MobileFirst Foundation Operational Analytics

Overview

IBM MobileFirst Foundation Operational Analytics collects data from app-to-server activities, client logs, client crashes, and server-side logs from the MobileFirst Runtime Server and client devices. The collected data then provides a rich view into both the mobile landscape and server infrastructure. Included are: default reports of user retention, crash reports, device type and operating system breakdowns, custom data and custom charts, network usage, push notification results, in-app behavior, debug log collection, and beyond.

MobileFirst Server comes pre-instrumented with network infrastructure reporting. When both the client and server are reporting network usage, the data is aggregated so you can attribute poor performance to the network, the server, or the back-end systems. In addition, you can control which logger data is accessed and used by analytics by defining filters both on the client-side and on the MobileFirst Analytics Server. You choose the verbosity and data retention policy of the reported events, set conditional alerts, build custom charts and engage with new data.

Platform support

MobileFirst Operational Analytics supports:

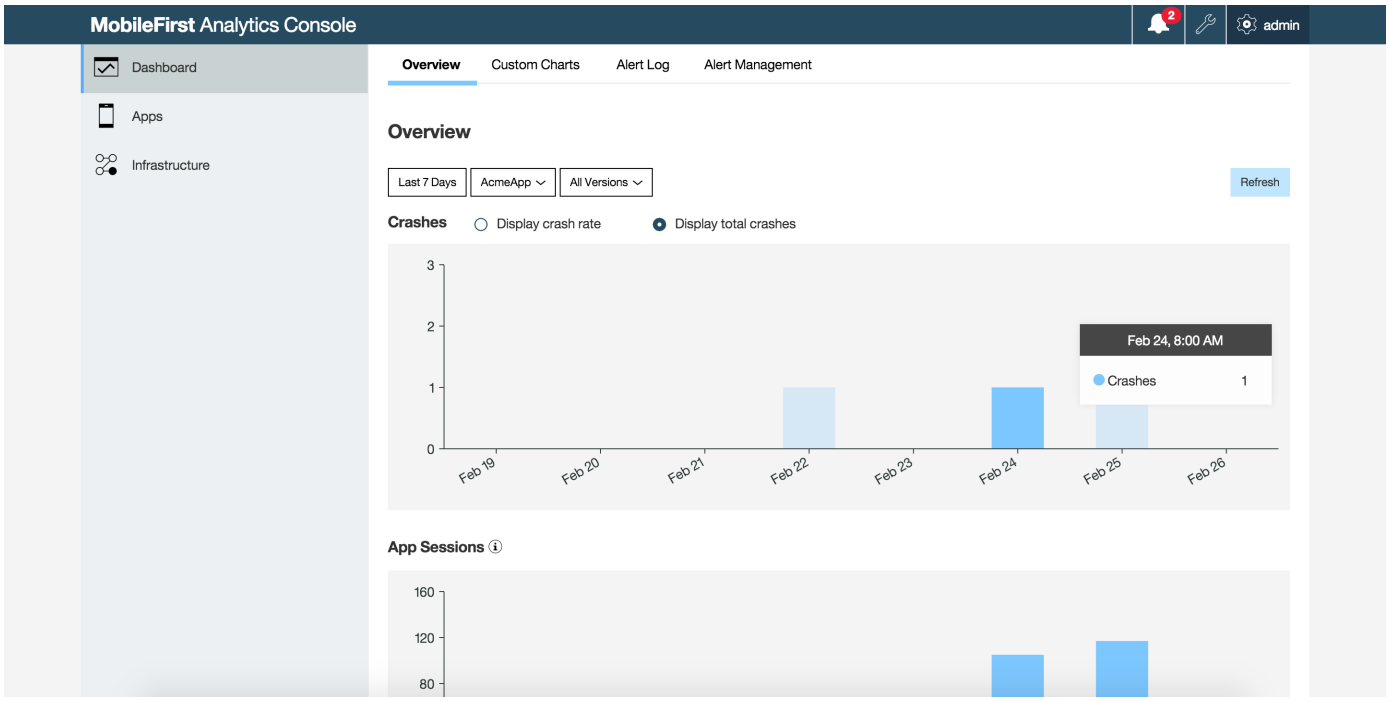
- Native iOS and Android clients
- Cordova applications (iOS, Android)
- Web applications
- Support is **not available** for the Windows 8.1 Universal or Windows 10 UWP native or Cordova platforms

Jump to

Operational Analytics Console

You can open the Analytics Console from the MobileFirst Operations Console by clicking on **Analytics Console** in the top-right navigation.
If you are using the Mobile Foundation Bluemix service (../bluemix/using-mobile-foundation) and require Analytics integration, see this blog post (file:////home/travis/build/MFPSamples/DevCenter/_site/blog/2016/07/11/analytics-bm-service/).

After navigating to the Analytics console, you see a dashboard like the following one (but with empty data).



Dashboard

In Dashboard you can review collected analytics data related to: application crashes, application sessions and server processing time. Additionally you can create custom charts as well as manage alerts.

Apps

In Apps you can review in-depth analytics data related to: usage and devices (such as total device and app sessions, active users, app usage, new devices, model usage and operating system), as well as crash-related data and search through client logs and for specific devices.

Infrastructure

In Infrastructure you can review analytics data related to: session processing time, average request size, server requests, network requests, adapters response time, procedure response time and size and adapters usage, as well as push notifications data such as notification requests and per mediator. You can also search through server logs.

> Learn more in the Analytics Workflows (workflows/) tutorial.

Operational Analytics Features

App Analytics

You can view App Session charts and App Usage charts to find out which app is being used most by your users.

Built-in Analytics

When you use the MobileFirst client SDK together with the MobileFirst Server, analytics data automatically gets collected for any request that your app makes to the MobileFirst Server. Basic device metadata gets collected and reported to the MobileFirst Analytics Server.

Custom Analytics

You can have your app send custom data and create custom reports on your custom data.

> Learn how to send custom analytics in the Analytics API (analytics-api/) tutorial.

Custom Charts

Custom charts allow you to visualize the collected analytics data in your analytics data store as charts that are not available by default in the MobileFirst Operational Analytics Console. This visualization feature is a powerful way to analyze business-critical data.

> Learn how to create custom charts in the Creating Custom Charts (custom-charts/) tutorial.

Manage Alerts

Alerts provide a proactive means to monitor the health of your mobile apps without having to check the MobileFirst Analytics Console regularly.

You can configure thresholds which, if exceeded, trigger alerts to notify administrators. You can visualize the triggered alerts on the console or handle them by using a custom webhook. A custom webhook allows you to control who is notified when an alert is triggered, and how.

> Learn how to manage alerts in the Manage Alerts (alerts/) tutorial.

Monitor App Crashes

App crashes are visualized on the Analytics Console, where you can quickly view crashes and act on them accordingly. Crash logs are collected on the device by default. When crash logs are sent to the analytics server, they automatically populate the crash charts.

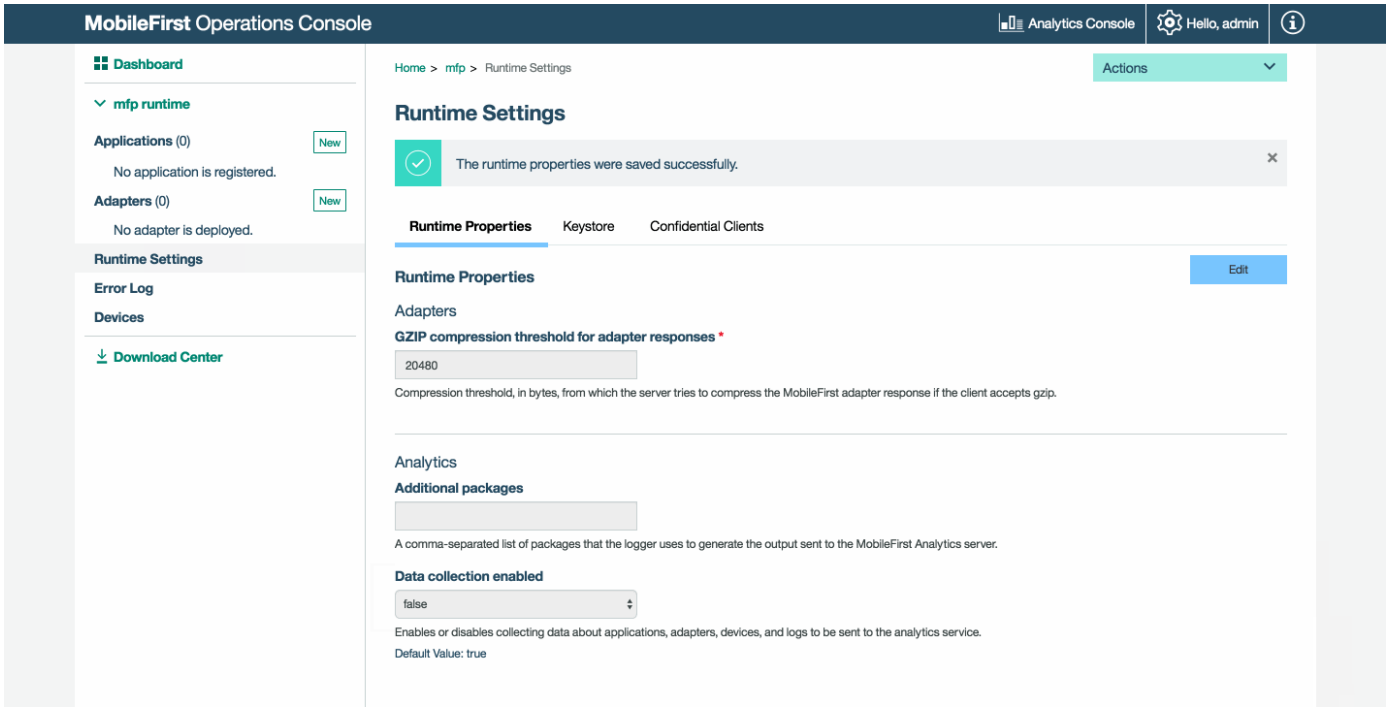
Monitor Network Data

The MobileFirst Operational Analytics Console monitors network data when it is sent to the analytics server and allows the user to query this information in different ways.

Enable/Disable Analytics Support

The collection of data for analysis by the Analytics server is enabled by default. You can disable it, for example to save processing time.

1. In the navigation sidebar, click on **Runtime settings**. To avoid inadvertent changes, runtime properties are displayed in read-only mode.
2. To make the settings editable, click the **Edit** button. If you logged in with a role other than *administrator* or *deployer*, the Edit button is not visible because you are not allowed to modify runtime properties.
3. From the **Data collection enabled** drop-down menu, select **false** to disable data collection.
4. Click Save.
5. Click the Read Only button to lock the properties again.



Role-based Access Control

Content in the MobileFirst Analytics Console is restricted by predefined security roles.
The MobileFirst Analytics Console shows different content based on the security role of the logged-in user. The following table shows the security role and the access that is granted to it in the MobileFirst Analytics Console.

Role	Role name	Viewing Access	Editing Access
Administrator	analytics_administrator	Everything.	Everything.
Infrastructure	analytics_infrastructure	Everything.	Everything.
Developer	analytics_developer	Everything except for the Administration pages.	Everything.
Support	analytics_support	Everything except for the Administration pages.	Everything.
Business	analytics_business	Everything except for the Administration and Infrastructure pages.	Everything.

> For information on setting up roles, see the Configuring user authentication for MobileFirst Server administration (http://www.ibm.com/support/knowledgecenter/en/SSHS8R_8.0.0/com.ibm.worklight.installconfig.doc/install_config/c_configuration_of_the_wl_admin.html?view=kc) user documentation topic.

Elasticsearch

Behind the scenes, running search queries and storing data for Operational Analytics is **Elasticsearch 1.5x**.
Elasticsearch is a real-time distributed search and analytics engine that provides the ability to explore data at speed and at a scale. Elasticsearch is used for full-text search, structured search.

Elasticsearch is used for storing all mobile and server data in JSON format in the MobileFirst Operational Analytics server in Elasticsearch instances.
The Elasticsearch instances are queried in real-time to populate the MobileFirst Operational Analytics Console.

MobileFirst Operational Analytics does not hide any Elasticsearch functionality. If knowledge about how to take full benefit of Elasticsearch, debug Elasticsearch, or optimize Elasticsearch instances is present, Operational Analytics does not prevent using it.

If you have interest in any Elasticsearch functionality besides what is predefined in MobileFirst Operational Analytics, you can read more about it in the Elasticsearch documentation.

> Read more in the Elasticsearch documentation (https://www.elastic.co/guide/en/elasticsearch/reference/1.5/index.html).

Elasticsearch properties

Elasticsearch properties are available through JNDI variables or environment entries.
One of the more useful JNDI properties to get started with viewing the Elasticsearch data is:

```
<jndiEntry jndiName="analytics/http.enabled" value="true"/>
```

This JNDI property allows you to view your Operational Analytics raw data in JSON format and to access your Elasticsearch instance through the port that is defined by Elasticsearch. The default port is 9500.

> **Note**: This setting is not secure and should not be enabled on a production environment.

Viewing data
You can view all your data by visiting the tenant's search REST endpoint.
Being able to access an Elasticsearch instance provides the ability to run custom queries and view more detailed information about the Elasticsearch cluster.

```
http://localhost:9500/*/_search
```

View cluster health

```
http://localhost:9500/_cluster/health
```

View information on current nodes

```
http://localhost:9500/_nodes
```

View the current mappings

```
http://localhost:9500/*/_mapping
```

> Elasticsearch exposes many more REST endpoints. To learn more, visit the Elasticsearch documentation.

Related Blog Posts

- More on Instrumenting Custom Analytics (file:////home/travis/build/MFPSamples/DevCenter/_site/blog/2016/01/22/howto-custom-in-app-behavior-analytics/)
- More on Instrumenting Webhooks (file:////home/travis/build/MFPSamples/DevCenter/_site/blog/2015/10/19/using-mfp-adapters-endpoint-analytics-alerts-webhooks/)

Tutorials to Follow Next