg List object, **[default-predictor]** or **[<predictor_name>]** section
- Default value: 1

- Valid values: (integer) any integer
- Changes take effect: On the next interaction

priority-init-interval

Controls the time interval, in seconds, the strategy waits before starting to increment priority for a queued interaction.

- Configured in: Predictive_Route_DataCfg List object, **[default-predictor]** or **[<predictor_name>]** section
- Default value: `300`
- Valid values: (integer) any non-negative integer
- Changes take effect: On the next interaction

priority-interval

Specifies the time period, in seconds, between priority increments for a queued interaction.

- Configured in: Predictive_Route_DataCfg List object, **[default-predictor]** or **[<predictor_name>]** section
- Default value: `10`
- Valid values: (integer) any integer greater than 5
- Changes take effect: On the next interaction

set-dynamic-priority

Specifies whether dynamic priority interaction handling is enabled and handled in the GPR subroutines. When set to `true` interaction priority is incremented based on the settings configured for the other priority options. When set to `false`, dynamic priority interaction handling is not set by the Predictive Routing subroutines. If dynamic priority parameters are set elsewhere in the strategy, the option must be set to `false`.

- Configured in: Predictive_Route_DataCfg List object, **[default-predictor]** or **[<predictor_name>]** section
- Default value: `false`
- Valid values: `false, true`

- Changes take effect: On the next interaction

Agent State Connector Configuration Options

[default] Section

Configure the following options in the **[default]** section:

agents-batch-size
auto-schema-discovery
cfg-reading-threads-size
cfg-retry-request-attempts
confserv-monitoring-reconnect-count
confserv-monitoring-reconnect-min
filter-by-groups
filter-by-skills
ignore-ascii-characters
ignore-employee-ids
ignore-person-annex-sections
include-person-annex-sections
include-groups
include-skills
jop-api-key
jop-base-url
jop-password
jop-username
jop-update-thread-wait-timeout
reset-jop-on-startup
skip-groups
ss-custom-statistic-name
ss-subscription-timeout
ss-monitoring-reconnect-count
ss-monitoring-reconnect-min
stat-srv-ws-conn
timebased-statistic-interval
threads-max-size

agents-batch-size

Defines the maximum number of agent configuration profiles that can be submitted in a single HTTP request to AI Core Services (AICS).

> **Important**
> Increasing this option value might reduce Agent State Connector startup time. However, setting it too high might cause the size of HTTP requests to become greater than 10 Mb, which is the default maximum size for an HTTP request body that AICS accepts by default.

- Configured in: Agent State Connector object, **[default]** section
- Default value: `500`
- Valid values: Integers from `1`–`1000`
- Changes take effect: On restart

auto-schema-discovery

Enables Agent State Connector (ASC), at startup, to check whether an Agent Profile schema is present. If there is no Agent Profile schema uploaded, ASC creates a schema. If an Agent Profile schema has already been uploaded, ASC checks the schema to validate that it is correctly structured. If there is no schema uploaded and ASC cannot create one, or if the uploaded schema is invalid, ASC generates an alarm message and shuts down.

- Configured in: Agent State Connector object, **[default]** section
- Default value: `true`
- Valid values: `true`, `false`
- Changes take effect: On restart

cfg-reading-threads-size

Enables you to specify whether to read agents and groups from Configuration Server using a multi-threading approach.

- Configured in: Agent State Connector object, **[default]** section
- Default value: `100`
- Valid values: Integers from `1`–`2000`
- Changes take effect: On restart

cfg-retry-request-attempts

Used to specify the number of times Agent State Connector (ASC) tries to get updated agent and agent group data from Configuration Server if the first try is unsuccessful.

> **Important**
> Genesys recommends that you do not set the value for this option higher than 5.

- Configured in: Agent State Connector object, **[default]** section
- Default value: 3
- Valid values: Any positive integer between 1 - 10
- Changes take effect: On restart
- Introduced in: 9.0.006.03

confserv-monitoring-reconnect-count

Specifies the maximum number of reconnect attempts to Configuration Server before ASC generates log event 60706, for which you should set an alarm. To be exact, if ASC detects a switchover or disconnection the number of times set in this option during the time period set in the **confserv-monitoring-reconnect-min** option, ASC generates the log event. The cancel event for this alarm should be 60707.

- Configured in: Agent State Connector object, **[default]** section
- Default value: 1
- Valid values: Integers from 10–1000
- Changes take effect: On restart

confserv-monitoring-reconnect-min

Specifies a time interval, in minutes, that ASC uses when monitoring multiple Configuration Server switchover events. If ASC detects as many switchovers or disconnects as specified in the **confserv-monitoring-reconnect-count** during the time period configured in this option, ASC generates log event 60702, for which you should set an alarm.

- Configured in: Agent State Connector object, **[default]** section
- Default value: 1
- Valid values: Integers from 10–1000

- Changes take effect: On restart

filter-by-groups

If the ASC configuration contains non-empty values for the **filter-by-skills** and/or **filter-by-groups** configuration options, ASC subscribes to Stat Server for agent statistics only for the agents included in the specified Agent Groups or those satisfying the configured skill expression. If both options are configured, the agents are subscribed for statistics if they *either* satisfy the skill expression specified in the **filter-by-skills** option *or* are included in one of the Agent Groups specified in the **filter-by-groups** option.

  - This functionality enables you to limit the number of agents monitored by GPR or to use GPR in environments where multiple Stat Servers are deployed to monitor different groups of agents.

- Configured in: Agent State Connector object, **[default]** section
- Default value: no default value
- Valid Values: A comma-separated list of valid Agent Group names
- Changes take effect: On restart

filter-by-skills

If the ASC configuration contains non-empty values for the **filter-by-skills** and/or **filter-by-groups** configuration options, ASC subscribes to Stat Server for agent statistics only for the agents included in the specified Agent Groups or those satisfying the configured skill expression. If both options are configured, the agents are subscribed for statistics if they *either* satisfy the skill expression specified in the **filter-by-skills** option *or* are included in one of the Agent Groups specified in the **filter-by-groups** option.

  - This functionality enables you to limit the number of agents monitored by GPR or to use GPR in environments where multiple Stat Servers are deployed to monitor different groups of agents.

> **Important**
>
> The & (ampersand) and | (pipe) operators are not supported in skill expressions used as the value of the **filter-by-skills** option. The skill expression must include only a single valid skill name.

- Configured in: Agent State Connector object, **[default]** section
- Default value: no default value
- Valid Values: A comma-separated list of valid agent skills

- Changes take effect: On restart

ignore-ascii-characters

Enables you to specify how Agent State Connector (ASC) handles Agent Profile columns with the following unsupported ASCII characters: `[Space], -, <, >`.

- To have ASC remove the specified characters for Agent Profile schema columns, but add the affected columns to the schema, set the option to `true`.
- To have columns with the specified characters entirely omitted from the schema, set the option to `false` (the default value), .

> **Important**
> Columns with other unsupported characters continue to be omitted from the schema. For a complete list of unsupported characters, see Configuring Agent Profiles.

- Configured in: Agent State Connector object, **[default]** section
- Default value: `false`
- Valid Values: `true, false`
- Changes take effect: On restart

ignore-employee-ids

Enables you to instruct Agent State Connector to skip processing for specified employee IDs. For example, the employee configuration might include symbols, such as $, that the database cannot process.

Additional use cases:

- An employee has an unusually large profile, which would create an unacceptable impact on predictive routing performance.
- An agent profile contains some data that produces an error when it is submitted to for predictive routing analysis.

- Configured in: Agent State Connector object, **[default]** section
- Default value: none
- Valid values: Valid employee ID numbers, separated by commas
- Changes take effect: On restart

ignore-person-annex-sections

Specifies which sections on the **Annex** tab of a `Person` configuration object the Agent State Connector (ASC) should skip when uploading the agent profile to AI Core Services. By default, ASC skips the sections related to Genesys Interaction Workspace.

- Configured in: Agent State Connector object, **[default]** section
- Default value: `interaction-workspace,interaction-workspace-recents,interaction-workspace-favorites`
- Valid values: One, or comma-separated list if more than one, valid section names on the Annex of a `Person` configuration object
- Changes take effect: On restart

Use this option to reduce startup time by stopping ASC from loading unnecessary data.

include-person-annex-sections

Specifies which sections on the **Annex** tab of a `Person` configuration object the Agent State Connector (ASC) should take information from when uploading the agent profile to AI Core Services. All unspecified sections are skipped. If both this option and **ignore-person-annex-sections** are configured, ASC disregards the value set for **ignore-person-annex-sections** and loads information only from the sections specified in the **include-person-annex-sections** option.

- Configured in: Agent State Connector object, **[default]** section
- Default value: none
- Valid values: One, or comma-separated list if more than one, valid section names on the Annex of a `Person` configuration object. For example, `interaction-workspace,interaction-workspace-recents,interaction-workspace-favorites`.
- Changes take effect: On restart

Use this option to reduce startup time by preventing ASC from loading unnecessary data.

include-groups

Use this option to specify a list of agent groups for ASC to monitor. This list is a subset of the total list of groups present in agent profiles. ASC ignores all groups except those you list. To monitor all groups, leave the option value empty (the default setting).

For example, you might set the value of this option as follows to have ASC monitor only two groups: `"GROUP1,GROUP2"`

- Configured in: Agent State Connector object, **[default]** section
- Default value: ""
- Valid Values: A comma-separated list of valid agent group names
- Changes take effect: On restart

include-skills

Use this option to specify a list of skills for ASC to monitor. This list is a subset of the total list of skills present in agent profiles. ASC ignores all skills except those you list. To monitor all skills, leave the option value empty (the default setting).

For example, you might set the value of this option as follows to have ASC monitor only two skills: `"CLOSING_AN_ACCOUNT,SALES"`

- Configured in: Agent State Connector object, **[default]** section
- Default value: ""
- Valid Values: A comma-separated list of valid skill names
- Changes take effect: On restart

jop-api-key

Specifies an access key that is used by the Agent State Connector or the ActivatePredictiveRouting subroutine in URS—depending on where the option is configured—to access the Genesys Predictive Routing API. To obtain the value of this option, open the **Accounts** tab in the Predictive Routing user interface and open your account (or create, to add a new account). The **API key** field appears in the **Account** configuration window. For details, see Settings: Configuring Accounts in the *Genesys Predictive Routing Help*.

- Configured in:
  - Predictive_Route_DataCfg List object, **[default]** section
  - Agent State Connector, **[default]** section
- Mandatory: yes
- Default value: none
- Valid values: Any valid AICS API key
- Changes take effect: After restart

jop-base-url

Specifies the common substring of Genesys Predictive Routing API endpoint URLs. This value is the host name of the server on which the AICS component is installed, followed by `/api/v2.0`.

To use HTTPS, specify `https://` in your base URL string.

- Configured in: Agent State Connector object, **[default]** section
- Mandatory: yes
- Default value: none
- Valid values: (string) A valid common substring of AICS endpoint URLs, in the following format:

`https://<aics_server_host_name>/api/v2.0` or `http://<aics_server_host_name>/api/v2.0`

- Changes take effect: After restart

jop-password

Specifies a user password valid for use with the Genesys Predictive Routing.

- Configured in:
  - Predictive_Route_DataCfg List object, **[default]** section
  - Agent State Connector, **[default]** section
- Mandatory: yes
- Default value: none
- Valid values: (string) The password for any valid Predictive Routing user
- Changes take effect: After restart

jop-username

Specifies a user's username to access Genesys Predictive Routing.

- Configured in:
  - Predictive_Route_DataCfg List object, **[default]** section
  - Agent State Connector, **[default]** section
- Mandatory: yes
- Default value: none

- Valid values: (string) Any valid email address registered with Predictive Routing.
- Changes take effect: After restart

jop-update-thread-wait-timeout

Specifies the thread waiting timeout, in milliseconds, applied to the AI Core Services (AICS) subscribe process. This timeout can prevent a polling loop from taking up unacceptable CPU bandwidth at busy periods.

- Configured in: Agent State Connector object, **[default]** section
- Default value: `50`
- Valid values: Any positive integer
- Changes take effect: On restart

reset-jop-on-startup

> **Important**
> This option is removed in Agent State Connector (ASC) release 9.0.006.08 and higher. To delete agents, delete your current agent profile schema in the Predictive Routing application, and then upload an updated schema.

Specifies whether agent profiles are recreated at startup. These profiles are used to query Stat Server about agent statistics.

- Configured in: Agent State Connector object, **[default]** section
- Default value: `true`
- Valid values:
  - `true`—When Agent State Connector (ASC) starts up, it deletes all agent profiles previously stored in the AI Core Services (AICS) database and recreates agent profiles from the Person data from Genesys Configuration Server for the Tenants that ASC monitors.
  - `false`—ASC uses the previously-stored agent profiles.
- Changes take effect: On restart

skip-groups

If this parameter set to `true`, ASC ignores all Configuration Server data about groups and events connected with updates to groups.

Set this option to `true` if the scoring request `action_filters` field contains only the skill expression filters and does not include filters by Agent Group names. ASC then skips reading Agent Group information from Configuration Server, which should significantly reduce ASC initialization time on start up.

- Configured in: Agent State Connector object, **[default]** section
- Default value: `false`
- Valid values: `true, false`
- Changes take effect: On restart
- Introduced in: 9.0.006.03

ss-custom-statistic-name

Use this option to specify the name of a custom statistic that ASC should read from Stat Server. By default, ASC subscribes for CurrentAgentState data.

To use this functionality, you must first configure the custom statistic in Stat Server before you can specify it as the value for the **ss-custom-statistic-name** option. Refer to Create a Custom Stat Server Statistic in the *Predictive Routing Deployment and Operations Guide* for complete instructions.

- Configured in: Agent State Connector object, **[default]** section
- Default value: `CurrentAgentState`
- Valid values: A string consisting of any valid custom statistic name
- Changes take effect: On restart

ss-subscription-timeout

Specifies a timeout, in milliseconds, between each subscription to avoid overloading Stat Server.

> **Important**
>
> - A series of Stat Server switchovers from primary to backup indicates that Stat Server is overloaded. If you see this pattern, increase the value of this option.

- To resolve this issue, you might also need to adjust the value of the **threads-max-size** option.

- Configured in: Agent State Connector object, **[default]** section
- Default value: `10`
- Valid values: Integers from `10`–`1000`
- Changes take effect: On restart

ss-monitoring-reconnect-count

Specifies the maximum number of reconnect attempts to Stat Server before ASC generates log event 60703, for which you should set an alarm. To be exact, if ASC detects a switchover or disconnection the number of times set in this option during the time period set in the **ss-monitoring-reconnect-min** option, ASC generates the log event. The cancel event for this alarm should be 60704.

- Configured in: Agent State Connector object, **[default]** section
- Default value: `4`
- Valid values: Integers from `1`–`10000`
- Changes take effect: On restart

ss-monitoring-reconnect-min

Specifies a time interval, in minutes, that ASC uses when monitoring multiple Stat Server switchover events. If ASC detects as many switchovers or disconnects as specified in the **ss-monitoring-reconnect-count** during the time period configured in this option, ASC generates log event 60701, for which you should set an alarm.

- Configured in: Agent State Connector object, **[default]** section
- Default value: `60`
- Valid values: Integers from `1`–`50000`
- Changes take effect: On restart

stat-srv-ws-conn

> **Important**
> This option has been removed in Agent State Connector (ASC) release 9.0.006.08 and higher. ASC now supports warm standby connections by default.

ASC can now establish a warm standby connection to a primary/backup Stat Server pair.

- Configured in: Agent State Connector object, **[default]** section
- Default value: `true`
- Valid Values:
  - `true`—ASC establishes a warm stand by connection to a primary and backup Stat Server (if a backup Stat Server is configured). On startup, ASC reads the connection parameters for the primary Stat Server from the ASC `Application` object.
  - `false`—ASC establishes a connection only to the primary Stat Server.
- Changes take effect: On restart

timebased-statistic-interval

Specifies the interval (in seconds) between statistic update requests to Stat Server for any statistics you have configured Agent State Connector (ASC) to monitor. See Configure ASC to Monitor Statistics for how to configure ASC and, if necessary, Stat Server. Use this option to ensure that you do not overload Stat Server. Environments with very large agent pools (30,000+ agents) might need to adjust the value of this option.

> **Important**
> In the initial release of this functionality (9.0.012.01), the only supported statistic is StatAgentOccupancy.

- Configured in: Agent State Connector object, **[default]** section
- Default value: `60` (seconds)
- Valid Values: Any positive integer
- Changes take effect: On restart
- Introduced In: 9.0.012.01

threads-max-size

Specifies the maximum number of threads used to subscribe for agent updates from Stat Server using a multithreading approach. Adjust the value for this option as needed in your environment to ensure that you do not overload Stat Server.

> **Important**
>
> - A series of Stat Server switchovers from primary to backup indicates that Stat Server is overloaded.
> - To reduce load on Stat Server *decrease* the value of this option.
> - To resolve this issue, you might also need to adjust the value of the **ss-subscription-timeout** option.

- Configured in: Agent State Connector object, **[default]** section
- Default value: `100`
- Valid Values: Any positive integer up to `2000`
- Changes take effect: On restart

[log] Section

Configure the following options in the **[log]** section:

standard
all
verbose

standard

Specifies the outputs to which an application sends the log events of the Standard level. The log output types must be separated by a comma when more than one output is configured. For example:

```
standard = stderr, network
```

- Configured in: Agent State Connector Application object, **[log]** section
- Mandatory: yes
- Default value: `stdout`
- Valid values: (string)

  - `stdout`—Log events are sent to the Standard output (`stdout`).
  - `stderr`—Log events are sent to the Standard error output (`stderr`).
  - `network`—Log events are sent to Message Server, which can reside anywhere on the network. Message Server stores the log events in the Log Database.
  - `memory`—Log events are sent to the memory output on the local disk. This is the safest output in terms of the application performance.
  - `<filename>`—Log events are stored in a file with the specified name. If a path is not specified, the file is created in the application's working directory.
- Changes take effect: Immediately

all

Specifies the outputs to which an application sends the log events of the `all` level. The log output types must be separated by a comma when more than one output is configured. For example:

```
all = stdout, logfile
```

- Configured in: Agent State Connector Application object, **[log]** section
- Mandatory: yes
- Default value: `stdout`
- Valid values: (string)
  - `stdout`—Log events are sent to the Standard output (`stdout`).
  - `stderr`—Log events are sent to the Standard error output (`stderr`).
  - `network`—Log events are sent to Message Server, which can reside anywhere on the network. Message Server stores the log events in the Log Database.
    Setting the **all** log level option to the `network` output enables an application to send log events of the Standard, Interaction, and Trace levels to Message Server. Debug-level log events are neither sent to Message Server nor stored in the Log Database.
  - `memory`—Log events are sent to the memory output on the local disk. This is the safest output in terms of the application performance.
  - `<filename>`—Log events are stored in a file with the specified name. If a path is not specified, the file is created in the application's working directory.
- Changes take effect: Immediately

verbose

Determines whether a log output is created. If it is, specifies the minimum level of log events generated. The log events levels, starting with the highest priority level, are Standard, Interaction, Trace, and Debug.

- Configured in: Agent State Connector Application object, **[log]** section
- Mandatory: yes
- Default value: `standard`
- Valid values: (string)
  - `all`—All log events (that is, log events of the Standard, Trace, Interaction, and Debug levels) are generated.
  - `debug`—The same as `all`.
  - `trace`—Log events of the Trace level and higher (that is, log events of the Standard, Interaction, and Trace levels) are generated, but log events of the Debug level are not.
  - `interaction`—Log events of the Interaction level and higher (that is, log events of the Standard and Interaction levels) are generated, but log events of the Trace and Debug levels are not.
  - `standard`—Log events of the Standard level are generated, but log events of the Interaction, Trace, and Debug levels are not.
  - `none`—No output is produced.
- Changes take effect: Immediately