Predictive Routing Deployment and Operations Guide

How Does GPR Score Agents?

5/10/2025

Contents

- 1 How Does GPR Score Agents?
  - 1.1 Default Agent Scores
  - 1.2 Score Adjustment
  - 1.3 Threshold Scores
  - 1.4 How the Availability Status of Agents in the Target Agent Group is Determined
  - 1.5 (Optional) Store Scoring Data in the GPR Log File

How Does GPR Score Agents?

GPR scores agents based on the data uploaded to the Agent and Customer Profiles or, if you are using the GPR API, on data passed in the API score request as part of the context parameter. If both are present, data from the API request takes priority over data from the Agent and Customer Profiles.

This topic explains how GPR handles various scoring scenarios, depending on your environment and your configuration settings.

Default Agent Scores

If an agent belongs to the target Agent Group but GPR does not score the agent, the isAgentScoreGood and ScoreIdealAgent subroutines assign a score for that agent according to the value set for the **default-agent-score** configuration option.

For agents who have a default score assigned, the following KVPs reflect that value:

- gpmAgentScore, which records the value specified in the **default-agent-score** option if the agent who handled the interaction had the default score. If the the AICS scoring engine calculated a score for the agent, gpmAgentScore reports the calculated score value.
- gpmDefaultAgentScore, which records the value specified in the **default-agent-score** option.
- gpmDefaultScoreUsed, which indicates whether the selected agent was assigned the default score.
- gpmDefaultScoredAgents, which records the number of agents assigned the default agent score.

Example 1

Agents A and B log in after the scoring request is made and are each assigned the default score, which is 40. Agent A receives the interaction. The related KVPs have the following values:

- gpmAgentScore = 40
- gpmDefaultAgentScore = 40
- gpmDefaultScoreUsed = 1
- gpmDefaultScoredAgents = 2

Example 2

GPR assigns Agent C a score of 80. The default score is 40. No agents are assigned the default score. The related KVPs have the following values:

- gpmAgentScore = 80
- gpmDefaultAgentScore = 40
- gpmDefaultScoreUsed = 0
- gpmDefaultScoredAgents = 0

Score Adjustment

The GPR subroutines enable you to adjust agent scores using an occupancy factor when URS sorts them. You control the agent occupancy setting in the **agent-occupancy-factor** configuration option. Scores adjusted using an agent occupancy factor are recorded in the gpmAdjustedAgentScore KVP.

Example

GPR returns a score of 80 for Agent A. The **agent-occupancy-factor** option value is `0.5`. If this agent selected to receive the interaction, the agent score KVPs have the following values:

- gpmAdjustedAgentScore = 40
- gpmAgentScore = 80

> **Important**
> This adjusted score is used only for sorting the agent scores in the ScoreIdealAgent subroutine. The adjusted agent score is used in the isAgentScoreGood subroutine to compare the agent score with the configured threshold. The actual returned score is used.

Threshold Scores

To implement the agent holdout feature, GPR checks the score returned for the agent against the threshold value configured in the **score-based-threshold** option. URS calls the isAgentScoreGood subroutine to suppress routing to an agent who is in ready state if this agent does not provide an acceptable match for the interaction. This is used in conjunction with relaxation thresholds to target better-matched agents preferentially, expanding the pool of agents if the best-matched agents are unavailable.

- See Agent Holdout Options for the complete list of options used for agent holdout and threshold settings.

Score Relaxation Timeouts

By design, URS checks the threshold relaxation ("awakens") at two-second intervals. As a result, the minimum *real-world* value for **threshold-relaxation-timeout** is 2 because the threshold relaxation is checked only every two seconds. Even though the default value for the **threshold-relaxation-timeout** option is 1, URS applies the threshold relaxation only at two-second intervals.

When the **initial-threshold-timeout** value has elapsed, the minimum score required to allow an agent to handle the interaction is reduced by the configured relaxation step. This relaxation step can be applied multiple times, depending on the option settings you specify.

Example 1

GPR returns a score of 50 for Agent A. The threshold and relaxation options have the following values:

- score-based-threshold = 55
- initial-threshold-timeout = 2
- threshold-relaxation-timeout = 4
- threshold-relaxation-step = 4

Agent A is selected after 6 seconds. The related KVPs have the following values:

- gpmAgentScore =50
- gpmInitialScoreThreshold = 55
- gpmFinalScoreThreshold = 47

URS attempts for Agent A	Threshold Value	Result
Interaction queued and scoring completed in the same second	55 (initial threshold value)	agent score (50) < threshold (55)
after 2 seconds	51 (first relaxation applied, 55-4)	agent score (50) < threshold (51)
after 4 seconds	51 (no change from previous step)	agent score (50) < threshold (51)
after 6 seconds	47 (second relaxation applied, 51-4)	agent score (50) > threshold (47); interaction routed to agent

Example 2

GPR returns a score of 30 for Agent B. The threshold and relaxation options have the following values:

- score-based-threshold = 40
- initial-threshold-timeout = 5
- threshold-relaxation-timeout = 2
- threshold-relaxation-step = 5

Agent A is selected after 8 seconds. The related KVPs have the following values:

- gpmAgentScore =30
- gpmInitialScoreThreshold = 40
- gpmFinalScoreThreshold = 30

URS attempts for Agent B	Threshold Value	Result
Interaction queued and scoring completed in the same second	40 (initial threshold value)	agent score (30) < threshold (40)
after 2 seconds	40 (no change from previous step)	agent score (30) < threshold (40)
after 4 seconds	40 (no change from previous	agent score (30) < threshold (40)

URS attempts for Agent B	Threshold Value	Result
	step)	
after 6 seconds (initial timeout is 5 seconds, but relaxation is applied only when URS awakens)	35 (first relaxation applied, 40-5)	agent score (30) < threshold (35)
after 8 seconds	30 (second relaxation applied, 35-5)	agent score (30) = threshold (30); interaction routed to agent

Example 3

GPR returns a score of 35 for Agent C. The threshold and relaxation options have the following values:

- score-based-threshold = 40
- initial-threshold-timeout = 5
- threshold-relaxation-timeout = 1 (no value specified, default value used)
- threshold-relaxation-step = 1 (no value specified, default value used)

Agent C is selected after 10 seconds. The related KVPs have the following values:

- gpmAgentScore =35
- gpmInitialScoreThreshold = 40
- gpmFinalScoreThreshold = 34

URS attempts for Agent C	Threshold Value	Result
Interaction queued and scoring completed in the same second	40 (initial threshold value)	agent score(35) < threshold (40)
after 2 seconds	40 (no change from previous step)	agent score (35) < threshold (40)
after 4 seconds	40 (no change from previous step)	agent score (35) < threshold (40)
after 6 seconds (initial timeout is 5 seconds, but relaxation is applied only when URS awakens)	38 first and second relaxation applied: • first relaxation after initial 5 seconds • second relaxation after next 1 second	agent score (35) < threshold (38)
after 8 seconds	36 (third and fourth relaxations applied, 38 - 2)	agent score (35) < threshold (36)
after 10 seconds	34 (fifth and sixth relaxations applied, 36 - 2)	agent score (35) > threshold (34); interaction routed to agent

How the Availability Status of Agents in the Target Agent Group is Determined

The source for agent availability information depends on your release of Agent State Connector (ASC).

Agent State Connector release 9.0.015.04 and higher

The connection between ASC and Stat Server is optional. If you do not specify a Stat Server in the ASC `Application` object **Connections** tab, URS provides agent availability information. It checks with Stat Server on the agent login status of the specified target group before making the scoring request, and adds the list of matching agents in the request field 'action_filters'.

The GetActionFilters subroutine reads the login statuses specified to be available for routing from the **login-status-expression** configuration option.

- To use the new functionality, set the value of the **use-action-filters** configuration option, introduced in Ai Core Services release 9.0.015.03, to `false` (the default value is `true`, which maintains the same functionality as in previous releases).

Sample scoring request showing a list of agents employee IDs in the field `action_filters`, where `POC0x` strings indicate IDs of agents with a required login status:

```
{
"token":"<api_token>",
"format_as_map":"true",
"context_id":"3600",
"log_request":"true",
"action_filters":"employeeId in [\"POC01\",\"POC02\",\"POC03\",\"POC04\"]",
"context":
{
    "PR_TYPE":"Gold",
   "PR_LANG":"French"
}
}
```

> **Important**
>
> - This architecture increases the load on URS by approximately 15%. Use the Sizing Guide to verify that you have sufficient URS bandwidth available.
> - Only alphanumeric characters, spaces, and underscores are supported in the names of Stat Server `Application` objects. Names including other special characters cause a malformed scoring request.

Agent State Connector release 9.0.015.01 and lower

All agent status changes (such as login, logout, ready, and so on) are synced to AICS by ASC, which receives the status updates from Stat Server. When a scoring request is made, the target group

names include the login statuses (expression) that mark an agent as available for routing in the action filters field. GPR returns the scores only for the agents present in the target group that matching the specified login status.

Sample scoring request showing the use of action filters with a skill expression and a login status expression, where all agents with the skill 'SkillFrench', having a skill level higher than 10, and who are currently logged in, are selected:

```
{
"token":"<api_token>",
"format_as_map":"true",
"context_id":"3600",
"log_request":"true",
"action_filters":"Skill_French>10 & ((loginStatus>1 & loginStatus<23) | loginStatus>23)",
"context":
{
    "PR_TYPE":"Gold",
   "PR_LANG":"French"
}
}
```

(Optional) Store Scoring Data in the GPR Log File

You can configure the GPR subroutines to log the interaction context and the scoring response details to the GPR API when the **log-to-api** configuration option is set to `true`.

> ### Important
>
> Score log functionality requires the following releases of GPR components:
>
> - AICS 9.0.015.03 and higher
> - URS Strategy Subroutines 9.0.015.00 and higher

With two exceptions, the same values are stored in the score log and in the Genesys Info Mart tables. The exceptions are the following, which appear in the score log but not in the Info Mart database:

- gpmAgentID - Genesys Info Mart requires only the Agent DBID to retrieve detailed agent information during aggregation.
- ConnID - Genesys Info Mart requires only the CallUUID to retrieve detailed interaction information, including the ConnID, during aggregation.

To support real-time reporting in Genesys Pulse, all GPR KVPs are added in two places:

- In user data, which is in the outer body of the score log request.
- In the context field of the score log request, along with other user data typically included in the context field.

Integrate with Genesys Reporting includes a complete list of all KVPs stored in the Genesys Info Mart database.

Configure GPR to Log Scoring Details

To enable score logging, configure the following:

1. Add the following environment variables to the AICS **tango.env** file:
   - LOGS_COLLECTOR_ENABLED = `True`
   - SCORE_LOG_BACKUPCOUNT = `30` (The maximum number of backup copies of the compressed log files.)
   - SCORE_LOG_FILENAME = `/var/log/gpr/supportability-tool-logs/score_logs.log` (The path where the score log files are kept. The log file is mounted to **/datadir/supportability-tool-logs/** on the host running the Tango container.)
2. Restart the Tango container.

Clean Up Scoring Logs

GPR doesn't remove score logs automatically or clean up the score logs associated with a Predictor when that Predictor is deleted. To delete unneeded score logs from MongoDB, AI Core Services release 9.0.015.03 and higher includes the **/scr/gpr/scripts/clean_score_logs.py** script, which is located in the tango container. If you are running a release lower than 9.0.015.03, you can copy the script into the **/scripts** directory and run it successfully.

The commands required to run the **clean_score_logs.py** script depend on your version of AICS:

For release 9.0.013.01 and higher, run the script using the following commands:

```
$ docker exec -ti tango /bin/bash
$ cd /src/gpr/scripts
$ MODE=prod python3.6 clean_score_logs.py <parameters>
```

For release 9.0.012.01 and lower, run the script using the following commands:

```
$ docker exec -ti tango /bin/bash
$ cd /src/gpr/scripts
$ MODE=prod python clean_score_logs.py <parameters>
```

In an HA deployment, execute the script on any node in the Docker Swarm cluster that is running the mongodb service.

Determining the Correct Parameters for the clean_score_logs.py Script

The parameters to be passed with the script to clean up unneeded score logs depend on how the score log is structured.

- When you pass an API POST /score_log request that includes a Predictor ID, GPR stores the score log in a collection named using the following format:

`scorelog_<predictor_id><predictor_name_without_spaces-or_tabs>`.
To delete such a collection, run the script *without* parameters. The script automatically detects collections that are left after the corresponding predictors were deleted and deletes them.

- When you pass an API POST /score_log request that does *not* include a Predictor ID, GPR stores the score log in a *default* account-specific collection named `scorelog_default_nopredictor_<account_id>`. See Deleting From Default Score Log Collections (below) for specific instructions.
- The optional `--dry_run` parameter instructs the script not to clean up the score log files, but only to display how many collections/records are going to be deleted.

Deleting From Default Score Log Collections

The exact format of the default score logs depends on the Universal Routing Server (URS) strategy configuration. To remove score logs for deleted Predictors from the default collection, define the path to the Predictor ID in the score log body structure and pass it to the script as a separate parameter. If you do not specify the path, the script cannot automatically detect which Predictor data to delete.

The following examples show different score log structures in the default collection.

- Note that the examples assume you are using AICS 9.0.013.01 or higher. If you are running an earlier version, adapt the commands as explained at the beginning of the Clean Up Score Logs section (above).

1. To delete logs formatted as in Example 1 and Example 2 (below) run the script with the parameter `--predictor_key=p_id`, as shown in the following command:

   ```
   python3.6 src/gpr/scripts/clean_score_logs.py --predictor_key=p_id
   ```

2. To delete logs formatted as in Example 3 (below) run the script with the parameter `--predictor_key=context.gpmPredictorId`, as shown in the following command:

   ```
   python3.6 clean_score_logs.py --predictor_key=context.gpmPredictorId
   ```

Example 1

```
{
    "_id" : ObjectId("5d0c94e341c24c30bdc970ff"),
    "param" : "1",
    "p_id": "5ae0cb37e2d22d221dc07c5b",
    "created_at_slog" : ISODate("2019-06-21T08:27:15.722Z")
},
```

Example 2

```
{
    "_id" : ObjectId("5d0c94e341c24c30bdc97101"),
    "param" : "2",
    "p_id": "5ae0cb37e2d22d221dc07c5b",
    "created_at_slog" : ISODate("2019-06-21T08:33:12.107Z")
},
```

Example 3

```
{
    "context_id" : "3600",
    "prrPredictor" : "",
    "context" : {
```

```
        "gpmMedianScore" : "9",
        "last_name" : "A",
        "gpmPredictorId" : "5ad6802b8615b3002c8985d0",
        "gpmAgentScore" : "0",
        "sex" : "M",
        "seniority" : "NEW",
    },
    "created_at_slog" : ISODate("2019-05-17T08:20:11.503Z
}
```

Sample Score Log Output

The following sample shows what you might receive in response to a request for score log details from the API. For details, see the *Predictive Routing API Reference*. (Requires a password for access. Please contact your Genesys representative if you need to view this document.)

```
{
"prrPredictor": "manual_test",
 "gpmMode": "prod",
 "gpmStatus": "agent-surplus",
 "gpmPredictor": "manual_test",
 "gpmPredictorId": "5cb07deeb3fc5800397fb8f4",
 "gpmModel": "manual_model",
 "gpmModelId": "5cb07e11c0eb2f00353b6151",
 "gpmResult": "1",
 "gpmGlobalScore": "0",
 "gpmMedianScore": "21",
 "gpmMaxScore": "72",
 "gpmMinScore": "9",
 "gpmTargetSize": "4",
 "gpmUse": "1",
 "gpmCustomerFound": "1",
 "CALLID": "01VI5E6TOSED503B16OAHG5AES000004",
 "START_TS": "1559063048",
 "gpmRouteAttemptId": "1",
 "context_id": "3600",
 "gpmAgentScore": "72",
 "gpmAgentRank": "1",
 "gpmScoreAboveMedian": "1",
 "gpmPredictorType": "Service",
 "gpmDefaultAgentScore": "21",
 "gpmDefaultScoredAgents":"4",
 "gpmDefaultScoreUsed": "0",
 "gpmInitialScoreThreshold": "50",
"gpmFinalScoreThreshold": "50",
 "gpmAdjustedAgentScore": "72",
 "gpmSuitableAgentsCount": "2",
 "gpmGlobalScoreCount": "0",
 "gpmAgentID" : "POC01",
 "ConnID": "006c02dcefa47049"
 "gpmAgentDBID": "104",
 "gpmWaitTime": "6",
 "context":
 {
       "PR_TYPE": "Gold",
       "PR_LANG": "French",
       "RPVQID": "0016FD27SOEDB5861EOAHG5AES000004"
 }
}
```