Operational Analytics

Overview

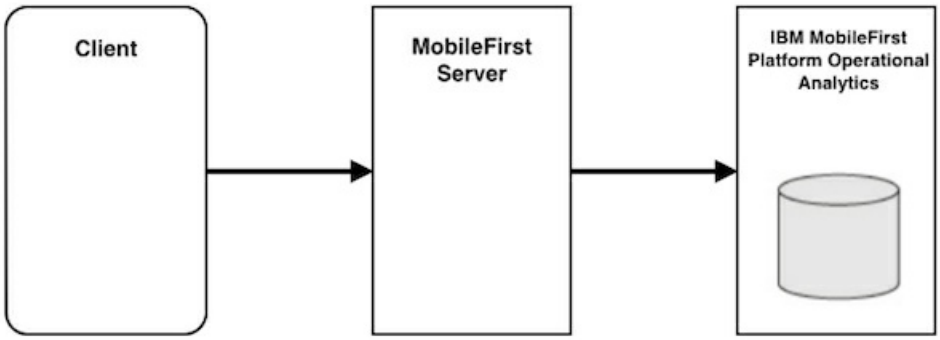
The MobileFirst Platform Operational Analytics Console is a web-based UI that is dedicated to the ongoing monitoring of the deployed applications, adapters, and servers.

Agenda

- Introducing IBM MobileFirst Platform Operational Analytics
- Viewing the Analytics Dashboard - Configurations
- Capturing data
- Capturing data - Network Activities
- Capturing data - Notification Activities
- Capturing data - Server logs
- Capturing data - Client logs
- Analytics logs
- Sending data
- Creating a custom chart
- Alerts
- Migration

Introducing IBM MobileFirst Platform Operational Analytics

MobileFirst Operational Analytics collects data about applications, adapters, devices, logs, and your own custom events to give a high-level view of the client interaction with the IBM MobileFirst Platform Server.



Starting with IBM MobileFirst Platform Foundation V7.0, MobileFirst Operational Analytics is delivered as an EAR file which contains two WAR files: `analytics.war` and `analytics-service.war`. You can deploy the EAR file to the following supported application servers:

- Liberty
- WebSphere Application Server
- Tomcat

In MobileFirst Studio, the two WAR files are installed and available by default in the embedded Liberty server.

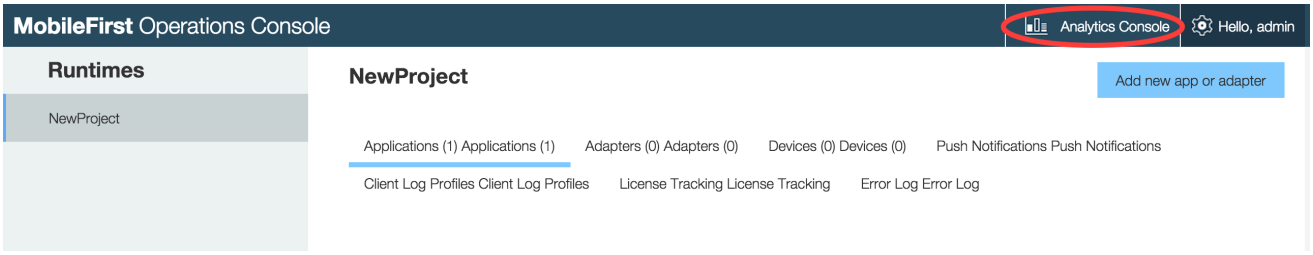
Viewing the Analytics Dashboard - Configurations

The `wl.analytics.url` property must be set to send data to the Analytics server and the `wl.analytics.console.url` must be set to access the Analytics dashboard.

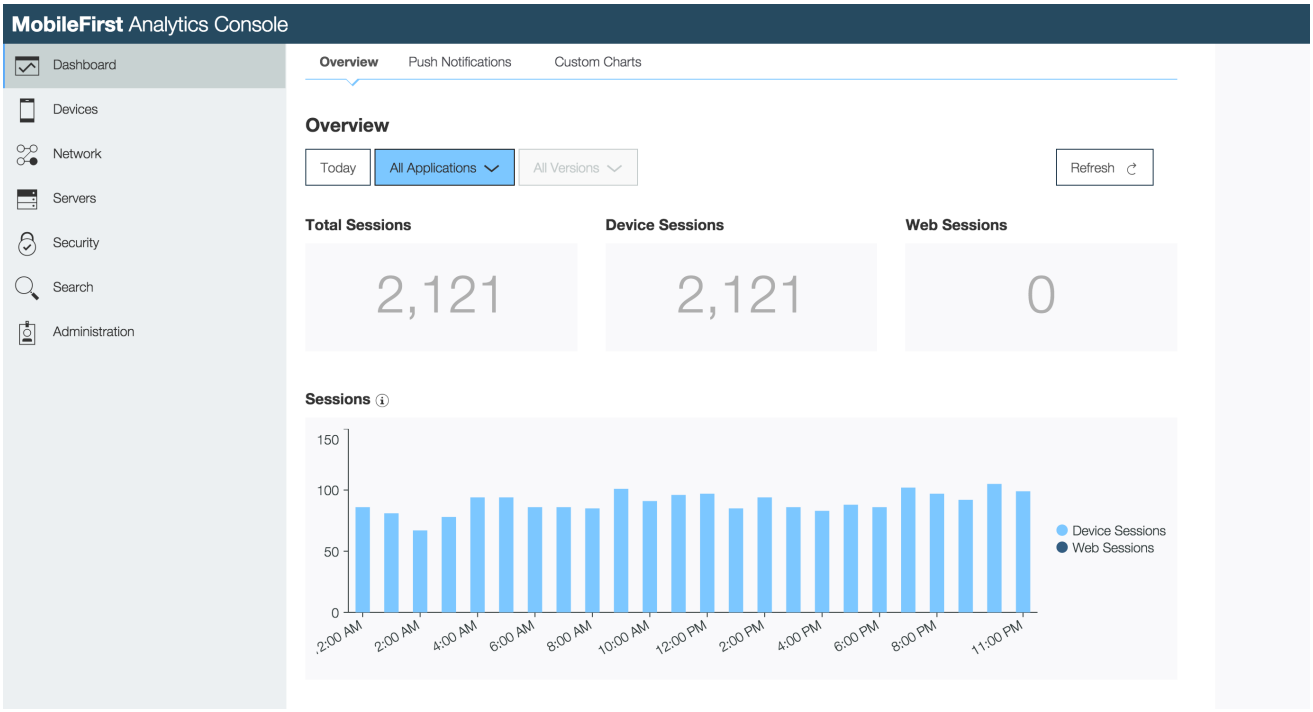
1. You can set these two properties in the `server.xml` file:

   ```
   <jndiEntry jndiName="AppName/wl.analytics.url" value="http://localhost:10080/analytics-service/data"/>
   <jndiEntry jndiName="AppName/wl.analytics.console.url" value="http://localhost:10080/analytics/console"/>
   ```

   After the property is set, the **Analytics Dashboard** link appears in the MobileFirst Operations Console.



2. Click the **Analytics Dashboard** link to open up the dashboard in a new window.



Capturing data

Different types of analytics events are captured by the MobileFirst Operational Analytics server: network activities, notification activities, server logs, and client logs.

Network activities

- Client interacting with the server

Notification activities

- Push notifications

Server Logs

- Server events
- Server stack traces

Client Logs

- Debug logs
- Crashes
- Custom events
- Network latency information

Capturing data - Network activities

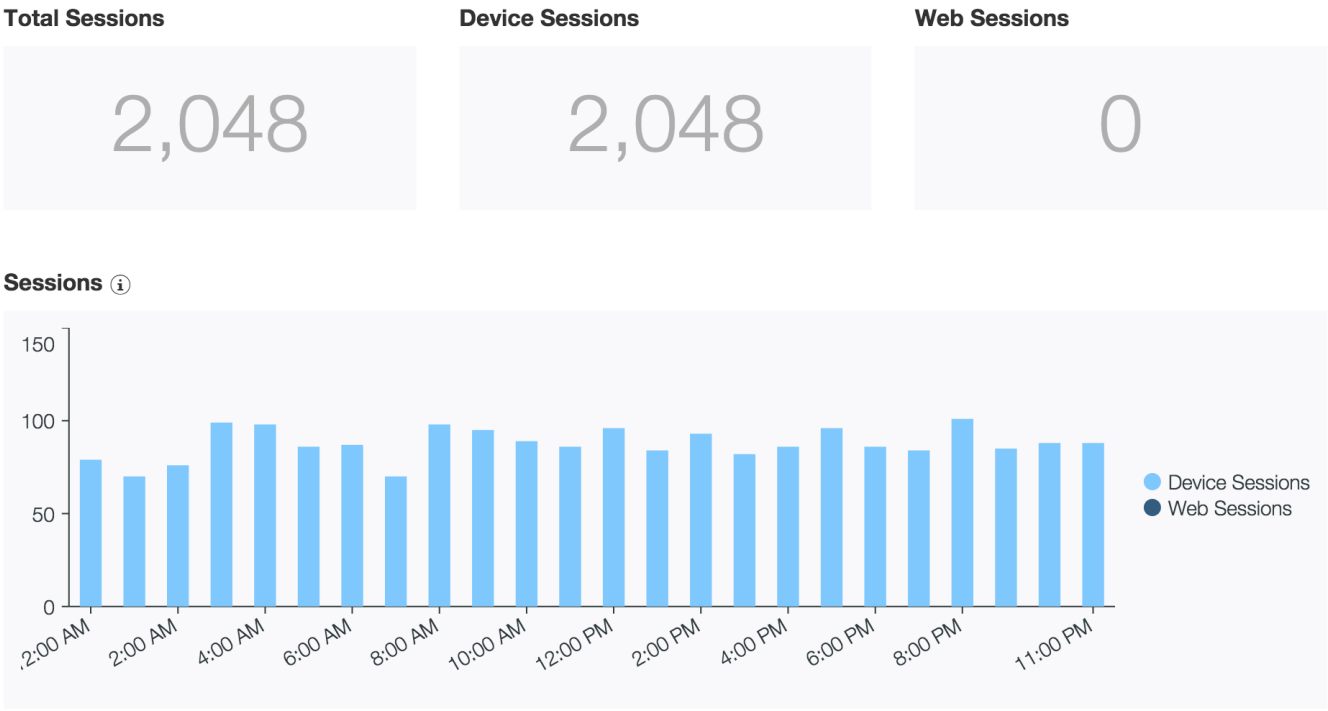
When a network activity occurs, the event is captured automatically and forwarded to the MobileFirst Operational Analytics server.

- The following API call results in a *session hit* that is visualized on MobileFirst Operational Analytics:

```
// a 'session hit' will be recorded upon a successful connection
WL.Client.connect();
```

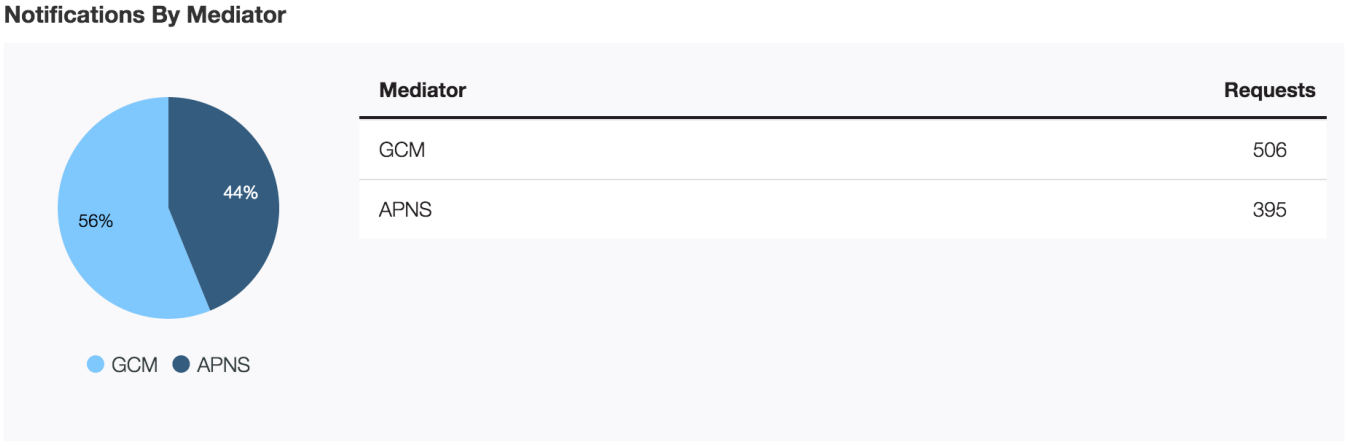
- The following API call results in an *adapter hit* and a *session hit* that are visualized on the MobileFirst Operational Analytics dashboard:

```
// an 'adapter hit' and a 'session hit' will be recorded upon a successful adapter invocation
WLResourceRequest.send();
```



Capturing data - Notification Activities

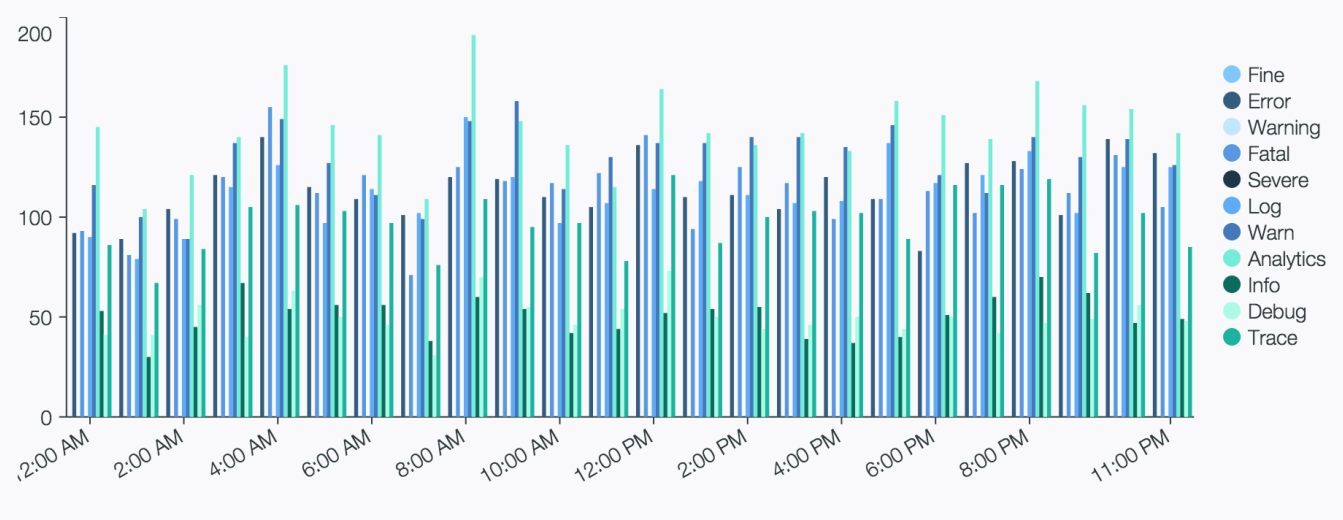
When a push notification occurs, the event is captured automatically and forwarded to the MobileFirst Operational Analytics server.



Capturing data - Server Logs

The log data that is generated by MobileFirst Server is automatically forwarded to the MobileFirst Operational Analytics server, where the data can be searched and downloaded.

Weekly Log Severities



Server Logs

Date	Severe	Warn	Info	Fine	Download
Tuesday, Jun 23, 2015, 12:00 AM	0	116	53	0	
Tuesday, Jun 23, 2015, 1:00 AM	0	100	30	0	
Tuesday, Jun 23, 2015, 2:00 AM	0	89	45	0	
Tuesday, Jun 23, 2015, 3:00 AM	0	137	67	0	
Tuesday, Jun 23, 2015, 4:00 AM	0	149	54	0	
Tuesday, Jun 23, 2015, 5:00 AM	0	127	56	0	
Tuesday, Jun 23, 2015, 6:00 AM	0	111	56	0	

To disable this behavior, set the `wl.analytics.logs.forward` property to `false`.

Capturing data - Client Logs

You can instrument a MobileFirst application with client logs to record client debugging information and events.

You can use the following APIs to create client logs which are then forwarded to the MobileFirst Operational Analytics server, where they can be searched and downloaded.

```
// Set the log level to trace so that all logs are captured
WL.Logger.config({"level": "TRACE"});
// Create a client side log that is persisted locally until it is sent to the server
WL.Logger.trace("Create a client log at the TRACE level.");
WL.Logger.debug("Create a client log at the DEBUG level.");
WL.Logger.info("Create a client log at the INFO level.");
WL.Logger.warn("Create a client log at the WARN level.");
WL.Logger.error("Create a client log at the ERROR level.");
WL.Logger.fatal("Create a client log at the FATAL level.");
```

Analytics Logs

Client-side logs are captured based on the logging level that is set on the client. If you want to create analytics logs that are always captured regardless of the logging level, you can use the `WL.Analytics` API.

```
// Create an analytics log message
WL.Analytics.log("Analytics log message");
// Create a custom activity>
WL.Analytics.log({_activity: "customActivity"});
// Create a custom activity with a log message
WL.Analytics.log({_activity: "customActivity"}, "Analytics log message");
```

Sending data

Logs that are captured by the client-side logging APIs and the `WL.Analytics` APIs are sent to the server automatically upon a successful server connection or a successful adapter call.

```
// Logs sent upon successful connection
WL.Client.connect();
// Logs sent upon successful adapter invocation
WLResourceRequest.send();
```

You can disable this automatic behavior by using the following call:

```
// Disable automatic sending of client and analytics logs
WL.Logger.config({autoSendLogs: false});
```

If you want to send this data more frequently, you can use the following API calls:

```
// Send client debug logs
WL.Logger.send();
// Send analytics logs
WL.Analytics.send();
```

Custom charts

Custom charts are a new feature of MobileFirst Platform Operational Analytics 7.0 and later. By using custom charts, you can take data that is already collected, like device ID, device model, device OS, etc., or log your own custom data and create charts. To understand how to log and send data, see Analytics logs and Sending data.

Chart types

- Bar Graph
- Flow Chart
- Line Graph
- Metric Group
- Pie Chart
- Table

Creating a custom chart

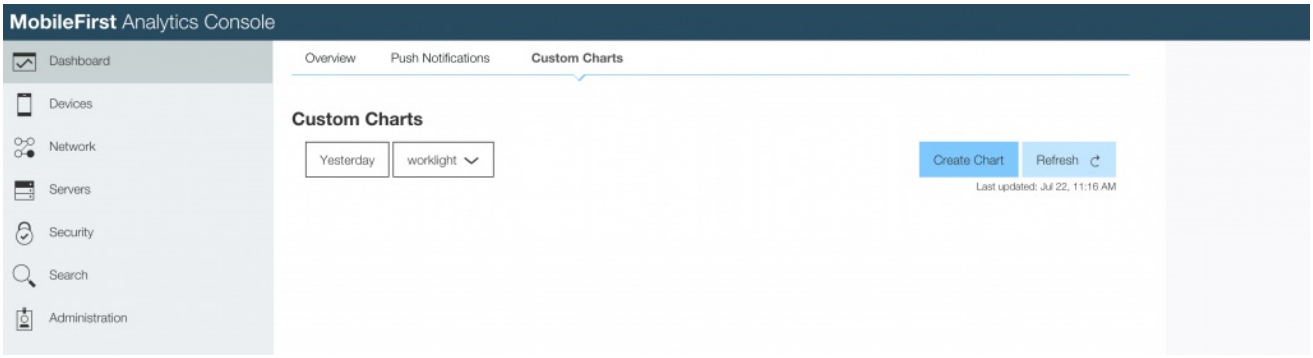
Creating a custom chart is simple. The following example walks you through creating a pie chart, based on the user pressing three buttons.

The messages that are logged to the Operational Analytics server in this example are hard-coded buttons. Those messages look like this:

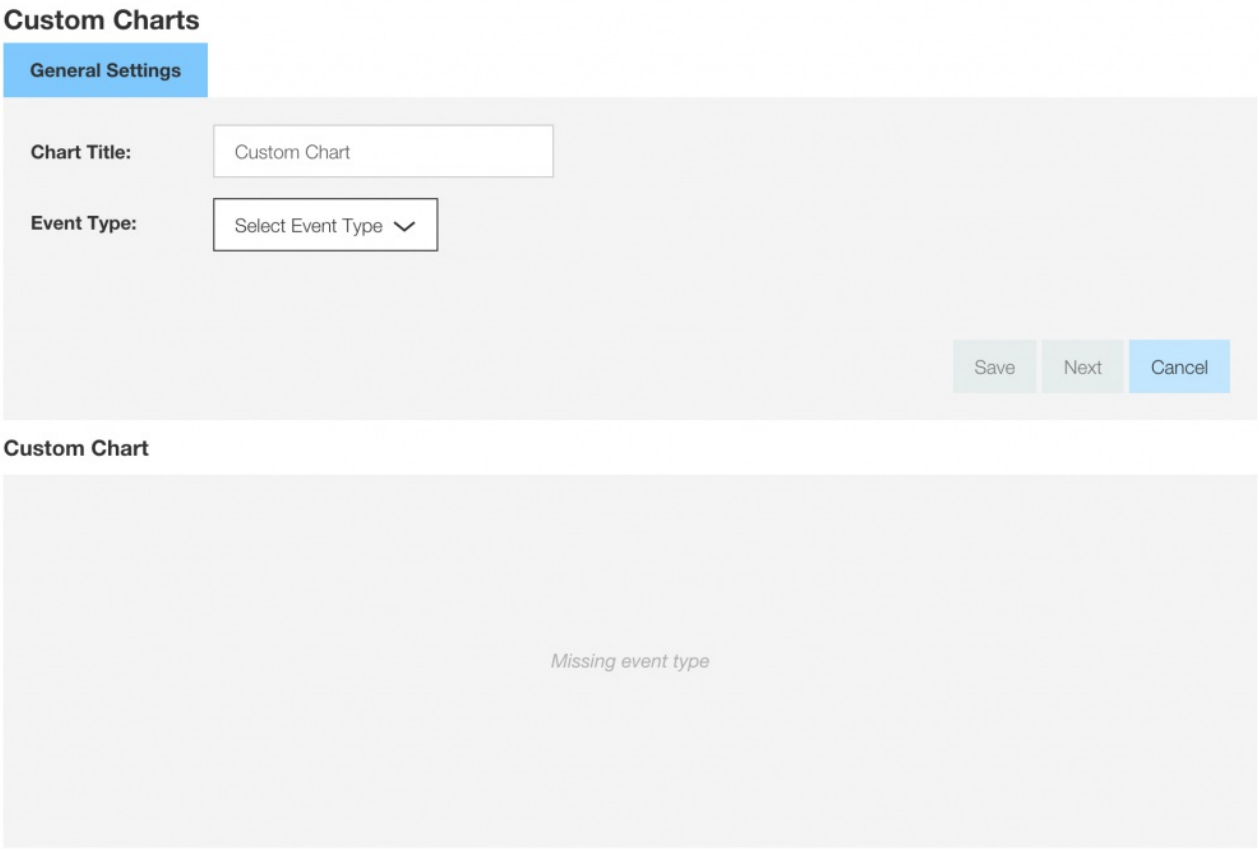
```
WL.Analytics.log({buttonPress: "buttonA"}, "press");
WL.Analytics.log({buttonPress: "buttonB"}, "press");
WL.Analytics.log({buttonPress: "buttonC"}, "press");
```

To create a chart, follow these steps.

1. Go to the **Custom Charts** tab and click **Create Chart**.

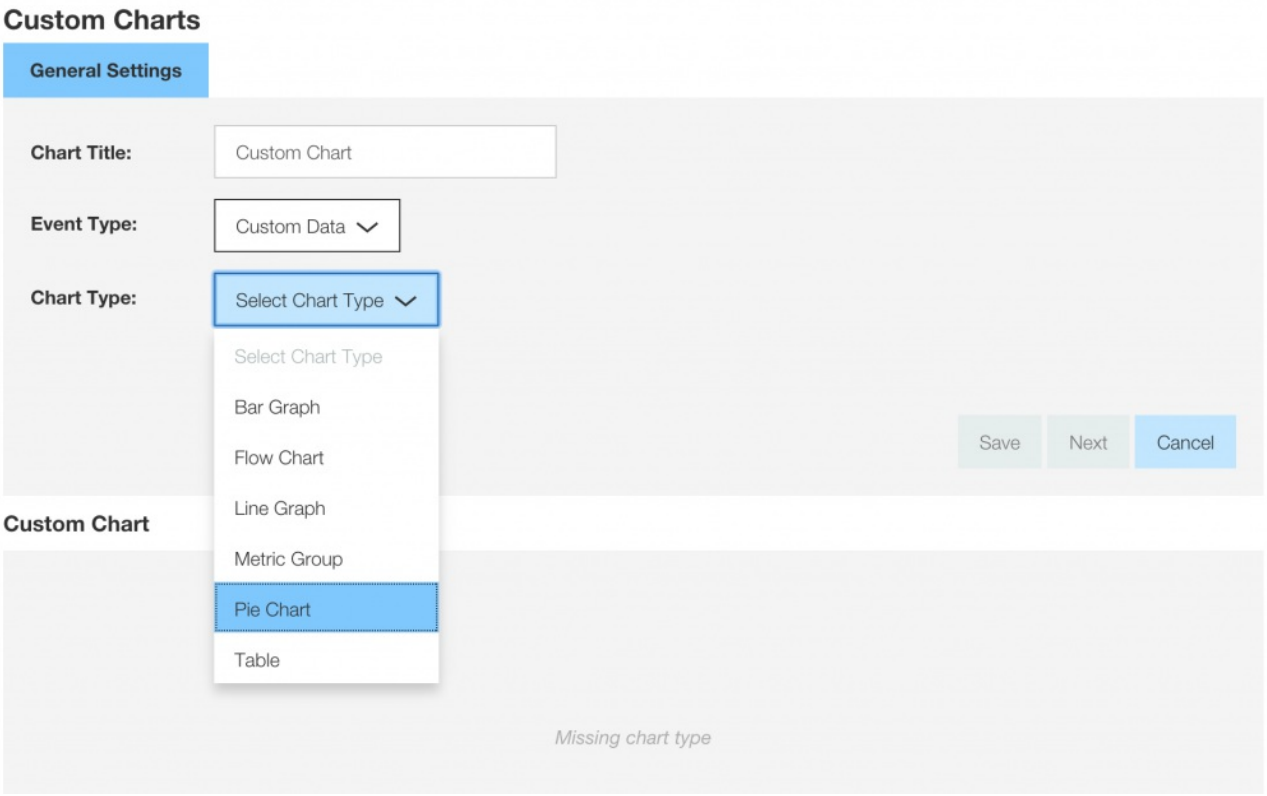


The following will appear.

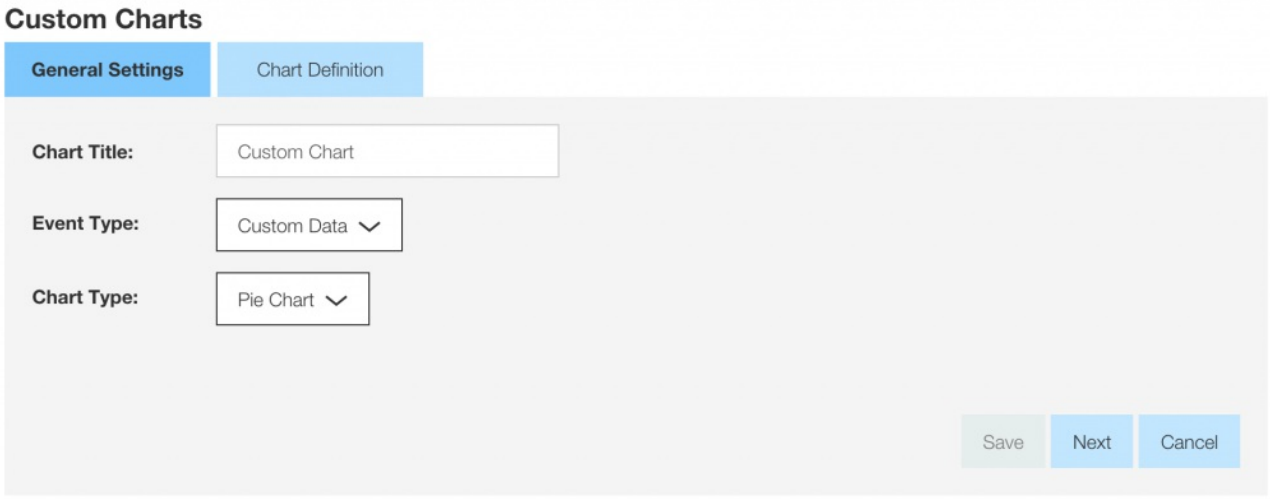


As you fill out your information, more input fields are displayed.

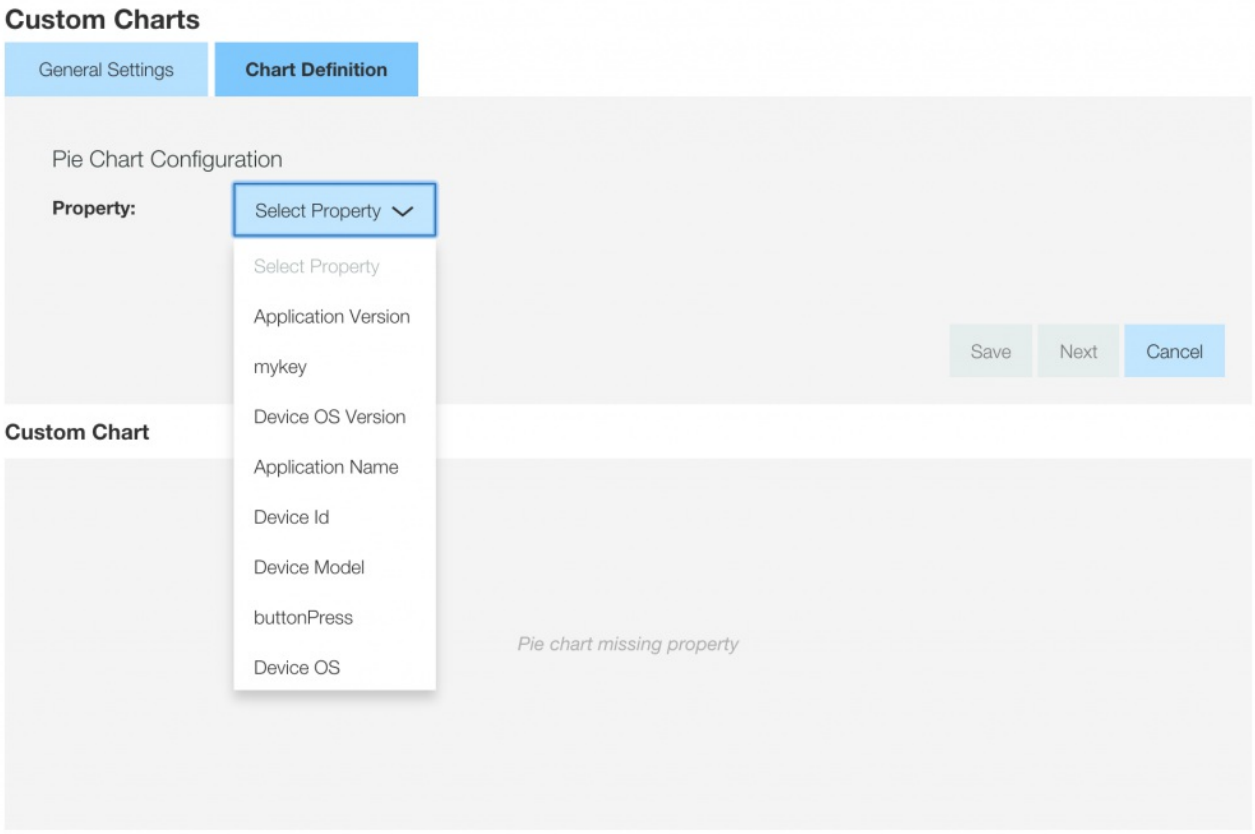
2. Enter a **Chart Title**
3. Select **Custom Data** as the **Event Type**
4. Select a **Chart Type**. This example uses a **Pie Chart**.



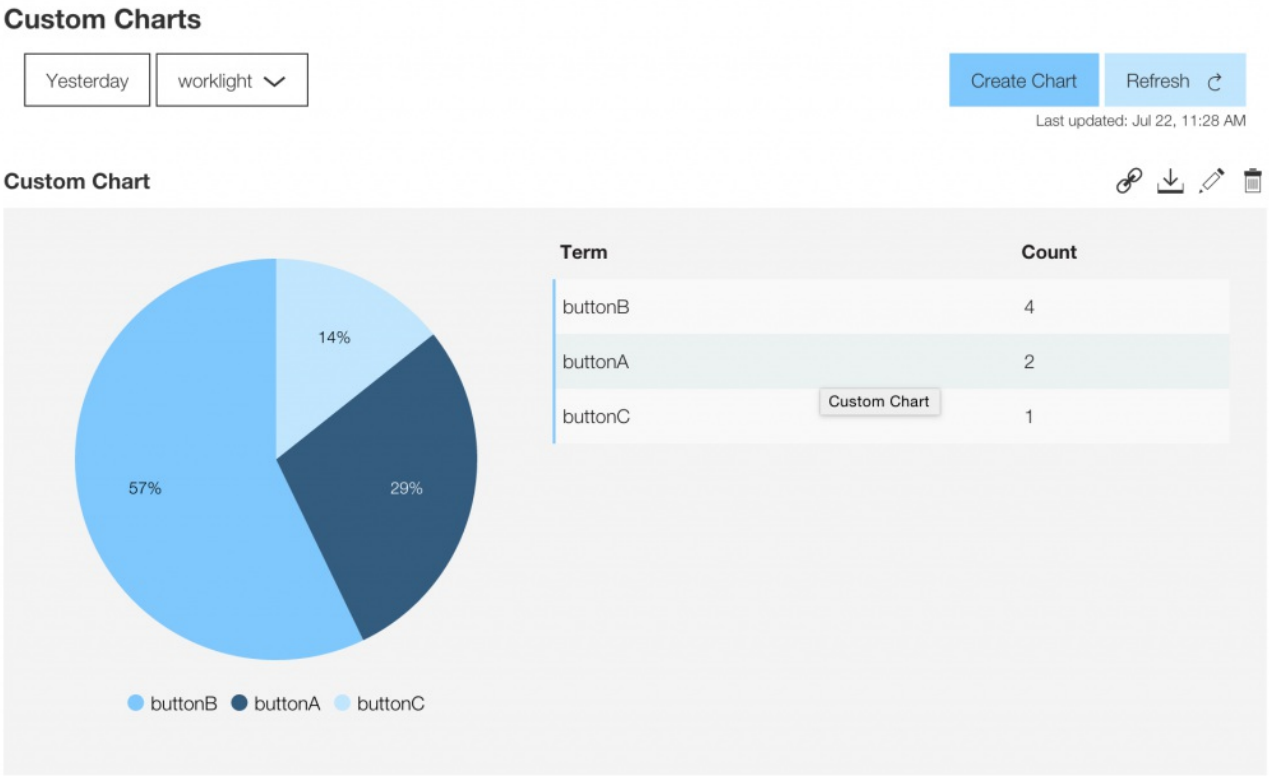
5. Click on the **Chart Definition** tab.



6. Select a **Property**. This example uses `buttonPress`.



7. Click **Save**. The chart is saved under the **Custom Charts** tab in the main dashboard.



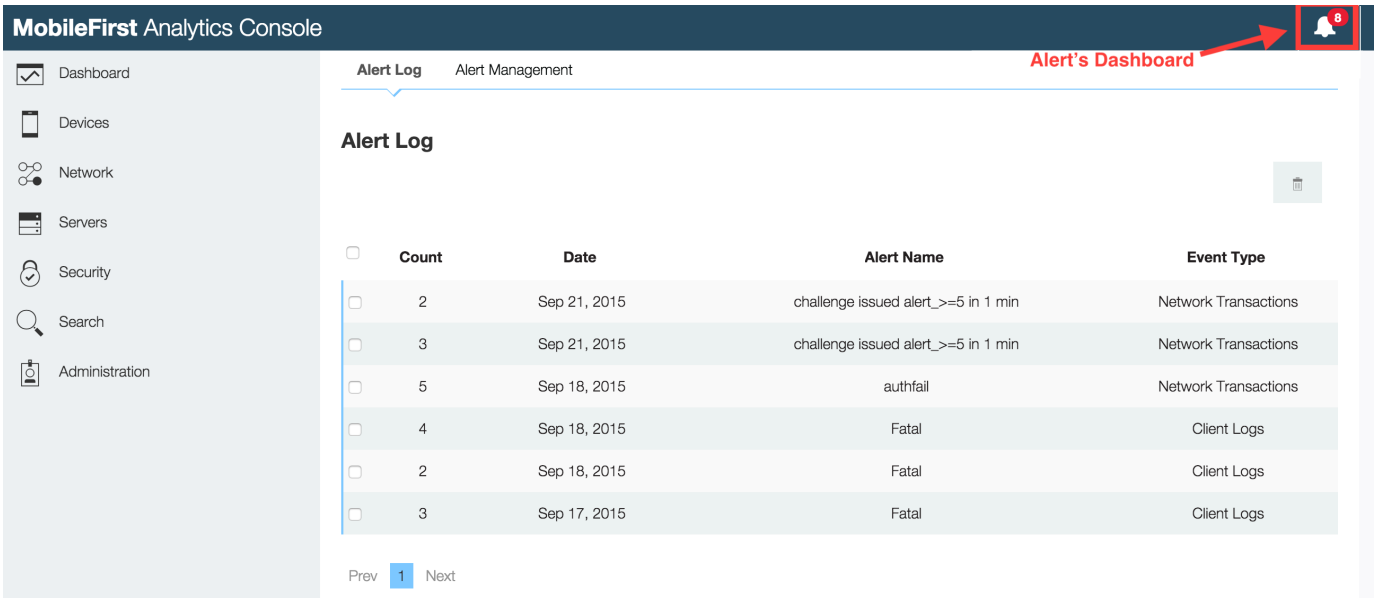
Alerts

If you are using the latest interim fix of MobileFirst, you can set thresholds in alert definitions in the IBM MobileFirst™ Platform Operational Analytics Console to better monitor your activities.

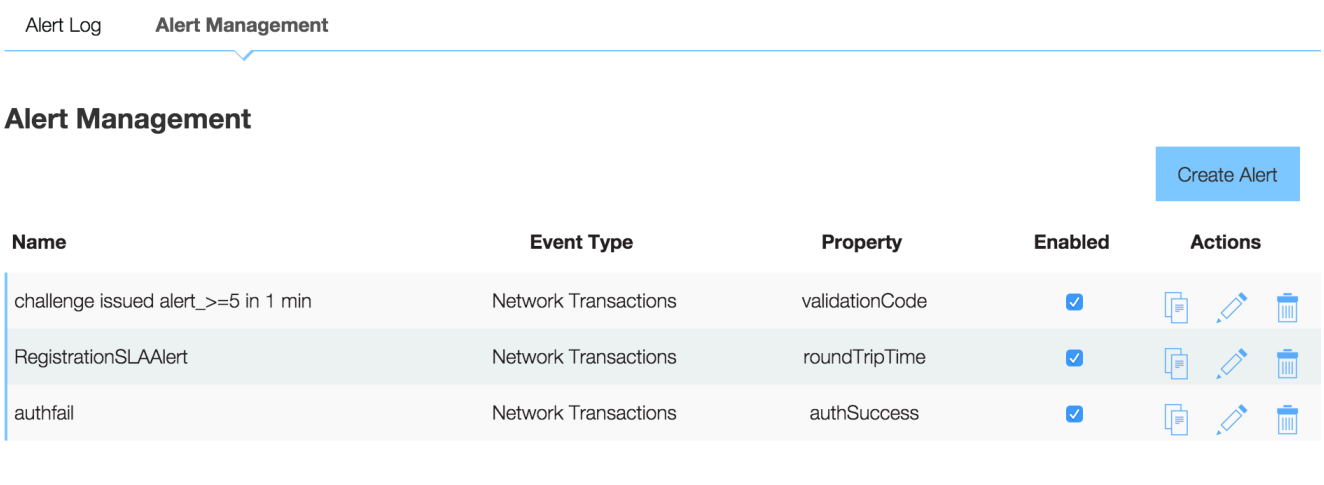
You can configure thresholds, which if exceeded, trigger alerts to notify the MobileFirst Operational Analytics Console monitor. The triggered alerts can be visualized on the console, or the alerts can be handled by a custom webhook. This feature provides a proactive means of detecting client log errors, server log errors, extended periods of network latency, and authentication failures. Reactive thresholds and alerts keep you from having to sift through your data and set thresholds at a wide spectrum of granularity.

Navigating Alerts

To navigate to the Alert's dashboard there is a button with a bell in the top right corner as indicated in the picture below. The button may have a number associated, like in the image below. This number represents the number of alerts that have gone off since the last time last time a user has marked that they were notified of the alerts.



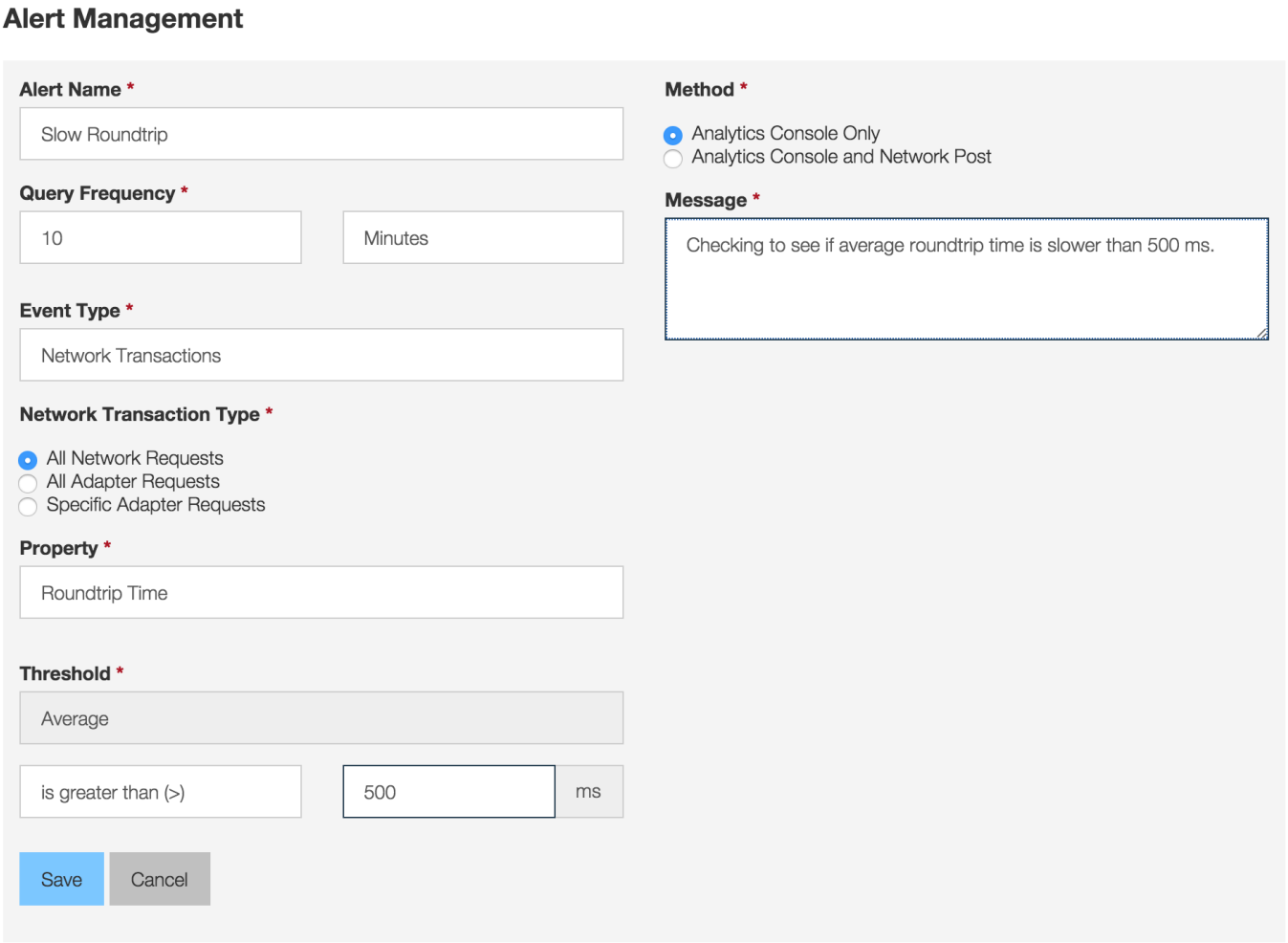
The initial page to the Alert's dashboard is the Alert's log page. In the Alert's log page you can see previous alerts that you were alerted about. If an alert has not been marked as viewed, then you will see a red circle over the alert count. To mark this as viewed, just press on the red circle.



If you navigate to the 'Alert Management' tab, you can see the previous alerts that have been created. Here you can enable the alerts, copy the alerts for a starting point when creating a new alert, edit an existing alert, delete an existing alert, or create a new alert.

Creating an Alert

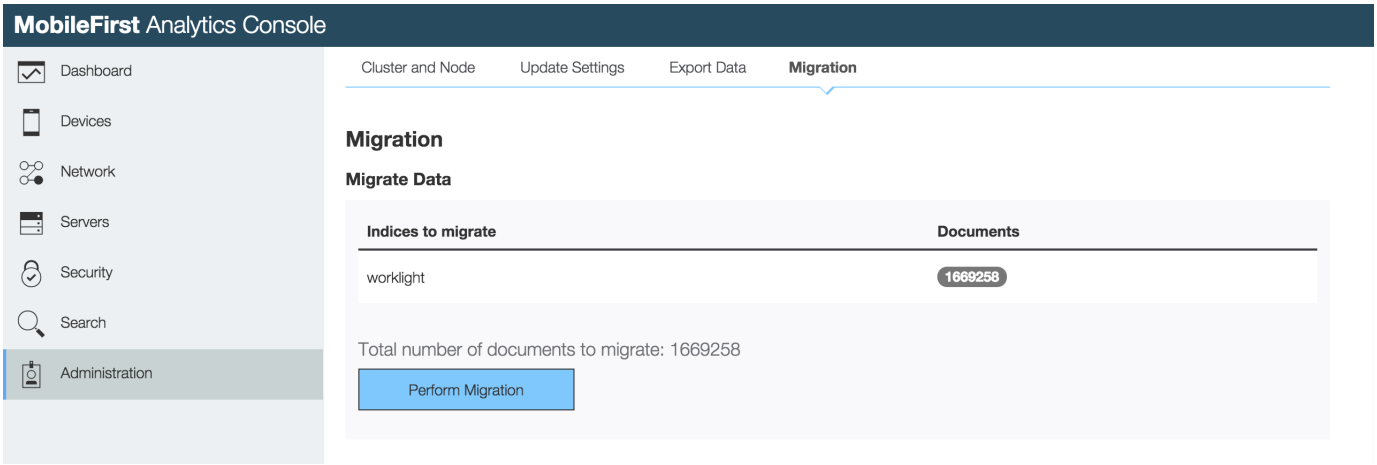
To create an alert, press the 'Create Alert' button under the 'Alert Management' tab in the Alert's dashboard. When creating an alert, the alert's information will populate as you are filling in information because not every alert will take the same values. Below you can see an alert with information already filled in.



Migration

MobileFirst Platform Foundation Analytics now has a simple migration tool for users who are on an earlier version of Operational Analytics than MobileFirst Platform Foundation 7.1.0. This feature is needed because some of the search mappings in previous versions of IBM MobileFirst Platform Foundation that are necessary to populate the Analytics console have been changed.

To run this search, go to the **Administration** tab on the left side of the console. After the administration console loads, click the **Migration** tab on the top tab bar. You can then see how many documents you have to migrate and the progress of the migration. You can perform a migration by pressing the **Perform Migration** button. After the migration process, all your data is migrated to IBM MobileFirst Platform Foundation 7.1.0 with all the correct mappings. You can see an example of the migration tab below.



Last modified on

Privacy
(http://www.ibm.com/privacy/us/en/)
Terms of use
(file:////home/travis/build/MFPSamples/DevCenter/_site/terms-of-use/)
Third party notice
(file:////home/travis/build/MFPSamples/DevCenter/_site/third-party-notice/)

(https://twitter.com/ibmmobiledev)
YouTube
(https://www.youtube.com/channel/UC1-C71A4eaKzn-ci2Qusu97Q)
GitHub
(https://github.com/MobileFirst-Platform-Developer-Center)

(https://github.com/MobileFirst-Platform-Developer-Center/DevCenter/issues/new)
Contribute
(https://github.com/MobileFirst-Platform-Developer-Center/DevCenter/blob/master/contributing.m
Report abuse
(https://www.ibm.com/developerworks/commu