Forcepoint Behavioral Analytics Administration and Troubleshooting

Administration and Troubleshooting | v3.3.x, v3.4.x | 17-Feb-2021

Overview

This Administration Manual provides information for administering and fine tuning software processes affecting the Forcepoint Behavioral Analytics UI, as well as troubleshooting FAQs and common How To's. The document provides rationale behind application performance and considerations to keep in mind when using the interface.

Frontend Configuration

Monitored Entities

For performance reasons, we only compute analytics and display the results on the Analytic Dashboard and Behaviors page on Entities with a **Monitored Entity** Attribute value set to True. The Monitored Entity attribute controls whether an Entity is included in the Elasticsearch aggregations used in computing UTA results and in the associated Elasticsearch cache writes.

Keep the following consequences in mind:

- Only Monitored Entities with values set to True contribute to model score normalization.
- Including an outlier Entity can distort the model scores for other Entities.

  For example, an email counting model might produce strange results if an outlier Entity, such as a bulk mailing address, has a Monitored Entity Attribute.
- If the Monitored Entities list is too small, there may not be matching events for the Models given a user's Entitlement and the dates selected.
- Adding unnecessary Monitored Entities can degrade UTA performance, particularly because it increases the number of documents that the compute_dashboard job has to write.

- Adding unnecessary Monitored Entities can increase the memory footprint of UTA caching.

Models

Performance Concerns

Reuse Models whenever possible. UTA performs computations and writes cache entries for each separate Model in a Scenario. However, one Model reused in multiple Scenarios only contributes workload for the one Model.

UTA performance testing has shown acceptable results when computing approximately 30 Models. Defining significantly more than 30 Models has an unknown impact on computation time, storage space used, and memory footprint.

Models that use outlier probabilities are more expensive than Models that do not. When outlier probabilities don't provide significant benefits, do not use outliers.

Defining Useful Models

Model With No Features

A Model without any Features is usually used to either count the number of Events that meet certain criteria or to sum an Attribute on those Events.

- Primary role determines what Events and Entities are flagged by this Model. For example, a role of Sender means an Event only matters to an Entity's results on Behaviors or Analytic Dashboard if that Entity is a Sender on that Event. Events without Senders have no impact.
- Advanced search is an optional field that can contain an RQL filter.
- **Qualifying Events:** Given Entity A and a specified time interval, a qualifying Event is one that matches the advanced search filter and has Entity A in the primary role.
- The aggregation method determines how scores are computed. For a Model with no Features, the useful aggregation methods are:
  - **Event Count:** How many qualifying events are there?
  - **Secondary Role Cardinality:** (Requires selecting a secondary role) Across qualifying events, how many different entities appear in the secondary role?
  - **Attribute Sum:** (Requires selecting an event attribute) Across qualifying events, what is the total value for the selected attribute?
  - **Attribute Max:** (Requires selecting an event attribute) Across qualifying events, what is the maximum value for the selected attribute?

Model With Features

A Model with Features is usually used when event scoring is important. For example, events where an Entity sends an email with an attachment may be of interest, but it

might be more interesting when those emails are sent on the weekend. In this case, using Features is sensible.

In addition to the information above for Models with no Features, the following applies to Models with Features:

- **Qualifying Events:** Given Entity A and a specified time interval, a qualifying event is one that matches the advanced search filter and has Entity A in the primary role. When Features are defined, the scoring algorithm additionally requires that the event match at least one Feature.
- **Model Event Score:** Roughly, each feature that matches on an event contributes to the model score for that event. The contribution of a given feature is greater if the score in question is high/rare. For categorical features, a feature contributes more if it is less common. For numeric features, a feature contributes more if its value is greater.
- The aggregation methods that require features are:
  - **Aggregate Model Event Score:** The sum of model scores, across the qualifying events whose model score is significantly greater than average.
  - **Max Model Event Score:** The highest model score across all qualifying events.
- To use features to filter without concern for their impact on model score, it is recommended to use RQL rather than adding features to the model. The result is more clear in its intent and interpretation, and allows more expressiveness in terms of AND vs OR relationships.

Outlier Probabilities

- Outlier probabilities extend the information above to account for usual vs unusual time intervals.
- For example, say a model without outlier probabilities assigns a score to each interval, e.g., event count might assign the value of 13 if it observes 13 events in a time interval. If that model had outlier probabilities turned on, the score for that interval would then be between 0 and 1, signifying the probability that 13 (the event count) is an unusual outlier. To make this decision, the outlier probability algorithm looks at the prior ten intervals and compares them with the present interval.
  - **Outlier Above** gives time intervals with an unusually high value a high score. Time intervals with an unusually low value get a 0 score.
  - **Outlier Below** gives time intervals with an unusually low value a high score. Time intervals with an unusually high value get a 0 score.
  - **Outlier General** gives time intervals with an unusual value a high score, regardless of whether that value is itself high or low.

Scenarios

Performance Concerns

Each Scenario requires its own additional computation and cache writes, along with the extra computation and extra cache writes necessary for each model in a Scenario. This suggests that restricting the set of **analytic scenarios** (scenarios used by the Analytics Dashboard) will improve performance while still allowing users to explore the data by defining their own scenarios on the Behaviors page.

Defining Useful Scenarios

For any given entity time interval, a model produces a score between 0 and 1. This is true of both event models (with or without outliers) and entity models. A scenario aggregates those into a single score by aggregating these scores (take the logarithm of each model score, average them together, then exponentiate). This scoring process has several important implications:

- If a model has "No Matching Events" for a given time interval, it is ignored as far as calculating the scenario score. That is, if we have five models and Model Five shows "No Matching Events", we only average four models, not five. This can sometimes produce unintuitive sorting results on the Behaviors page.
- If a model periodically has one very high model score, then an entity time interval with only that single model matching any events might actually outscore another time interval with several models matching, but each with a low score.
- No normalization is performed on entity models. This means that the range of values set for an entity model essentially determines the weight of that entity model.
- Outlier models very easily drop to a score of 0 or climb to a score of 1. This makes them problematic for smooth, consistent sorting.

Scenarios best practice:

- Reuse models from other scenarios when possible
- Do not contain too many separate models
- Do not contain too many entity models or too highly weighted entity models
- Contain models with a consistent range of scores (e.g., many outlier models or many non-outlier models)

Setting up Analytics

Features, Models and Scenarios are all manually setup with backend commands that enable the use of front-end Forcepoint Behavioral Analytics configuration tools. The following section describes how to setup the front-end UI on the backend, and how to use the front-end tools to properly configure features, models and Scenarios.

Backend Setup

Setting Up the “Behaviors” Page:

1. **Standard MDS Setup:** ingest Events, resolve Entities, add Features (any Features), run Feature scoring.
2. **Behavioral Analyst Role:** Both the Analytics Dashboard and the Behaviors page are locked behind the Behavioral Analyst role. Make sure this role is enabled for the user account.
3. **Monitored Entities:** Unified Theory of Analytics (UTA) only computes statistics for a subset of Entities called monitored Entities. Other Entities simply won't show up on the “Analytics Dashboard” and “Behaviors” page. This is intended to represent internal entities / entities of interest. To set an entity as monitored, go to the “Entities” page and add a boolean entity attribute, "Monitored Entity". Set this attribute to *True*.

> **Note**
> It is recommended to set at least a few monitored entities for local testing and to flag all employees for production deployments.

4. **Create Model(s):** Any Models are acceptable, so long as they capture Events that include some of the Monitored Entities.

   For example, if there are emails in the data set, a basic Model for testing would be an email counter: **sender** as Primary Role for Aggregation, **recipient** as Secondary Role for Aggregation, **event count** for the Aggregation Method, and **mode:email** in Advanced Search.
5. **Create Scenarios:** To add Models to a Scenario, go to the “Behaviors” page. The left-most panel is where Scenarios are defined. Give your Scenario a name, select an end date, and add Models with the + sign. Save the Scenario (save icon is in the top-right corner).
6. **Using the Behaviors Page:** When the Scenario is fully configured, the Behaviors page is ready to display the results. Hit Apply (top-right corner) to generate the heatmap.

Setting Up Analytics Dashboard:

1. **Setup Behaviors:** See above.
2. **Reset the Cache:** If you've used UTA (“Behaviors” page or “Analytics Dashboard”) before and now would like to add/modify/delete Models or Scenarios, then you will have to clear the cache. “Analytics Dashboard” uses cached results to improve performance. This cache is not automatically updated when you change Models or Scenarios. When you want to see updated results on the “Analytics Dashboard,” you must first clear the cache (below).
3. **Reset Cache Curl**

   ```
   curl -XPOST http://localhost:8080/reference/analytics/clear_cache
   ```

4. **Background Computation:** To test the “Analytics Dashboard,” you have to first run an asynchronous computation job that prepares the “Analytics Dashboard” cache. In actual deployments, this is something that we expect Ops to control via a cron job, but for developer deployments, this will have to be handled manually.
5. Precompute Curl

   ```
   curl -XPOST -H'Content-Type: application/json' http://
   localhost:8080/reference/analytics/compute_dashboard -d'{}'
   | jq '.'
   ```
6. **Using the “Analytics Dashboard”:** You can now navigate to the “Analytics Dashboard” where you should see results. Creating more Scenarios (see 4 and 5) will make this graph more interesting.
7. **Reset the Cache:** If you decide to add/modify/delete Models or Scenarios, parts of your cache will either be evicted (in the case of Models) or become stale (in the case of Scenarios). If you add/modify/delete Models or Scenarios, you should return to **Step 2**.
8. **Additional note:** The “Analytic Dashboard” overrides the end date and Entity Time Interval configuration from the “Behaviors” page. For example, if you create Scenario A with an end date of 1/1/2015 and an Entity Time Interval size of 5 days, the “Analytic Dashboard” will display Scenario A's results with an end date of today and an Entity Time Interval size of 24 hours. On the “Behaviors” page, Scenario A will continue to operate with the original configuration.

Additional Configuration

1. Score Comparison Attributes: By default, the Score Comparison chart on the Analytics Dashboard (bottom right) groups entities by their Location and Department attributes. To define these attributes for a given Entity, go to the Entity pages for the set Monitored Entities and add a Location and/or Department attribute. If you want to use different attributes, this can be reconfigured by setting the value in AppConfig.

   Score Comparison Categories Curl

   ```
   curl -XPUT "localhost:8080/reference/config/analytics" -d
   '{"score_comparison_categories": "Occupation,Clearance"}'
   ```
2. Analytic Scenarios: By default, every saved Scenario is shown on the Analytics Dashboard. For production deployments, this is probably undesirable, as it means end users can impact the Entity Scores while defining Scenarios for their own use. At present, the Forcepoint Behavioral Analytics solution for this is a list of Scenario IDs stored in AppConfig. If this list is defined, only Scenarios from this list will show up on the Analytics Dashboard. To define this, first curl MDS and identify the target Scenarios.
3. Score Comparison Categories Curl

   ```
   curl -GET "localhost:8080/reference/analytics/collection
   ```
4. The key field is id_of_root. Once you have all the id_of_root fields for Scenarios you want to analyze, you insert them into AppConfig with:
5. Score Comparison Categories Curl

```
curl -k -XPUT https://localhost:8080/reference/config/
analytics -d'{"analytic_scenarios":
["AVjgisANKkiDYuc29app","AVglIn_Vejq51xgk7I7T"}'
```

a. Assuming our Scenarios have id_of_root of "AVjgisANKkiDYuc29app" and "AVglIn_Vejq51xgk7I7T"

Performance

The **Behavioral Analyst** role controls who can see the Analytic Dashboard and the Behaviors page. The **Developer** role, among other things, controls who can see entity models. If a user has the Behavioral Analyst role, but not the Developer role, they will not be able to see entity models on either the Analytic Dashboard or the Behaviors page.

Entitlements require caching a full set of results for each entitlement. Subsequently, performance degrades as you add more users with a Behavior Analyst role, and give these users more entitlements.

So, for example, if five users share the same entitlement and all have the Behavioral Analyst role, we only have to cache results once (their shared entitlement means they'll see the same results). However, if those five users have five different entitlements, we have to compute and cache everything five times to account for the entitlement differences.

UTA has two types of computation jobs that can see poor performance:

UTA runs computations on-demand on the **Behaviors** page. This is a relatively cheap computation, so as the Analytic Dashboard performs well, the Behaviors Page will have acceptable performance.

Computation cost on the **Analytic Dashboard** is much higher. If results have not been cached before, the cost of computing the Analytic Dashboard is about 24 times the cost of computing a single model for the Behaviors page, per model or scenario. However, because we usually have multiple Scenarios, with multiple models in each, the cost is usually hundreds of times higher.

However, in production environments the Analytic Dashboard usually computes much faster, because we run the computation every hour instead of the whole prior week, meaning it only has to cache results from the last hour. When run every hour, the cost is 1/24th of the full cost of computing the whole Dashboard.

Two other factors influence the cost of computing the Analytic Dashboard: number of **monitored entities** and number of users with the **Behavioral Analyst** role, both discussed below.

If you see excessively poor performance from the **/compute_dashboard** job, check:

- How many analytic Scenarios are configured? (Default: All Scenarios)
- How many Models are defined across the analytic Scenarios?
- How often is the compute_dashboard job running? The more often, the better performance per run.

- How many Monitored Entities are there?
- How many users have the Behavioral Analyst role and how many different entitlements do they have?

Review Dashboard Configuration

The Review Dashboard is not populated by subscribed profiles but rather a user's saved searches.

Finding Saved Searches

Saved search information exists in Postgres in two tables:

```
psql -U postgres -t the_ui

select * from filters;
select * from filters_users;
```

The filters table stores the RQL for the search and other properties used by the Dashboard.

Filters column headers:

```
id | integer | not null default
nextval('filters_id_seq'::regclass)
created_at | timestamp with time zone |
updated_at | timestamp with time zone |
title | character varying(255) |
description | character varying(255) |
values | text |
marked_for_escalation | boolean | default false
marked_for_review | boolean | default false
```

The filters_users table stores which users are "subscribed" or associated with a saved search.

Filters_users table headers:

```
id | integer | not null default
nextval('filters_users_id_seq'::regclass)
filter_id | integer | not null
user_id | integer | not null
```

Since we are inserting new rows into the filters_users table manually, we need to make sure the table does not get out of sync with the UI. If it does, users will be unable to create their own saved searches after doing this.

To see what the last ID in the table is, run the following:

```
SELECT MAX(id) FROM filters_users;
```

```
14
```

Now, we need to make sure that the next value the UI will try to set is the next available row id (15 in this example):

```
SELECT nextval('filters_users_id_seq');
15
```

If you are out of sync (i.e. nextval <= max id), you need to perform the following:

```
SELECT setval('filters_users_id_seq', (SELECT MAX(id) FROM
filters_users));
```

Populating the Review Dashboard from the UI

1. Input the RQL to define the saved search into the Advanced Search bar.
2. Click **Save Search.**
3. Enter in descriptive information and click **Save**.
4. Click **Reviews**.
5. Confirm that the Dashboard now has content.

Confirm that the search now appears in the filters table as filter #6 and #6 is now associated with My User ID in the filters_users table:

```
the_ui=# select * from filters_users;
 1 |         1 |      13
 3 |         3 |      38
 6 |         6 |      28
the_ui=# select * from filters;
 6 | 2016-11-22 08:52:14.088-05 | 2016-11-22 08:52:20.987-05
| Selected
for Review       | Events selected for daily review includin
g AFR hits and RedOwl enrichments |
{"externalSender":false,"selected_portfolio_managers":[],"se
lected_secur
ities":[],"selected_labels
":[],"selected_identifiers":[],"attachmentWords":[],"selecte
d_participan
t_ids":[],"participant_akas":[],"selected_interval":"day","w
or
ds":[],"recipientIdsLengthMinimum":null,"attachmentSizeMinim
um":null,"se
lected_date_time_bucket_type":null,"start_date":146138400000
0,
"wordLists":[],"selected_modes":[],"participant_ids":[],"sel
ected_recipi
```

```
ent_ids":[],"externalRecipient":false,"selected_start_date":
nu
ll,"modes":[],"selected_digital_actions":[],"end_date":14798
77199999,"se
lected_date_time_bucket_values":[],"selected_sender_ids":[],
"i
nternalSender":false,"personalSender":false,"selected_end_da
te":null,"se
lectionQDSL":null,"selected_actions":[],"personalRecipient":
fa
lse,"recipientIdsLengthMaximum":null,"selected_digital_modif
ier_ids":[],
"attachmentSizeMaximum":null,"internalRecipient":false,"feat
ur
es":[],"expert_search":"label=\"selected-for-
review\"","digital_action":
null,"hasAttachment":false,"selected_digital_identifiers":[]
,"
Labels":[]}
```

Populating the Review Dashboard from the Backend

Partially through the UI

If a saved search has been created through the UI, associate a pre-existing saved search with another user by modifying the filters_users table. Define the “id” as one not already in user in the table:

```
INSERT INTO filters_users (id, filter_id, user_id) VALUES
(7, 6, 38);
the_ui=# select * from filters_users;
1 | 1 | 13
3 | 3 | 38
6 | 6 | 28
7 | 6 | 38
```

Completely through the backend

> **Note**
> While this is a viable option, it is recommended to opt to partially configure through the UI (process listed above).

Update both the filters and filters_users tables to avoid creating the initial filter through the UI.

```
INSERT INTO filters (id, created_at, updated_at, title,
description, values, marked_for_escalation,
```

```
marked_for_review) VALUES (9, '2016-11-22 09:35:44.243-05',
'2016-11-22 09:45:06.162-05', 'Excluded from Review',
'Events flagged by various rules but not selected (i.e. bulk
senders)', 'changeme', f, t);
```

Note

The table expects an ES query, NOT the RQL.

Example backend query format:


```
{"externalSender":false,"selected_portfolio_managers":[],"se
lected_securities":[],"selected_labels

":[],"selected_identifiers":[],"attachmentWords":[],"selecte
d_participant_ids":[],"participant_akas":[],"selected_interv
al":"day","wor

ds":[],"recipientIdsLengthMinimum":null,"attachmentSizeMinim
um":null,"selected_date_time_bucket_type":null,"start_date":
1461384000000,

"wordLists":[],"selected_modes":[],"selection":"","participa
nt_ids":[],"selected_recipient_ids":[],"externalRecipient":f
alse,"selected

_start_date":null,"modes":[],"selected_digital_actions":[],"
end_date":1479877199999,"selected_date_time_bucket_values":n
ull,"selected_

sender_ids":[],"internalSender":false,"personalSender":false
,"selected_end_date":null,"selectionQDSL":null,"selected_act
ions":[],"pers

onalRecipient":false,"recipientIdsLengthMaximum":null,"selec
ted_digital_modifier_ids":[],"attachmentSizeMaximum":null,"i
nternalRecipie

nt":false,"features":[],"expert_search":"label=\"afr-
account-operations\" label=\"afr-complaints\" label=\"afr-
complaints-jj\" label=\

"afr-conflicts\" label=\"afr-cornwall-capital\" label=\"afr-
fcpa-bribery\" label=\"afr-feargal-hogan\" label=\"afr-
gifts-entertainment

\" label=\"afr-insider-trading\" label=\"afr-kyle-tatz\"
label=\"afr-mnpi\" label=\"afr-pay-to-play\" label=\"afr-
political-research\"

label=\"afr-section13\" label=\"afr-stephen-santrach\"
label=\"afr-strategic-investors\" label=\"afr-sudo\"
label=\"afr-trade-errors\

" label=\"afr-trade-errors-all\" label=\"attachments-to-
personal\" label=\"focal-employee\" (feature:(\"Exclude
Bulk\" > 0) feature:(\

"Exclude Bulk\" > 0) feature:(\"Exclude Calendar Replies\" >
0) feature:(\"Exclude HF Whitelist\" > 0) feature:(\"Exclude
Internal Bul
```

```
k\" > 0) feature:(\"Exclude Sellside\" > 0)
feature:(\"Exclude TPM\" > 0) OR label=\"bulk-message\" OR
label=\"nonreviewable-mode\")",
"digital_action":null,"hasAttachment":false,"selected_digita
l_identifiers":[],"labels":[]}
```

Insert the RQL into the **expert_search** field and save this in a file.

```
\set content `cat /home/kflynn/saved_searches/
excluded_from_review.txt`
update filters set values=:'content' where id=9;
```

Once again, make sure the table is still in sync. If not, update it:

```
SELECT setval('filters_id_seq', (SELECT MAX(id) FROM
filters));
Lastly, update your filters_users table to apply the new
saved search to a user:
INSERT INTO filters_users (id, filter_id, user_id) VALUES
(15, 9, 38);
```

Scoring Jobs

To run a job via the UI:

1. Click **Run Scoring** on the bottom of the Features page.

To run a job via command-line:

1. Run the following curl command:

   ```
   curl -XPOST http://mds1:8080/reference/eventfeatures/jobs
   ```

2. Purge old scores from all events:

   ```
   curl -XPOST http://mds1:8080/reference/eventfeatures/
   jobs?clear_old_features=true
   ```

3. Checking job status the via the UI (using a job ID of 17 as an example):

   ```
   curl http://mds1:8080/jobs/17
   ```

4. Use the Features page in the UI for percentage complete.

Removing an actor from refdata

This functionality is no longer supported.

How To's

Secure Copy File

From the directory that you have the file saved (likely locally) run one of the following commands:

```
scp -i ~/.ssh/id_rsa <fileName> user@mds1-qrc:~/
```

OR

```
curl -XPOST https://api-qrc.ro.internal/communication -d@/
Users/user/Desktop/LongSubject.json -H "Content-Type:
application/json" --insecure
```

Note

If the event being ingested is not an email, change the curl command to:

```
curl -XPOST https://api-qdev/event/bulk --
data-binary @something.json -H content-
type:application/x-json-stream
```

Alternative method of ingesting is:

1. scp file to api
2. ssh into api
3. run following command using "localhost" vs. 'api-dev'

```
curl -XPOST http://localhost:9000/communication/bulk --data-
binary @commsscrubbedLarge.json -H 'content-
type:application/x-json-stream' | jq '.' - This is assuming
you're hitting the communication bulk endpoint.  Change to
the appropriate endpoint and obviously change your file
name.
```

Running Entity Resolution:

Definitions

- **entries**: a list of actorID alias entries, typically read from an uploaded ER key.
- **purgeEntityIndex**: This is no longer supported.
- **addAllEntitiesFromEvents**: This is no longer supported.
- **mergeStrategy**: This is no longer supported.

Upload Key File

Running Locally

```
curl -XPOST -F "file=@/Users/carrie/Documents/BHSamples/
bigERKey.csv;type=text/csv" http://localhost:9000/
resolution_key/upload
```

OR

```
curl -F "file=@ERKey.csv;type=text/csv" http://
localhost:9000/resolution_key/upload | jq .
```

Once Key Uploaded successfully, the following message is displayed:

```
{"id":"AVbIhNw40SVz0iJDyC2p","source":"UPLOADED","name":"ken
key.csv","status":"SUCCESS","started":1472242965558,"finishe
d":1472242965559,"percentCompleted":100.0}
```

- Copy the ID (example - AVbIhNw40SVz0iJDyC2p).

Load/Update Refdata

Running Locally

```
curl -XPOST 'http://localhost:9000/resolution_key/
load?keyId={MYKEYID}&purgeEntityIndex=true'
 | jq .
```

- Replace {MYKEYID} with the KeyID copied from the above.

Updating events in Eventdata

This takes a long time to run in a large data set.

Running Locally

```
curl -XPOST ' HYPERLINK "http://localhost:8080/reference/
disambiguation/run'" http://localhost:8080/reference/
disambiguation/run'
curl -XPOST ' HYPERLINK "http://localhost:8080/reference/
disambiguation/run/streaming'" http://localhost:8080/
reference/disambiguation/run/streaming'
```

After job runs successfully run following:

Running Locally

```
curl -XPOST ' HYPERLINK "http://localhost:8080/reference/
actorreport/refresh/cache'" http://localhost:8080/reference/
actorreport/refresh/cache'
curl -XPOST 'http://localhost:8080/reference/actorreport/
refresh/cache'
```

OR

```
curl -XPOST http://mds1-qrc:8080/reference/actor/refresh/
cache | jq .
```

Correcting Reverse Roles (Roles in refdata but no matching role in Event):

```
curl -XPOST http://<MDS:port>/reference/entity_role/
repair_reverse_orphans | jq .
```

To Recompute the Analytics Dashboard:

Pre-SSL Changes (up to version 2.50.0):

1. 1. ssh into MDS
2. 2. Run:

   ```
   curl -XPOST http://mds1-qrc.ro.internal:8080/reference/
   analytics/clear_cache | jq .
   ```
3. 3. Run:

   ```
   curl -XPOST http://mds1-qrc.ro.internal:8080/reference/
   analytics/compute_dashboard  | jq .
   ```

Post-SSL Changes (from versions 2.50.1):

1. ssh into MDS
2. Run:

   ```
   curl -XPOST -k -u elastic:changeme https://mds1-
   qrc.ro.internal:8080/reference/analytics/clear_cache | jq .
   ```
3. Run:

   ```
   curl -XPOST -k -u elastic:changeme https://mds1-
   qrc.ro.internal:8080/reference/analytics/
   compute_dashboard?endDate=1470110400000 | jq .
   ```

Clear Actor Cache:

```
curl -XPOST 'http://mds1-qdev:8080/reference/actor/refresh/
cache' | jq .
```

Note

In almost all cases when clearing the actor cache, do so on all 3 MDS nodes. To do this, ssh into the correct MDS instance (example: *ssh amartin@mds2-qrc.ro.internal*) and change the *mds1* portion of the curl to either *mds2* or *mds3*.

OR

```
curl -XPOST 'http://mds1-qdev:8080/reference/actorreport/
refresh/cache' | jq .
```

Checking MDS Logs:

1. ssh into mds and then run the following:

   ```
   tail -f /var/log/ro-mds/
   ```

To Kill a Job/Confirm Killing of a Job is working as Expected:

> **Note**
> Curl commands & their endpoints will be different to account for the job being run/killed.

To test with the Bulk Labels job:

1. Find the userID for the logged in user:
   a. ssh user@postgres-qdev.
   b. Next, psql -U postgres.
   c. Then, \connect the_ui.
   d. Then, select * from users.
2. Log into the environment and create a new label (example: "usertest".)

   Make sure to keep the UI open and navigate to the Job Status page to confirm seeing the status change in the UI as well as the console output.
3. To start the bulk label job, run the following command:

   ```
   curl -s -XPUT  -d'{"user_id": 2, "rql":""}' 'mds1-
   qdev.ro.internal:8080/reference/contentlabel/userkill/rql' |
   jq .
   ```

> **Note**
> Put some RQL in between the two quotes in your curl command to bulk a certain amount of events. Keep in mind that the smaller the number of the events, the faster the job will run and the less time will be available to attempt to kill it.

4. As the job is running, the UI shows “RUNNING”.
5. To kill the job:

   ```
   curl -XPOST mds:8080/jobs/22/kill
   ```

> **Note**
> The 22 in the query above is the job's ID. Replace this with the unique job ID.

6. As the job is being killed, the UI shows “KILLING”.
7. To confirm the job status:

   ```
   curl mds1-qrc:8080/jobs/22 | jq .
   ```

8. The job has been killed. The UI shows “KILLED”.

Adding system lexicons to a fresh environment:

```
curl -XPOST 'mds1-qrc.ro.internal:8080/reference/lexicon/
load_defaults?user_id=1'
```

Deleting an event from an environment:

> **Note**
> Find the ID of the event in Kibana OR by using CLI RQL tool (see below).

Once found, run the following command:

```
curl -XDELETE -k -u elastic:changeme https://mds1-
qdev.ro.internal:9200/eventdata_2017-01/alert/
ZG_0Fg4zHJyE9nTJ5xiVElLzHQw
```

Troubleshooting

Error Messages

These messages appear on the Analytic Dashboard to indicate a particular type of common issue.

"There are no monitored entities."

Either no entities have a Monitored Entity attribute set to true, or the Monitored Entity attribute is improperly defined (e.g., not Boolean). Fix this by setting some Monitored Entity boolean attributes.

"No cached analytics exist."

Most likely, this means that there are no cached results or no cached results relevant to your entitlement. This can occur if:

1. The *compute_intervals* job has never been run.
2. The UTA cache was cleared and *compute_intervals* has not yet been run.

3. Your entitlement was not included in the last *compute_intervals* run, probably because your user account didn't have access to Behavioral Analyst yet.

Fix this by running the *compute_intervals* job again.

"Data last updated at..."

The cached results in UTA are stale (by default, the last cached entry is more than four hours older than the last event in eventdata). This usually just means that the *compute_intervals* job hasn't been run in a while.

Fix this by running the *compute_intervals* job again.

"No results matching your Entitlements"

No events matched the combination of models and your entitlements. Your models and/or entitlements and/or the list of monitored entities will need to be changed. Please see the Troubleshooting Guide below and the sections above on choosing good models and Scenarios.

Behaviors page

- Behaviors page shading not consistent: Sometimes an entity might have the same value in two intervals, but show different shading. For example, an event count of 10 might be dark purple in one interval and light purple a couple intervals later. This is (potentially) working as designed, as interval scores are "normalized" by looking at recent historical intervals. So, while an event count of 10 might have been very high on Monday, if some entity has an event count of 50 on Tuesday, then that same event count of 10 is worth less from then on.
- Sorting order appears to be wrong: the response JSON from MDS contains the field "scores_for_ranking". This field is needed to debug issues with sorting on the Behaviors page.

If none of the above apply, start by checking whether any model or Scenario returns the expected results on the Behaviors page. If they do, the problem is likely related to your specific model or Scenario.

If no models or Scenarios work on the Behaviors page, check the Explore page and make sure your entitlements allow you to see at least some data. If you see nothing or less than expected on the Explore page, please configure your entitlements better.

Analytic Dashboard

- The recent intervals graph is not continuous: The Analytic Dashboard is probably still computing. It computes in batches of intervals spaced 1 day apart, so while it's still computing, whole sets of intervals (spaced 1 day apart) might have no cached results yet.

Generally, the first step is to run an a sync computation job. That fixes most issues.

Precompute Curl

```
curl -XPOST -H'Content-Type: application/json' http://
localhost:8080/reference/analytics/compute_dashboard -d'{}'
| jq '.'
```

If the usual async computation job does not solve the problem, try clearing the analytics cache of any outdated cache entries (e.g., old cache entries pointing to updated models) and then running the computation job as above.

Clear Outdated Cache Entries

```
curl -XPOST -H'Content-Type: application/json' http://
localhost:8080/reference/analytics/
clear_outdated_cache_entries -d'{}' | jq '.'
```

If that fails, clear the entire cache and compute again.

Clear Entire Cache

```
curl -XPOST -H'Content-Type: application/json' http://
localhost:8080/reference/analytics/clear_cache -d'{}' | jq
'.'
```

Models and Scenarios

- No results: Assuming you've already checked to make sure at least some monitored entities have events that should match your model, there's another possible problem. If a model contains a set of features that all match on every event in a given mode (e.g., all emails), then all the features will contribute nothing to the model score (i.e., because they have no rarity).

The first troubleshooting step is to replicate the model using the Explore page. The Explore page allows you to filter by features and with RQL. This, plus some additional RQL: AND entity.filter:("Monitored Entity"=true) will either show events or not. If it does not show any events, that means the model itself is too strict (given the list of monitored entities). Either adding monitored entities or loosening the RQL filter on the model would be good first steps to make the model more usable.

Use Cases and FAQs

Use case 1

User tries to add a Feature to the Feature list on the “Settings” page. User refreshes page, and notices that the Feature is not added to the list.

Initial actions

1. Check developer tool console for Javascript errors.
2. Check that the Feature is being properly added on the backend

3. Perform advanced RQL search to see if the Feature is discoverable via search query.

Diagnosis

Conflicting User roles prevented the user interface from displaying the added Feature. The User was assigned both the *Restricted Reviewer* and *Reviewer Role*. The *Reviewer Role* allowed the User to access the "Feature Configuration" page, but also prevented the User from configuring or seeing Features beyond adding them.

Fix

Access User roles via the "User Management Dashboard." Remove the *Restricted Reviewer Role*, and save the User. Features were then viewable and editable.

Administrator Notes

The *Restricted Reviewer Role* should never be paired with *Administrative Access*.

Use case 2

User tries to add Reviewees in the "Review" dashboard by adding an Entity Filter. The user creates an Entity filter that returns Entities, but still does not see any Entities added in the Reviewees list.

Initial actions

1. Re-add Entity Filter, and browser refresh the "Review" dashboard
2. Check developer tool console for Javascript errors.
3. Check that the Entity Filter is being properly added on the backend.

Diagnosis

The user was not set as *Active* in the "User Management" page

Fix

Access User via the "User Management" page. Set the user status to *Active*. Reviewees then were visible on the "Review" dashboard.

Administrator Notes

If any information is not appearing in the UI as expected, make sure the user is set to *Active*.

1. Not seeing Features or expected app functionality? Check your user roles. Verify that your current selection of user roles grants access to the information you expect to see. Also verify that you do not have the Restricted Reviewer role selected and paired with anything else. Restricted Reviewer is meant to be a stand alone role that severely limits functionality.
   a. **Settings > Profile** to check your user roles.

b. To change your user roles, login to the app as an admin user and navigate to **Settings > User Management** to edit user roles.

2. Check the Application Version. Verify that the version of the software is the same version in which you expect to the see existence of the features you want to see.
   a. **Settings > About Forcepoint Behavioral Analytics**.
3. If a user is not seeing events on the Explore page they would expect to see, check the date range selection. A common mistake for users is that they expect to see a full range of events spanning an entire dataset, but the date range default selection is just the last day of the dataset. Therefore, it is common for a user to make a very detailed search, not change the date range, and then return 0 results. Expanding the date range set is often the solution.
4. If a user isn't able to see events populate on the Behaviors page, there are a few things to check:
   a. Are there entities within the environment that contain the attribute Monitored Entity: True?
      - If not, you'll need to ensure that that's added - this is what drives this page and the ability to show specific actors
   b. What end date, if any, have you selected & what date range shows within your Scenario?
      - If you have selected an end date, navigate back to the Explore page and search for `**feature:("NAME OF FEATURE">0)**` - look at the date range here and compare it to what you're searching for on the Behaviors page. It's likely you're looking outside of this date range.
      - If you have not selected an end date, check to see what date range the Behaviors page is looking at by viewing the date range at the top of the heatmap - same rules apply here. It's likely you're looking outside of the date range.
   c. Do you have an entity filter selected within your configuration that is prohibiting you from seeing certain entities?
      - Example: If I say that I want to see any entity that is in the company's office in Denver and I select my entity filter where **Location=Denver**, but I have no entities that match that model/end date configuration, the page will populate with No Results.
5. If a user has a newly added entitlement and they navigate to the Analytics Dashboard and see the error message: "No Cached Analytics", this means that the dashboard must be recomputed.
   a. It's important to note here, possibly simply for internal purposes, that a client will have a job set up that runs at a specific time of day to recompute this page. Once the page is recomputed, however, the analytics page for that user will show based on their entitlement.
6. Are you seeing the Feature Scoring job error?
   a. Chances are, you may have a feature that's been created with a deleted lexicon. Check to make sure that all of your Lexicon based features are up to date. Also verify that features are not using lexicons that no longer exist. You should then be able to re-run scoring successfully.