FORGEROCK®

Monitoring Guide
/ ForgeRock Identity Management 7

Latest update: 7.0.4

Copyright © 2011-2020 ForgeRock AS.

Abstract

Guide to configuring ForgeRock® Identity Management server logs and monitoring metrics.

Table of Contents

Overview

This guide covers server logging, monitoring, and metrics.

Quick Start

Server Logs	Monitoring	Metrics
Manage and read server logs.	Set up systems to monitor IDM.	Find reference information for monitoring metrics.

ForgeRock Identity Platform™ serves as the basis for our simple and comprehensive Identity and Access Management solution. We help our customers deepen their relationships with their customers, and improve the productivity and connectivity of their employees and partners. For more information about ForgeRock and about the platform, see https://www.forgerock.com.

The ForgeRock Common REST API works across the platform to provide common ways to access web resources and collections of resources.

Chapter 1
Configure Server Logs

Server logging is not the same as auditing. Auditing logs activity on the IDM system, such as access, and synchronization. Server logging records information about the internal workings of IDM, like system messages, error reporting, service loading, or startup and shutdown messaging.

Server logging is configured in your project's `conf/logging.properties` file. Changes to logging settings require a server restart before they take effect. Alternatively, use JMX via jconsole to change the logging settings. In this case, changes take effect without restarting the server.

+ *Specify Where Messages Are Logged*

The way in which messages are logged is set in the `handlers` property in the `logging.properties` file. This property has the following value by default:

```
handlers=java.util.logging.FileHandler, java.util.logging.ConsoleHandler
```

Two handlers are listed by default:

- `FileHandler` writes formatted log records to a single file or to a set of rotating log files. By default, log files are written to `logs/openidm*.log` files.

- `ConsoleHandler` writes formatted logs to `System.err`.

Additional log message handlers are listed in the `logging.properties` file.

+ *Set the Log Message Format*

IDM supports the two default log formatters included with Java. These are set in the `conf/logging.properties` file:

- `java.util.logging.SimpleFormatter.format` outputs a text log file that is human-readable. This formatter is used by default.

- `java.util.logging.XMLFormatter` outputs logs as XML, for use in logging software that can read XML logs.

IDM extends the Java `SimpleFormatter` with the following formatting options:

- `org.forgerock.openidm.logger.SanitizedThreadIdLogFormatter`

This is the default formatter for console and file logging. It extends the `SimpleFormatter` to include the thread ID of the thread that generated each message. The thread ID helps with debugging when reviewing the logs.

In the following example log excerpt, the thread ID is `[19]`:

```
[19] May 23, 2018 10:30:26.959 AM org.forgerock.openidm.repo.opendj.impl.Activator start
INFO: Registered bootstrap repository service
[19] May 23, 2018 10:30:26.960 AM org.forgerock.openidm.repo.opendj.impl.Activator start
INFO: DS bundle started
```

The `SanitizedThreadIdLogFormatter` also encodes all control characters (such as newline characters) using URL-encoding, to protect against log forgery. Control characters in stack traces are not encoded.

- `org.forgerock.openidm.logger.ThreadIdLogFormatter`

Similar to the `SanitizedThreadIdLogFormatter`, but does not encode control characters. If you do not want to encode control characters in file and console log messages, change the file and console handlers in `conf/logging.properties` as follows:

```
java.util.logging.FileHandler.formatter = org.forgerock.openidm.logger.ThreadIdLogFormatter
```

```
java.util.logging.ConsoleHandler.formatter = org.forgerock.openidm.logger.ThreadIdLogFormatter
```

The `SimpleFormatter` (and, by extension, the `SanitizedThreadIdLogFormatter` and `ThreadIdLogFormatter`) lets you customize what information to include in log messages, and how this information is laid out. By default, log messages include the date, time (down to the millisecond), log level, source of the message, and the message sent (including exceptions). To change the defaults, adjust the value of `java.util.logging.SimpleFormatter.format` in your `conf/logging.properties` file. For more information on how to customize the log message format, see the related Java documentation.

+ *Set the Logging Level*

By default, IDM logs messages at the `INFO` level. This logging level is specified with the following global property in `conf/logging.properties`:

```
.level=INFO
```

You can specify different separate logging levels for individual server features which override the global logging level. Set the log level, per package to one of the following:

```
SEVERE (highest value)
WARNING
INFO
CONFIG
FINE
FINER
FINEST (lowest value)
```

For example, the following setting decreases the messages logged by the embedded PostgreSQL database:

```
# reduce the logging of embedded postgres since it is very verbose
ru.yandex.qatools.embed.postgresql.level = SEVERE
```

Set the log level to `OFF` to disable logging completely (see in Disable Logs), or to `ALL` to capture all possible log messages.

If you use `logger` functions in your JavaScript scripts, set the log level for the scripts as follows:

```
org.forgerock.openidm.script.javascript.JavaScript.level=level
```

You can override the log level settings, per script, with the following setting:

```
org.forgerock.openidm.script.javascript.JavaScript.script-name.level
```

For more information about using `logger` functions in scripts, see "Log Functions" in the *Scripting Guide*.

> **Important**
>
> It is strongly recommended that you do *not* log messages at the `FINE` or `FINEST` levels in a production environment. Although these levels are useful for debugging issues in a test environment, they can result in accidental exposure of sensitive data. For example, a password change patch request can expose the updated password in the Jetty logs.

+ *Configure Log File Rotation*

Log files are rotated when the size reaches 5 MB, and IDM retains up to 5 files. All system and custom log messages are also written to these files. You can modify these limits in the following properties in the `logging.properties` file for your project:

```
# Limiting size of output file in bytes:
java.util.logging.FileHandler.limit = 5242880

# Number of output files to cycle through, by appending an
# integer to the base file name:
java.util.logging.FileHandler.count = 5
```

> **Note**
>
> There is currently no `logging.properties` setting for time-based rotation of server log files. However, on UNIX systems you can use the **logrotate** command to schedule server log rotation at a regular interval. For more information, see the logrotate man page.

+ *Disable Logs*

If required, you can also disable logs. For example, to disable `ConsoleHandler` logging, make the following changes in your project's `conf/logging.properties` file before you start IDM.

Set `java.util.logging.ConsoleHandler.level = OFF`, and comment out other references to `ConsoleHandler`, as shown in the following excerpt:

```
# ConsoleHandler: A simple handler for writing formatted records to System.err
#handlers=java.util.logging.FileHandler, java.util.logging.ConsoleHandler
handlers=java.util.logging.FileHandler
...
# --- ConsoleHandler ---
# Default: java.util.logging.ConsoleHandler.level = INFO
java.util.logging.ConsoleHandler.level = OFF
#java.util.logging.ConsoleHandler.formatter = ...
#java.util.logging.ConsoleHandler.filter=...
```

Chapter 2
Set up Monitoring

IDM includes the following tools for monitoring metrics:

- A Dropwizard dashboard widget, for viewing metrics within IDM.

- A Prometheus endpoint, for viewing metrics through external resources such as Prometheus and Grafana.

The tool that you choose will depend on the metrics you monitor and on what you plan to use the metrics for.

IDM does not gather metrics by default. To enable metrics gathering, open `conf/metrics.json` and set the `enabled` property to `true`.

To verify that metrics are successfully enabled, run:

```
curl \
--header "X-OpenIDM-Username: openidm-admin" \
--header "X-OpenIDM-Password: openidm-admin" \
--header "Accept-API-Version: resource=1.0" \
--request GET \
'http://localhost:8080/openidm/metrics/api?_queryFilter=true'
```

If you have enabled metrics, this command returns information about the metrics that are being collected. Otherwise, the command returns a bad request response.

Metrics are reported only after they have been triggered by activity in IDM, such as a reconciliation.

For a list of available metrics, see "*Metrics Reference*".

Monitoring With the Dropwizard Widget

The Dropwizard widget creates a graph of metrics based on server activity. Locate the Dropwizard widget as follows:

1. Navigate to a dashboard in the Admin UI.

2. Select Add Widget, then select Dropwizard Table with Graph from the list of available widgets.

The Dropwizard widget is useful for lightweight, live monitoring of IDM, however, it has a few limitations:

- The graphs created by the widget do not persist; they are restarted if you reload or navigate away from the page.

- The widget only works with time-based metrics (other forms of metrics are not displayed in the widget).

Monitoring With the Prometheus Endpoint

Prometheus is a third-party tool used for gathering and processing monitoring data. Prometheus uses the `openidm/metrics/prometheus` endpoint to gather information. This endpoint is protected by a basic authentication filter, using the following credentials, set in the `resolver/boot.properties` file:

```
openidm.prometheus.username=username
openidm.prometheus.password=password
```

This section describes how to install Prometheus and Grafana to collect IDM metrics. These third-party tools are *not* supported by ForgeRock. For more information running Prometheus, see the Prometheus documentation.

+ *Configure Prometheus*

You must configure Prometheus using a `prometheus.yml` configuration file. For more information, see the Prometheus configuration documentation. An example `prometheus.yml` file would be:

```
global:
  scrape_interval: 15s
  external_labels:
    monitor: 'my_prometheus'

# https://prometheus.io/docs/operating/configuration/#scrape_config
scrape_configs:
  - job_name: 'openidm'
    scrape_interval: 15s
    scrape_timeout: 5s
    metrics_path: 'openidm/metrics/prometheus'
    scheme: http
    basic_auth:
      username: 'prometheus'
      password: 'prometheus'
    static_configs:
      - targets: ['localhost:8080']
```

This example configures Prometheus to poll the IDM endpoint every 5 seconds (`scrape_interval: 5s`), receiving metrics in a plain text format (`_fields: ['text']` and `_mimeType: ['text/plain;version=0.0.4']`). For more information about reporting formats, see the Prometheus documentation on Exposition Formats. Use **curl** to validate that this configuration returns the expected results:

```
curl \
--user prometheus:prometheus \
--header "Accept-API-Version: resource=1.0" \
--request GET \
'http://localhost:8080/openidm/metrics/prometheus'
```

Start Prometheus with your `prometheus.yml` configuration file by running:

```
prometheus --config.file=/path/to/prometheus.yml
```

You can confirm that Prometheus is gathering data from IDM by navigating to the Prometheus monitoring page (by default, `http://localhost:9090`).

+ *Configure Grafana*

Prometheus lets you monitor and process information provided by IDM. If you need deeper analytics, you can use tools such as Grafana to create customized charts and graphs based on Prometheus data. For information on installing and running Grafana, see the Grafana website.

> **Tip**
>
> The default username and password for Grafana is `admin` and `admin`.

To set up a Grafana dashboard with IDM metrics using Prometheus, add your Prometheus installation to Grafana, as a data source. Select Configuration > Data Sources from the left navigation panel in Grafana, then select Add Data Source.

Complete the options on the Add Data Source screen. The options in this list match the settings in the `monitoring.dashboard.json` file.

1. Give your data source a name, in this case, `ForgeRockIDM`.

2. Select Prometheus as the type.

3. Set the URL (by default, `http://localhost:9090`).

4. Set Access to proxy.

5. Enable Basic Auth.

6. Set a username and password of `prometheus` and `prometheus`.

7. Select Save & then Test Connection. If the configuration succeeds, you'll see the following message: "Data source is working."

When Prometheus has been set up as a data source in Grafana, you can create a dashboard with IDM metrics. Create a dashboard in one of the following ways:

• Download the Monitoring Dashboard Samples from the ForgeRock BackStage download site. Find `monitoring.dashboard.json` in the downloaded .zip file. In the Grafana administrative screen, select Home > Import and import the noted file.

• • Select Create > Dashboard.

  • Select Graph.

- Select Panel Title > Edit.

    Enter the metrics you want to display (which will be available in autocomplete as you type), or build more complex queries using the Prometheus query language.

To display a Grafana chart directly in IDM, create a new dashboard in the Admin UI. Configure the dashboard to be an Embedded URL and set the URL to your Grafana installation (`http://localhost:3000` by default).

Chapter 3
Metrics Reference

IDM exposes a number of metrics. Metrics are categorized as follows:

Timers

Timers provide a histogram of the duration of an event, along with a measure of the rate of occurrences. Timers can be monitored using the Dropwizard dashboard widget and the IDM Prometheus endpoint. Durations in timers are measured in milliseconds. Rates are reported in number of calls per second. The following example shows a Timer metric:

```
{
  "_id": "sync.source.perform-action",
  "count": 2,
  "max": 371.53391,
  "mean": 370.1752705,
  "min": 368.816631,
  "p50": 371.53391,
  "p75": 371.53391,
  "p95": 371.53391,
  "p98": 371.53391,
  "p99": 371.53391,
  "p999": 371.53391,
  "stddev": 1.3586395,
  "m15_rate": 0.393388581528647,
  "m1_rate": 0.311520313228562,
  "m5_rate": 0.3804917698002856,
  "mean_rate": 0.08572717156016606,
  "duration_units": "milliseconds",
  "rate_units": "calls/second",
  "total": 740.350541,
  "_type": "timer"
}
```

Summaries

Summaries are similar to Timers in that they measure a distribution of events. However, Summaries record values that aren't units of time, such as user login counts. Summaries cannot be graphed in the Dropwizard dashboard widget, but are available through the Prometheus endpoint, and by querying the `openidm/metrics/api` endpoint directly. The following example shows a Summary metric:

```
{
  "_id": "audit.recon",
  "m15_rate": 0.786777163057294,
  "m1_rate": 0.623040626457124,
  "m5_rate": 0.7609835396005712,
  "mean_rate": 0.16977218861919927,
  "units": "events/second",
  "total": 4,
  "count": 4,
  "_type": "summary"
}
```

Also see "*Set up Monitoring*".

All metrics are available at both the `openidm/metrics/api` and `openidm/metrics/prometheus` endpoints. The actual metric names can vary, depending on the endpoint used.

API Metrics

Metrics accessed at the `api` endpoint (such as those consumed by the Dropwizard dashboard widget) use dot notation for their metric names; for example, `recon.target-phase`. The following table lists the API metrics available in IDM:

API Metrics Available in IDM

API Metric Name	Type	Description
audit.*audit-topic*	Summary	Count of all audit events generated of a given topic type.
field.augmentation.edge	Timer	Rate of reading response objects, to fulfill the _fields requested (when the fields were not populated by the initial repo query).
field.augmentation.vertex	Timer	Rate of reading response objects, to fulfill the _fields requested (when the fields were not populated by the initial repo query).
filter.*filter-type.action.script-name*	Timer	Rate at which filter scripts are executed, per action. Monitors scripted filters and delegated admin.
icf.*system-identifier.objectClass*.query._queryExpression	Timer	Rate of ICF query executions with queryExpression, and time taken to perform this operation.
icf.*system-identifier.objectClass*.query._queryFilter	Timer	Rate of ICF query executions with queryFilter, and time taken to perform this operation.

API Metric Name	Type	Description
icf.*system-identifier.objectClass*.query._queryId.*queryId*	Timer	Rate of ICF query executions with queryId, and time taken to perform this operation.
icf.*system-identifier.objectClass*.query._UNKNOWN	Timer	Rate of ICF query executions when the query type is UNKNOWN, and time taken to perform this operation.
internal.*managed-object.operation*	Timer	Rate of operations on internal objects.
internal.*managed-object*.relationship.fetch-relationship-fields	Timer	Rate of fetch operations of relationship fields for internal objects.
internal.*managed-object*.relationship.get-relationship-value-for-resource	Timer	Query rate on relationship values for internal objects.
internal.*managed-object*.script.*script-name*	Timer	Rate of script executions on internal object.
internal.*managed-object*.relationship.validate-relationship-fields	Timer	Rate of validate operations of relationship fields for internal objects.
managed.field.augmentation	Timer	Rate of responses requiring field augmentation. When the repository cannot retrieve all data in a single call, IDM performs additional read operations to complete (augment) the missing data.
managed.*managed-object.operation*	Timer	Rate of operations on a managed object.
managed.*managed-object*.relationship.fetch-relationship-fields	Timer	Rate of fetches of relationship fields of a managed object.
managed.*managed-object*.relationship.get-relationship-value-for-resource	Timer	Rate of queries to get relationship values for a resource on a managed object.
managed.*managed-object*.relationship.validate-relationship-fields	Timer	Rate of validations of relationship fields of a managed object.
managed.*managed-object*.script.*script-name*	Timer	Rate of executions of a script on a managed object.
managed.object.handle-temporal-constraints-on-create	Timer	Latency of enforcing temporal constraints on role objects during object creation.
managed.object.handle-temporal-constraints-on-delete	Timer	Latency of enforcing temporal constraints on role objects during object deletion.
managed.object.handle-temporal-constraints-on-update	Timer	Latency of enforcing temporal constraints on role objects during object update.
managed.relationship.handle-temporal-constraints-on-create	Timer	Latency of enforcing temporal constraints on relationship grants during edge creation.
managed.relationship.handle-temporal-constraints-on-delete	Timer	Latency of enforcing temporal constraints on relationship grants during edge deletion.

API Metric Name	Type	Description
`managed.relationship.handle-temporal-constraints-on-update`	Timer	Latency of enforcing temporal constraints on relationship grants during edge update.
`managed.relationship.validate.read-relationship-endpoint-edges`	Timer	Rate of reads on relationship endpoint edges for validation.
`null_array_filter.augmentation`*requestType*	Timer	Time spent in filter which maps non-nullable, null-valued array fields to an empty array. This filter is traversed for all repo access relating to internal and managed objects.
`recon`	Timer	Rate of executions of a full reconciliation, and time taken to perform this operation.
`recon-assoc-entry.merged-query.merge-results`	Timer	Rate of merge operations after source and/or target objects have been retrieved during a merged query of recon association entries.
`recon-assoc-entry.merged-query.page-assoc-entries`	Timer	Rate of individual paged recon association entry queries during a merged query. More than one page of entries might be requested to build a single page of merged results.
`recon-assoc-entry.merged-query.query-source`	Timer	Rate of source object retrieval via query when merging source objects to recon association entries.
`recon-assoc-entry.merged-query.query-target`	Timer	Rate of target object retrieval via query when merging target objects to recon association entries.
`recon.association-persistence.`*recon-id-operation*	Timer	The time taken to persist association data. The *operation* can be `source`, `target`, or `amendsource`, depending on whether data is being produced for a source-phase or target-phase recon association, or to amend the association for a specific source.
`recon.id-queries-phase`	Timer	Rate of executions of the id query phase of a reconciliation, and time taken to perform this operation.
`recon.source-phase`	Timer	Rate of executions of the source phase of a reconciliation, and time taken to perform this operation.
`recon.source-phase.page`	Timer	Rate of pagination executions of the source phase of a reconciliation, and time taken to perform this operation.

API Metric Name	Type	Description
`recon.target-phase`	Timer	Rate of executions of the target phase of a reconciliation, and time taken to perform this operation.
`repo.jdbc.relationship.edge.execute.joinedToVertex`	Timer	Time (ms) spent running the Edge->Vertex relationship join query on the database and collecting the result set.
`repo.jdbc.relationship.execute`	Timer	Rate of relationship graph query execution times.
`repo.jdbc.relationship.process`	Timer	Rate of relationship graph query result processing times.
`repo.raw._queryId.queryId`	Timer	Rate of executions of a query with queryId at a repository level, and time taken to perform this operation.
`repo.repo-type.cache.objecttypes.event.resource-mapping`	Count	Counts the usage statistics of the `objecttypeid` cache, which maps an object type to its `objecttypeid`. The expected count is a small number of misses (sometimes, only one) and the remainder of hits.
`repo.repo-typeget-connection`	Timer	Rate of retrievals of a repository connection.
`repo.repo-type.operation.action_name.command.resource-mapping`	Timer	Rate of actions to a repository datasource for a generic/explicit mapped table.
`repo.repo-type.operation._adhoc-expression.relationship`	Timer	Rate of filtered queries (using native query expressions) on the relationship table. This metric measures the time spent making the query (in ms), and the number of times the query is invoked.
`repo.repo-type.operation._adhoc-filter.relationship`	Timer	Rate of filtered queries (using the `_queryFilter` parameter) on the relationship table. This metric measures the time spent making the query (in ms), and the number of times the query is invoked.
`repo.repo-type.create_properties.execute.resource-mapping`	Timer	Rate of execution time on the JDBC database for the `create_properties` operations. This operation is performed for every generic object `create` when it persists the searchable properties. The rate measured here does not include the time taken to obtain a connection to the database from the connection pool. The physical connections to the database have already been established inside the connection pool.

API Metric Name	Type	Description
repo.*repo-type.operation.* execute.*resource-mapping*	Timer	Rate of execution time on the JDBC database for CRUD operations. This rate does not include the time taken to obtain a connection to the database from the connection pool. The physical connections to the database have already been established inside the connection pool.
repo.*repo-type*.query.execute.*resource-mapping*[.*queryType.*]	Timer	Rate of execution time on the JDBC database for queries (either `queryFilter` or `queryId`). This rate does not include the time taken to obtain a connection to the database from the connection pool. The physical connections to the database have already been established inside the connection pool.
repo.*repo-type.operation*.relationship	Timer	Rate of CRUDPAQ operations to a repository datasource for a generic/explicit/relationship mapped table.
repo.*repo-type.operation.* relationship.*stage.origin_type*	Timer	Time (ms) spent in the various phases to retrieve relationship expanded data referenced by queried objects.
repo.*repo-type.operation.resource-mapping*	Timer	Rate of initiations of a CRUDPAQ operation to a repository datasource.
router.*path-name.action.action-type*	Timer	Rate of actions over the router, and time taken to perform this operation.
router.*path-name*.create	Timer	Rate of creates over the router, and time taken to perform this operation.
router.*path-name*.delete	Timer	Rate of deletes over the router, and time taken to perform this operation.
router.*path-name*.patch	Timer	Rate of patches over the router, and time taken to perform this operation.
router.*path-name*.query.queryExpression	Timer	Rate of queries with queryExpression completed over the router, and time taken to perform this operation.
router.*path-name*.query.queryFilter	Timer	Rate of queries with queryFilter completed over the router, and time taken to perform this operation.
router.*path-name*.read	Timer	Rate of reads over the router, and time taken to perform this operation.
router.*path-name*.update	Timer	Rate of updates over the router, and time taken to perform this operation.

API Metric Name	Type	Description
scheduler.job-store. repo.*operation*.*scheduler-object*	Timer	Time spent storing scheduled jobs in the repository.
scheduler.*type*.*operation*	Timer	Execution rate of scheduler requests.
script.*script-name*.*request-type*	Timer	Rate of calls to a script and time taken to complete.
selfservice.user.password.reset	Summary	Count of all successful user self-service password resets.
selfservice.user. registration.*registration-type*	Summary	Count of all successful user self-service registrations by registration type.
selfservice.user. registration.*registration-type*.*provider*	Summary	Count of all successful user self-service registrations by registration type and provider.
sync.create-object	Timer	Rate of requests to create a target object, and time taken to perform the operation.
sync.delete-target	Timer	Rate of requests to delete a target object, and time taken to perform the operation.
sync.objectmapping.*mapping-name*	Timer	Rate of configurations applied to a mapping.
sync.queue.*mapping-name*.*action*.acquire	Timer	Rate of acquisition of queued synchronization events from the queue.
sync.queue.*mapping-name*.*action*.discard	Timer	Rate of deletion of synchronization events from the queue.
sync.queue.*mapping-name*.*action*.execution	Timer	Rate at which queued synchronization operations are executed.
sync.queue.*mapping-name*.*action*.failed	Summary	Number of queued synchronization operations that failed.
sync.queue.*mapping-name*.*action*. precondition-failed	Summary	Number of queued synchronization events that were acquired by another node in the cluster.
sync.queue.*mapping-name*.*action*.rejected-executions	Summary	Number of queued synchronization events that were rejected because the backing thread-pool queue was at full capacity and the thread-pool had already allocated its maximum configured number of threads.
sync.queue.*mapping-name*.*action*.release	Timer	Rate at which queued synchronization events are released.
sync.queue.*mapping-name*.*action*.release-for-retry	Timer	Times the release of queued synchronization events after a failure and before exceeding the retry count.
sync.queue.*mapping-name*.*action*.submit	Timer	Rate of insertion of synchronization events into the queue.

API Metric Name	Type	Description
`sync.queue.`*`mapping-name`*`.poll-pending-events`	Timer	The latency involved in polling for synchronization events.
`sync.raw-read-object`	Timer	Rate of reads of an object.
`sync.source.assess-situation`	Timer	Rate of assessments of a synchronization situation.
`sync.source.correlate-target`	Timer	Rate of correlations between a target and a given source, and time taken to perform this operation.
`sync.source.determine-action`	Timer	Rate of determinations done on a synchronization action based on its current situation.
`sync.source.perform-action`	Timer	Rate of completions of an action performed on a synchronization operation.
`sync.target.assess-situation`	Timer	Rate of assessments of a target situation.
`sync.target.determine-action`	Timer	Rate of determinations done on a target action based on its current situation.
`sync.target.perform-action`	Timer	Rate of completions of an action performed on a target sync operation.
`sync.update-target`	Timer	Rate of requests to update an object on the target, and the time taken to perform this operation.
`user.login.`*`user-type`*	Summary	Count of all successful logins by user type.
`user.login.`*`user-type`*`.`*`provider`*	Summary	Count of all successful logins by user type and provider.
`virtual-properties-from-relationships.not-found.`*`virtual_properties.resource_collection_relationship_field`*	Summary	Number of 404 responses encountered when querying the `resource_collection`/`relationship_field` specified in the traversal_depthX tag for the most recent X.
`virtual-properties-from-relationships.unsatisfied-temp-constraint.`*`virtual_properties.resource_collection_relationship_field`*	Summary	Number of edges skipped due to an unsatisfied temporal constraint on either the edge or the referred-to vertex. Encountered when querying the resource collection and relationship field at the traversal_depthX tag for the most recent X.
`virtual-properties-from-relationships.`*`virtual_properties.resource_collection_relationship_field`*	Timer	Time spent traversing relationship fields to calculate the specified virtual properties. The managed objects linked to by the traversal relationship fields define a tree, whose root is the virtual property host. This object tree is traversed depth-first, with the traversal_depthX corresponding

API Metric Name	Type	Description
		to the latency involved with each relationship traversal. Traversal_depth0 corresponds to the first relationship field traversed. Because the tree is traversed depth-first, traversal_depthX will subsume all of the traversal latencies for all traversal_depth Y, where Y>X.

Prometheus Metrics

Metrics accessed through the Prometheus endpoint are prepended with `idm_` and use underscores between words, for example `idm_recon_target_phase_seconds`. The following table lists the Prometheus metrics available in IDM:

Prometheus Metrics available in IDM

Prometheus Metric Name	Type	Description
`idm_audit{audit_topic=`*audit-topic*`}`	Summary	Count of all audit events generated of a given topic type.
`idm_field_augmentation{origin-type=edge}`	Timer	Rate of reading response objects, to fulfill the `_fields` requested (when the fields were not populated by the initial repo query).
`idm_field_augmentation{origin-type=vertex}`	Timer	Rate of reading response objects, to fulfill the `_fields` requested (when the fields were not populated by the initial repo query).
idm_filter_seconds{action=*action*,filter_type=*filter-type*,script_name=*script-name*}	Timer	Rate at which filter scripts are executed, per action. Monitors scripted filters and delegated admin.
`idm_icf_`*system-identifier*`_objectClass_query__queryExpression_seconds`	Timer	Rate of ICF query executions with queryExpression, and time taken to perform this operation.
`idm_icf_`*system-identifier*`_objectClass_query__queryFilter_seconds`	Timer	Rate of ICF query executions with queryFilter, and time taken to perform this operation.
`idm_icf_`*system-identifier*`_objectClass_query__queryId_`*queryId*`_seconds`	Timer	Rate of ICF query executions with queryId, and time taken to perform this operation.
`idm_icf_`*system-identifier*`_objectClass_query__UNKNOWN_seconds`	Timer	Rate of ICF query executions when the query type is UNKNOWN, and time taken to perform this operation.
`idm_internal_`*managed-object*`_relationship_fetch_relationship_fields_seconds`	Timer	Rate of fetch operations of relationship fields for internal objects.

Prometheus Metric Name	Type	Description
idm_internal_*managed-object*_relationship_ get_relationship_value_for_resource_ seconds	Timer	Query rate on relationship values for internal objects.
idm_internal_*managed-object*_relationship_ validate_relationship_fields_seconds	Timer	Rate of validate operations of relationship fields for internal objects.
idm_internal_*managed-object*_ script_*script-name*_seconds	Timer	Rate of script executions on internal objects.
idm_internal_seconds{managed_ object=*managed-object*, operation=*operation*>}	Timer	Rate of operations on internal objects.
idm_managed_field_augmentation_seconds	Timer	Rate of responses requiring field augmentation. When the repository is unable to retrieve all the data in a single call, IDM performs additional read operations to complete (augment) the missing data.
idm_managed_*managed-object*_relationship_ fetch_relationship_fields_seconds	Timer	Rate of fetches of relationship fields of a managed object.
idm_managed_*managed-object*_relationship_ get_relationship_value_for_resource_ seconds	Timer	Rate of queries to get relationship values for a resource on a managed object.
idm_managed_*managed-object*_relationship_ validate_relationship_fields_seconds	Timer	Rate of validations of relationship fields of a managed object.
idm_managed_*managed-object*_script_*script-name*_seconds	Timer	Rate of executions of a script on a managed object.
idm_managed_object_handle_temporal_ constraints_on_create	Timer	Latency of enforcing temporal constraints on role objects during object creation.
idm_managed_object_handle_temporal_ constraints_on_delete	Timer	Latency of enforcing temporal constraints on role objects during object deletion.
idm_managed_object_handle_temporal_ constraints_on_update	Timer	Latency of enforcing temporal constraints on role objects during object update.
idm_managed_relationship_handle_temporal_ constraints_on_create	Timer	Latency of enforcing temporal constraints on relationship grants during edge creation.
idm_managed_relationship_handle_temporal_ constraints_on_delete	Timer	Latency of enforcing temporal constraints on relationship grants during edge deletion.
idm_managed_relationship_handle_temporal_ constraints_on_update	Timer	Latency of enforcing temporal constraints on relationship grants during edge update.
idm_managed_relationship_validate_read_ relationship_endpoint_edges_seconds	Timer	Rate of reads on relationship endpoint edges for validation.

Prometheus Metric Name	Type	Description
`idm_managed_seconds{managed_object=`*`managed-object`*`,operation=`*`operation`*`}`	Timer	Rate of operations on a managed object.
`idm_null_array_filter.augmentation`*`requestType`*	Timer	Time spent in filter which maps non-nullable, null-valued array fields to an empty array. This filter is traversed for all repo access relating to internal and managed objects.
`idm_recon-assoc-entry_merged-query_merge-results`	Timer	Rate of merge operations after source and/or target objects have been retrieved during a merged query of recon association entries.
`idm_recon-assoc-entry_merged-query_page-assoc-entries`	Timer	Rate of individual paged recon association entry queries during a merged query. More than one page of entries might be requested to build a single page of merged results.
`idm_recon-assoc-entry_merged-query_query-source`	Timer	Rate of source object retrieval via query when merging source objects to recon association entries.
`idm_recon-assoc-entry_merged-query_query-target`	Timer	Rate of target object retrieval via query when merging target objects to recon association entries.
`idm_recon_association-persistence{recon-id=`*`reconId`*`,operation=`*`operation`*`}`	Timer	The time taken to persist association data. The *operation* can be `source`, `target`, or `amendsource`, depending on whether data is being produced for a source-phase or target-phase recon association, or to amend the association for a specific source.
`idm_recon_id_queries_phase_seconds`	Timer	Rate of executions of the id query phase of a reconciliation, and time taken to perform this operation.
`idm_recon_seconds`	Timer	Rate of executions of a full reconciliation, and time taken to perform this operation.
`idm_recon_source_phase_page_seconds`	Timer	Rate of pagination executions of the source phase of a reconciliation, and time taken to perform this operation.
`idm_recon_source_phase_seconds`	Timer	Rate of executions of the source phase of a reconciliation, and time taken to perform this operation.
`idm_recon_target_phase_seconds`	Timer	Rate of executions of the target phase of a reconciliation, and time taken to perform this operation.

FORGEROCK®

Prometheus Metric Name	Type	Description
`idm_repo_adhoc-expression_relationship_seconds{operation=`*`operation`*`,repo_type=`*`repo-type`*`}`	Timer	Rate of filtered queries (using native query expressions) on the relationship table. This metric measures the time spent making the query (in ms), and the number of times the query is invoked.
`idm_repo_adhoc-filter_relationship_seconds{operation=`*`operation`*`,repo_type=`*`repo-type`*`}`	Timer	Rate of filtered queries (using the `_queryFilter` parameter) on the relationship table. This metric measures the time spent making the query (in ms), and the number of times the query is invoked.
`idm_repo_execute_seconds{operation=create_properties,repo_type=`*`repo-type`*`,resource_mapping=`*`resource-mapping`*`}`	Timer	Rate of execution time on the JDBC database for the `create_properties` operations. This operation is performed for every generic object `create` when it persists the searchable properties. The rate measured here does not include the time taken to obtain a connection to the database from the connection pool. The physical connections to the database have already been established inside the connection pool.
`idm_repo_execute_seconds{operation=`*`operation`*`,repo_type=`*`repo-type`*`,resource_mapping=`*`resource-mapping`*`}`	Timer	Rate of execution time on the JDBC database for CRUD operations. This rate does not include the time taken to obtain a connection to the database from the connection pool. The physical connections to the database have already been established inside the connection pool.
`idm_repo_execute_seconds{operation="query",queryType=`*`queryFilter`*`|`*`queryId`*`,repo_type=`*`repo-type`*`,resource_mapping=`*`resource-mapping`*`}`	Timer	Rate of execution time on the JDBC database for queries (either `queryFilter` or `queryId`). This rate does not include the time taken to obtain a connection to the database from the connection pool. The physical connections to the database have already been established inside the connection pool.
`idm_repo_get_connection_seconds{repo_type=`*`repo-type`*`}`	Timer	Rate of retrievals of a repository connection.
`idm_repo_jdbc_cache_objecttypes_count{event="`*`hit`*`|`*`miss`*`",type=`*`resource-mapping`*``	Count	Counts the usage statistics of the `objecttypeid` cache, which maps an object type to its `objecttypeid`. The expected count is a small number of misses (sometimes, only one) and the remainder of hits.

FORGEROCK

Prometheus Metric Name	Type	Description
idm_repo_jdbc_relationship_edge_execute_ seconds{joinedToVertex=*joinedToVertex*>	Timer	Time (ms) spent running the Edge->Vertex relationship join query on the database and collecting the result set.
idm_repo_jdbc_relationship_execute_ seconds	Timer	Rate of relationship graph query execution times.
idm_repo_jdbc_relationship_process_ seconds	Timer	Rate of relationship graph query result processing times.
idm_repo_raw__queryid_credential_*queryId*_ seconds	Timer	Rate of executions of a query with queryId at a repository level, and time taken to perform this operation.
idm_repo_relationship_ count{operation=*operation*,origin_ type=*origin_type*,repo_type=*repo_type*, stage=*stage*}	Timer	Time (ms) spent in the various phases to retrieve relationship expanded data referenced by queried objects.
idm_repo_relationship_ seconds{operation=*operation*,repo_ type=*repo-type*}	Timer	Rate of CRUDPAQ operations to a repository datasource for a generic/ explicit/relationship mapped table.
idm_repo_seconds{action_name=*action-name*, command=*command*,operation=*operation*,repo_ type=*repo-type*,resource_mapping=*resource-mapping*}	Timer	Rate of actions to a repository datasource for a generic/explicit mapped table.
idm_repo_seconds{operation=*operation*, repo_type=*repo-type*,resource_ mapping=*resource-mapping*}	Timer	Rate of initiations of a CRUDPAQ operation to a repository datasource.
idm_router_*path-name*_action_*action-type*_ seconds	Timer	Rate of actions over the router, and time taken to perform this operation.
idm_router_*path-name*_create_seconds	Timer	Rate of creates over the router, and time taken to perform this operation.
idm_router_*path-name*_delete_seconds	Timer	Rate of deletes over the router, and time taken to perform this operation.
idm_router_*path-name*_patch_seconds	Timer	Rate of patches over the router, and time taken to perform this operation.
idm_router_*path-name*_query_ queryExpression_seconds	Timer	Rate of queries with queryExpression completed over the router, and time taken to perform this operation.
idm_router_*path-name*_query_queryFilter_ seconds	Timer	Rate of queries with queryFilter completed over the router, and time taken to perform this operation.
idm_router_*path-name*_read_seconds	Timer	Rate of reads over the router, and time taken to perform this operation.
idm_router_*path-name*_update_seconds	Timer	Rate of updates over the router, and time taken to perform this operation.

Prometheus Metric Name	Type	Description
idm_scheduler_job_store_repo_seconds{operation=*operation*,scheduler_object*scheduler-object*}	Timer	Time spent storing scheduled jobs in the repository.
idm_scheduler_seconds{operation=*operation*,type=*type*}	Timer	Execution rate of scheduler requests.
idm_script *script-name_request-type*	Timer	Rate of calls to a script and time taken to complete.
idm_selfservice_user_password_reset	Summary	Count of all successful user self-service password resets.
idm_selfservice_user_registration{provider=*provider*,reg_type=*registration-type*}	Summary	Count of all successful user self-service registrations by registration type and provider.
idm_selfservice_user_registration{reg_type=*registration-type*}	Summary	Count of all successful user self-service registrations by registration type.
idm_sync_create_object_seconds	Timer	Rate of requests to create an object on the target, and the time taken to perform this operation.
idm_sync_delete_target_seconds	Timer	Rate of requests to delete an object on the target, and the time taken to perform this operation.
idm_sync_objectmapping_seconds{mapping_name=*mapping-name*}	Timer	Rate of configurations applied to a mapping.
idm_sync_queue_acquire{mapping_name=*mapping-name*, action=*action*}	Timer	Rate of acquisition of queued synchronization events from the queue.
idm_sync_queue_discard{mapping_name=*mapping-name*, action=*action*}	Timer	Rate of deletion of synchronization events from the queue.
idm_sync_queue_execution{mapping_name=*mapping-name*, action=*action*}	Timer	Rate at which queued synchronization operations are executed.
idm_sync_queue_failed{mapping_name=*mapping-name*, action=*action*}	Summary	Number of queued synchronization operations that failed.
idm_sync_queue_poll_pending_events{mapping_name=*mapping-name*}	Timer	The latency involved in polling for synchronization events.
idm_sync_queue_precondition_failed{mapping_name=*mapping-name*, action=*action*}	Summary	Number of queued synchronization events that were acquired by another node in the cluster.
idm_sync_queue_rejected_executions{mapping_name=*mapping-name*, action=*action*}	Summary	Number of queued synchronization events that were rejected because the backing thread-pool queue was at full capacity and the thread-pool had already allocated its maximum configured number of threads.

Prometheus Metric Name	Type	Description
idm_sync_queue_release_for_retry{mapping_name=*mapping-name*, action=*action*}	Timer	Times the release of queued synchronization events after a failure and before exceeding the retry count.
idm_sync_queue_release{mapping_name=*mapping-name*, action=*action*}	Timer	Rate at which queued synchronization events are released.
idm_sync_queue_submit{mapping_name=*mapping-name*, action=*action*}	Timer	Rate of insertion of synchronization events into the queue.
idm_sync_raw_read_object_seconds	Timer	Rate of reads of an object.
idm_sync_source_assess_situation_seconds	Timer	Rate of assessments of a synchronization situation.
idm_sync_source_correlate_target_seconds	Timer	Rate of correlations between a target and a given source, and time taken to perform this operation.
idm_sync_source_determine_action_seconds	Timer	Rate of determinations done on a synchronization action based on its current situation.
idm_sync_source_perform_action_seconds	Timer	Rate of completions of an action performed on a synchronization operation.
idm_sync_target_assess_situation_seconds	Timer	Rate of assessments of a target situation.
idm_sync_target_determine_action_seconds	Timer	Rate of determinations done on a target action based on its current situation.
idm_sync_target_perform_action_seconds	Timer	Rate of completions of an action performed on a target sync operation.
idm_sync_update_target_seconds	Timer	Rate of requests to update an object on the target, and the time taken to perform this operation.
idm_user_login{user_type=*user-type*}	Summary	Count of all successful logins by user type.
idm_user_login_total{provider=*provider*, user_type=*user-type*}	Summary	Count of all successful logins by user type and provider.
idm_virtual_properties_from_relationships{virtual_properties=*calculated-virtual-properties*, traversal_depthX=*traversal-origin-resource-collection and traversal relationship*,not_found}	Summary	Number of 404 responses encountered when querying the resource_collection/relationship_field specified in the traversal_depthX tag for the most recent X. X corresponds to the relationship field sequence.
idm_virtual_properties_from_relationships{virtual_properties=*calculated-virtual-properties*, traversal_depthX=*traversal-origin-resource-collection and traversal relationship*,unsatisfied_temp_constraint}	Summary	Number of edges skipped due to an unsatisfied temporal constraint on either the edge or the referred-to vertex. Encountered when querying the resource collection and relationship field at the traversal_depthX tag for the most recent

Prometheus Metric Name	Type	Description
		X. X corresponds to the relationship field sequence.
`idm_virtual_properties_from_relationships{virtual_properties=`*`calculated-virtual-properties`*`, traversal_depthX=`*`traversal-origin-resource-collection and traversal relationship`*`}`	Timer	Time spent traversing relationship fields to calculate the specified virtual properties. The managed objects linked to by the traversal relationship fields define a tree, whose root is the virtual property host. This object tree is traversed depth-first, with the traversal_depthX corresponding to the latency involved with each relationship traversal. Traversal_depth0 corresponds to the first relationship field traversed. Because the tree is traversed depth-first, traversal_depthX will subsume all of the traversal latencies for all traversal_depth Y, where Y>X. X corresponds to the relationship field sequence.

IDM Glossary

correlation query	A correlation query specifies an expression that matches existing entries in a source repository to one or more entries in a target repository. A correlation query might be built with a script, but it is not the same as a correlation script. For more information, see *"Correlating Source Objects With Existing Target Objects"* in the *Synchronization Guide*.
correlation script	A correlation script matches existing entries in a source repository, and returns the IDs of one or more matching entries on a target repository. While it skips the intermediate step associated with a `correlation query`, a correlation script can be relatively complex, based on the operations of the script.
entitlement	An entitlement is a collection of attributes that can be added to a user entry via roles. As such, it is a specialized type of `assignment`. A user or device with an entitlement gets access rights to specified resources. An entitlement is a property of a managed object.
JCE	Java Cryptographic Extension, which is part of the Java Cryptography Architecture, provides a framework for encryption, key generation, and digital signatures.
JSON	JavaScript Object Notation, a lightweight data interchange format based on a subset of JavaScript syntax. For more information, see the JSON site.
JSON Pointer	A JSON Pointer defines a string syntax for identifying a specific value within a JSON document. For information about JSON Pointer syntax, see the JSON Pointer RFC.

JWT	JSON Web Token. As noted in the JSON Web Token draft IETF Memo, "JSON Web Token (JWT) is a compact URL-safe means of representing claims to be transferred between two parties." For IDM, the JWT is associated with the `JWT_SESSION` authentication module.
managed object	An object that represents the identity-related data managed by IDM. Managed objects are configurable, JSON-based data structures that IDM stores in its pluggable repository. The default configuration of a managed object is that of a user, but you can define any kind of managed object, for example, groups or roles.
mapping	A policy that is defined between a source object and a target object during reconciliation or synchronization. A mapping can also define a trigger for validation, customization, filtering, and transformation of source and target objects.
OSGi	A module system and service platform for the Java programming language that implements a complete and dynamic component model. For more information, see What is OSGi? Currently, only the Apache Felix container is supported.
reconciliation	During reconciliation, comparisons are made between managed objects and objects on source or target systems. Reconciliation can result in one or more specified actions, including, but not limited to, synchronization.
resource	An external system, database, directory server, or other source of identity data to be managed and audited by the identity management system.
REST	Representational State Transfer. A software architecture style for exposing resources, using the technologies and protocols of the World Wide Web. REST describes how distributed data objects, or resources, can be defined and addressed.
role	IDM distinguishes between two distinct role types - provisioning roles and authorization roles. For more information, see "Managed Roles" in the *Object Modeling Guide*.
source object	In the context of reconciliation, a source object is a data object on the source system, that IDM scans before attempting to find a corresponding object on the target system. Depending on the defined mapping, IDM then adjusts the object on the target system (target object).
synchronization	The synchronization process creates, updates, or deletes objects on a target system, based on the defined mappings from the source system. Synchronization can be scheduled or on demand.

system object

A pluggable representation of an object on an external system. For example, a user entry that is stored in an external LDAP directory is represented as a system object in IDM for the period during which IDM requires access to that entry. System objects follow the same RESTful resource-based design principles as managed objects.

target object

In the context of reconciliation, a target object is a data object on the target system, that IDM scans after locating its corresponding object on the source system. Depending on the defined mapping, IDM then adjusts the target object to match the corresponding source object.