GVP Deployment Guide

How the Policy Server Works

5/1/2025

How the Policy Server Works

Policy Server exposes a read-only HTTP/HTTPS interface that can be used by other components to perform GVP policy queries for the entire deployment.

HTTP Response Format

The HTTP interface returns data (responds to queries) in JSON format with the content-type `application/json` and supports JSONP when a query parameter named callback is supplied. The JSON callback response is returned with the content-type `text/javascript`.

If the target service does not recognize the query parameter, the query is silently ignored. A request within a partial URL that does not directly match an existing service receives a 404 response.

Further information about how Policy Server performs its role in a GVP deployment is provided in the topics:

- DID Management
- Policy Management
- Service Description
- High Availability

DID Management

Policy Server recognizes Direct Inward Dialing (DID) number range specifiers and named DID groups.

DID Range Specifiers

Policy Server recognizes two or more DID numbers expressed as DID range specifiers in one of the following three forms:

1. A single DID, for example, 300.
2. A range of DIDs, for example, 300-400 means all numbers between 300 and 400, inclusive. The lower number must be first in the range and the higher number second. For example, 400-300 is an invalid specifier.
3. A DID prefix, for example, 45* means all numbers that start with the digits 45 (for example: 45, 450, 4500, 4501, 4599 and up to the maximum allowable that match the prefix).

A string that does not match any of these three forms is considered an invalid specifier, and two DID range specifiers are considered overlapping when at least one DID in each span is the same.

Named DID Groups

Named DID groups can contain zero or more DID range specifiers and a tenant can have zero or more DID group assignments. Policy Server maintains DID groups, DID range specifiers for the entire deployment in its in-memory store. Updates to DID Groups or their DID range-specifiers are immediately reflected in-memory.

DID Overlap Queries

Genesys Administrator can query Policy Server for any overlaps in the DID range specifiers in the deployment by using an HTTP GET request in the following format:

`GET /dids/overlaps/?spec=<specifier>[&spec=<specifier>]`
...where <specifier> is a DID range specifier, for example: `GET /dids/overlaps/?spec=300-400&spec=550&spec=551`

A request can have multiple spec parameters in a GET request to query overlaps for multiple DID range specifiers or it can have none. If there are no spec parameters in the request, no overlap results are returned.

The maximum number of spec parameters that can be simultaneously specified depends on the client s URL length limits and the server implementation. If there are any invalid specifiers in the request, a 400 response is returned. If no overlaps are found or no spec parameters are specified, a 200 response with an empty array is returned, for example:

`No overlap response [ ]`

If overlaps are found, a 200 response is returned with an array of overlap details. Each array item is an object with the following properties:

- **specifier** the DID range specifier in the request.
- **overlaps** An array of objects that provide details about the overlap. The objects have the following properties:
  - **tenant** The tenant with the id property (the DBID of the tenant).
  - **group** The DID group with the name property.
  - **specifier** The DID range specifier that contains the overlap.

See the following example of an overlap response:

```
[
 {
   "specifier": "55*",
   "overlaps": [
     {
       "tenant": { "id": 101 },
       "group": { "name": "Group1" },
       "specifier": "500-600"
     },
     {
       "tenant": { "id": 101 },
       "group": { "name": "Group1" },
       "specifier": "5567"
```

```
      }
     ]
    },
      {
        "specifier": "6700",
        "overlaps": [
          {
            "tenant": { "id": 101 },
            "group": { "name": "Group1" },
            "specifier": "6000-8500"
      }
    ]
  }
]
```

Configuration of Maximum Overlaps

You can use the did.max_overlaps option to configure the maximum number of overlaps that can be returned. The default value is 10.

Policy Management

The Resource Manager makes call processing and resource allocation decisions, based on the policies that are defined within the tenant hierarchy and IVR profiles. Some policies are inherited and thereby, enforced by the parent tenant.

The moment the Resource Manages serves up a resource (for example, call or speech processing), the policies that are in effect are resolved by the Resource Manager on-the-fly, based on the current resource utilization.

Static Analysis of Policies

Policy Server performs static analysis on some policies to validate and determine the inherited values or enforcements that are currently in effect for a tenant or IVR profile. This information can be used by other components to determine the policy values that are currently in effect.

Tenant Policy Queries

You can query the Resource Manager tenant policies by using an HTTP GET request in the following format:

`GET /tenants/<tenant id>/policies/[<policy>]` where:

- `<tenant id>` is the DBID of the tenant
- `<policy>` is the name of the policy (optional), for example:

```
GET /tenants/101/policies
```

```
GET /tenants/101/policies/max-ports
```

> **Tip**
> The <policy> parameter is optional. If it is omitted, Policy Server returns all known policies (not just the currently defined tenant policies).

Response Rules For Tenants

If a specified tenant does not exist, a 404 response is returned. If a specified policy does not match a known policy, Policy Server performs the following steps in this order:

1. Checks for parent enforcement of this policy. If yes, the enforcement value is used as the effective value.
2. Checks for an existing value for this policy. If yes, this value is used as the effective value.
3. Checks the query to see if there is a new assigned value for this policy. If yes, use the new value as the effective value.
4. If there is no parent enforcement, existing value, or newly assigned value, there is no effective value for this policy.

When an existing tenant and a known policy are specified (or no policy is specified), a 200 response is returned with an array of policy details. Each array item is an object with the following properties:

- **name** The name of the policy.
- **value** The value of the policy that is defined for the queried tenant.
- **enforcement** The value that is enforced by the immediate parent of the queried tenant.
- **effective** The effective value that is resolved, based on the inheritance and enforcement rules for this policy.

See the following example of an effective policy response:

```
[
 {
   "name": "max-ports",
   "value": 300,
   "enforcement": 250,
   "effective": 250
 },
 {
   "name": "conference-enabled",
   "value": false,
   "effective": false
 }
]
```

IVR Profile Policy Queries

You can query the Resource Manager IVR Profile policies by using an HTTP GET request in the following format:

`GET /tenants/<tenant id>/ivrprofiles/<profile id>/policies/[<policy>]` where:

- `<tenant id>` Is the DBID of the tenant.
- `<profile id>` Is the DBID of the IVR profile.
- `<policy>` Is the name of the policy (optional), for example:
  `GET /tenants/101/ivrprofiles/42/policies`
  `GET /tenants/101/ivrprofiles/42/policies/max-ports`

> **Tip**
> The <policy> parameter is optional. If it is omitted, Policy Server returns all known policies (not just the currently defined IVR Profile policies).

Response Rules For IVR Profiles

The response rules are the same for the for IVR Profile policy queries as they are for the tenant policy queries. See Response Rules For Tenants.

Service Level Policies

Managed Service Providers in general and Enterprise departments in particular may use GVP to provide self-service to many different, independent entities, using the same shared GVP system of resources. Some entities may also do outbound calling in bulk mode at various times of the day, and this load could impact inbound self-service call handling, due to the potentially excessive load.

To manage these different interests and loads, and also to accomplish billing solutions for MSPs, GVP offers limit settings that are based on the number of simultaneous calls allowed for a given application, which is effectively equivalent to the number of ports.

Level 1 and 2 thresholds are non-blocking, and when exceeded by traffic demands, result in this fact being recorded in all CDR records for that application during the excess traffic. Level 3 can be set to blocking, which means that if the number of simultaneous calls equals the level 3 setting, then any subsequent calls arriving for that application will be rejected until traffic volumes decline below the threshold.

By writing the levels into the CDR record, the potential exists to bill the application owner (or tenant owning the application) for different levels of call volumes without blocking calls.

Procedure: Setting Service Levels

Use Genesys Administrator (GA) to set the parameters that specify these service levels, for both tenant and profile.

1. Set the parameter `[gvp.policy] burst-allowed` (which determines whether bursting is allowed or not) to `true`.
2. Enter the number of ports assigned to the customer in the parameter `[gvp.policy]usage-limits`.

These are level 1 ports the initial ports that a customer purchases for which they will be billed at normal rates. Rule 1 refers to the default usage policy value that is present when a GVP installation is done for a customer and specific number of ports are allotted. If its value=0 (not empty), all calls fail. You can set Rule 1 at the Tenant level or at the specific IVR application level.

3. Enter the number of ports assigned to the customer in the parameter `[gvp.policy]level2-burst-limit`.
   These are level 2 ports used when all of the Level 1 Ports are exhausted. For these ports, the customer is billed at a different rate. `Level 2 Ports = Level 1 Ports + Level 2 bursting` For example, if the customer has 1000 ports at level 1 and they want 100 more ports in the next level, set level 2 ports at 1100. Therefore, if the customer IVR profile receives 1100 simultaneous calls, 1000 ports are billed at normal rates, and 100 ports are billed at the level 2 rate.
4. Enter the number of ports assigned to the customer in the parameter `[gvp.policy]level3-burst-limit`.
   These are level 3 ports the next level of bursting, billed at a higher rate than level 2. For example, if the customer wants 100 ports in level 3, then set the level 3 Ports field to `1200 (1000 + 100 + 100)`.
5. Enter the number of billable ports assigned to the customer in the parameter `[gvp.policy]`.
   Billable ports are used by some customers where the number of ports to be billed is different from the number of ports provisioned (port levels 1, 2, and 3). This information is transferred to Reporting Server and stored as part of the CDR records. The parameter `[gvp.policy]` is available only if Reporting Server is installed.

Policy Resolution For Uncommitted Values

Policy Server can stage a policy resolution of uncommitted values by providing the value and enforcement query parameters when effective policies are queried for either a tenant or an IVR profile. This can be useful when you want to know what the effective policies will be before you save any changes that were made to the policies.

Policy Server supports a limited set of policy resolution types, which is a subset of the types that the Resource Manager can achieve during real-time call processing.

- Policy Server supports two top-down policy resolution types limit and feature-allowed. For example, if the tenant s usage-limits option has a value of 23 and its parent tenant usage-limits option has a value of 8, then the tenant's effective value for usage-limits will be 8.
- When there is no defined resolution type, Policy Server supports bottom-up (or pass-through) resolution of policy values in the following ways:
  - If the tenant's policy value is available, it is used as the effective value.
  - If the tenant's policy value is not available, the first available value of a parent tenant's policy is used from the bottom-up.
- Policy Server supports one-level enforcement policy resolution. The tenant's policy value is enforced by its immediate parent tenant. When the policy value is enforced, the enforced value is used, regardless of any other policy values. Enforcement policy resolution is applied for tenants only, (not for IVR Profile policies).

Service Description

Policy Server returns a Web Application Description Language document that describes the services it provides whenever a user visits the root URL ("/"). The WADL document is styled with an embedded Extensible Stylesheet Language (XSL) reference to render an HTML page that acts as documentation for the services. Policy Server serves the static files that are required for this transformation (XSL, Cascading Stylesheet (CSS), and image files) from the /static URL.

The WADL document includes the following information:

- The name of the product (VP Policy Server).
- The name of the Management Framework Application object that is represented by this instance.
- The product version.

High Availability

You can deploy Policy Server in warm standby mode for High Availability by using the Backup Server option in the Server Info section of the Policy Server Application. The active standby status is determined by the Solution Control Server (SCS), and because the Policy Server is stateless, data does not require synchronization.

Data Storage

Policy Server uses in-memory to store the data that it obtains from Management Framework. It does not require external databases or files for data storage.