

MobileFirst Analytics Server installation guide

Overview

MobileFirst Analytics Server is implemented and shipped as a set of two Java EE standard web application archive (WAR) files, or one enterprise application archive (EAR) file. Therefore, it can be installed in one of the following supported application servers: WebSphere® Application Server, WebSphere Application Server Liberty, or Apache Tomcat (WAR files only).

MobileFirst Analytics Server uses an embedded Elasticsearch library for the data store and cluster management. Because it intends to be a highly performant in-memory search and query engine, requiring fast disk I/O, you must follow some production system requirements. In general, you are most likely to run out of memory and disk (or discover that disk I/O is your performance bottleneck) before CPU becomes a problem. In a clustered environment, you want a fast, reliable, co-located cluster of nodes.

Jump to

- [System requirements](#)
- [Capacity considerations](#)
- [Installing MobileFirst Analytics on WebSphere Application Server Liberty](#)
- [Installing MobileFirst Analytics on Tomcat](#)
- [Installing MobileFirst Analytics on WebSphere Application Server](#)
- [Installing MobileFirst Analytics with Ant tasks](#)
- [Installing MobileFirst Analytics Server on servers running previous versions](#)

System requirements

Operating systems

- CentOS/RHEL 6.x/7.x
- Oracle Enterprise Linux 6/7 with RHEL Kernel only
- Ubuntu 12.04/14.04
- SLES 11/12
- OpenSuSE 13.2
- Windows Server 2012/R2
- Debian 7

JVM

- Oracle JVM 1.7u55+
- Oracle JVM 1.8u20+
- IcedTea OpenJDK 1.7.0.55+

Hardware

- RAM: More RAM is better, but no more than 64 GB per node. 32 GB and 16 GB are also acceptable. Less than 8 GB requires many small nodes in the cluster, and 64 GB is wasteful and problematic due to the way Java uses memory for pointers.
- Disk: Use SSDs when possible, or fast spinning traditional disks in RAID 0 configuration if SSDs are not possible.
- CPU: CPU tends not to be the performance bottleneck. Use systems with 2 to 8 cores.
- Network: When you cross into the need to scale out horizontally, you need a fast, reliable, data center with 1 GbE to 10 GbE supported speeds.

Hardware configuration

- Give your JVM half of the available RAM, but do not cross 32 GB
 - Setting the **ES_HEAP_SIZE** environment variable to 32g.
 - Setting the JVM flags by using -Xmx32g -Xms32g.
- Turn off disk swap. Allowing the operating system to swap heap on and off disk significantly degrades performance.
 - Temporarily: `sudo swapoff -a`
 - Permanently: Edit **/etc/fstab** according to the operating system documentation.
 - If neither option is possible, set the Elasticsearch option **bootstrap.mlockall: true** (this value is the default in the embedded Elasticsearch instance).
- Increase the allowed open file descriptors.
 - Linux typically limits a per-process number of open file descriptors to a small 1024.
 - Consult your operating system documentation for how to permanently increase this value to something much larger, like 64,000.
- Elasticsearch also uses a mix of NioFS and MMapFS for the various files. Increase the maximum map count so plenty of virtual memory is available for mmapmed files.
 - Temporarily: `sysctl -w vm.max_map_count=262144`
 - Permanently: Modify the **vm.max_map_count** setting in your **/etc/sysctl.conf**.
- If you use BSDs and Linux, ensure that your operating system I/O scheduler is set to **deadline** or **noop**, not **cfq**.

Capacity considerations

Capacity is the single-most common question. How much RAM do you need? How much disk space? How many nodes? The answer is always: it depends.

IBM MobileFirst Analytics gives you the opportunity to collect many heterogeneous event types, including raw client SDK debug logs, server-reported network events, custom data, and much more. It is a big data system with big data system requirements.

The type and amount of data that you choose to collect, and how long you choose to keep it, has a dramatic impact on your storage requirements and overall performance. As an example, consider the following questions.

- Are raw debug client logs useful after a month?
- Are you using the **Alerts** feature in MobileFirst Analytics? If so, are you querying on events that occurred in the last few minutes or over a longer range?
- Are you using custom charts? If so, are you creating these charts for built-in data or custom instrumented key/value pairs? How long do you keep the data?

The built-in charts on the MobileFirst Analytics Console are rendered by querying data that the MobileFirst Analytics Server already summarized and optimized specifically for the fastest possible console user experience. Because it is pre-summarized and optimized for the built-in charts, it is not suitable for use in alerts or custom charts where the console user defines the queries.

When you query raw documents, apply filters, perform aggregations, and ask the underlying query engine to calculate averages and percentages, the query performance necessarily suffers. It is this use case that requires careful capacity considerations. After your query performance suffers, it is time to decide whether you really must keep old data for real-time console visibility or purge it from the MobileFirst Analytics Server. Is real-time console visibility truly useful for data from four months ago?

Indices, Shards, and Nodes

The underlying data store is Elasticsearch. You must know a bit about indices, shards and nodes, and how the configuration affects performance. Roughly, you can think of an index as a logical unit of data. An index is mapped one-to-many to shards where the configuration key is shards. The MobileFirst Analytics Server creates a separate index per document type. If your configuration does not discard any document types, you have a number of indices that are created that is equivalent to the number of document types that are offered by the MobileFirst Analytics Server.

If you configure the shards to 1, each index only ever has one primary shard to which data is written. If you set shards to 10, each index can balance to 10 shards. However, more shards have a performance cost when you have only one node. That one node is now balancing each index to 10 shards on the same physical disk. Only set shards to 10 if you plan to immediately (or nearly immediately) scale up to 10 physical nodes in the cluster.

The same principle applies to **replicas**. Only set **replicas** to something greater than 0 if you intend to immediately (or nearly immediately) scale up to the number of nodes to match the math.

For example, if you set **shards** to 4 and **replicas** to 2, you can scale to 8 nodes, which is $4 * 2$.

Installing MobileFirst Analytics on WebSphere Application Server Liberty

Ensure that you already have the MobileFirst Analytics EAR file. For more information on the installation artifacts, see Installing MobileFirst Server to an application server (`../appserver`). The **analytics.ear** *file is found in the analytics* folder. For more information about how to download and install WebSphere Application Server Liberty, see the About WebSphere Liberty (<https://developer.ibm.com/wasdev/websphere-liberty/>) article on IBM® developerWorks®.

1. Create a server by running the following command in your `./wlp/bin` folder.

```
./server create <serverName>
```

2. Install the following features by running the following command in your `./bin` folder.

```
./featureManager install jsp-2.2 ssl-1.0 appSecurity-1.0 localConnector-1.0
```

3. Add the **analytics.ear** file to the `./usr/servers/apps` folder of your Liberty Server.
4. Replace the contents of the `<featureManager>` tag of the `./usr/servers/server.xml` file with the following content:

```
<featureManager>
  <feature>jsp-2.2</feature>
  <feature>ssl-1.0</feature>
  <feature>appSecurity-1.0</feature>
  <feature>localConnector-1.0</feature>
</featureManager>
```

5. Configure **analytics.ear** as an application with role-based security in the **server.xml** file. The following example creates a basic hardcoded user registry, and assigns a user to each of the different analytics roles.

```
<application location="analytics.ear" name="analytics-ear" type="ear">
  <application-bnd>
    <security-role name="analytics_administrator">
      <user name="admin"/>
    </security-role>
    <security-role name="analytics_infrastructure">
      <user name="infrastructure"/>
    </security-role>
    <security-role name="analytics_support">
      <user name="support"/>
    </security-role>
    <security-role name="analytics_developer">
      <user name="developer"/>
    </security-role>
    <security-role name="analytics_business">
      <user name="business"/>
    </security-role>
  </application-bnd>
</application>

<basicRegistry id="worklight" realm="worklightRealm">
  <user name="business" password="demo"/>
  <user name="developer" password="demo"/>
  <user name="support" password="demo"/>
  <user name="infrastructure" password="demo"/>
  <user name="admin" password="admin"/>
</basicRegistry>
```

For more information about how to configure other user registry types, such as LDAP, see the [Configuring a user registry for Liberty](http://ibm.biz/knowctr#SSAW57_8.5.5/com.ibm.websphere.wlp.nd.iseries.doc/ae/twlp_sec_registries.html) (http://ibm.biz/knowctr#SSAW57_8.5.5/com.ibm.websphere.wlp.nd.iseries.doc/ae/twlp_sec_registries.html) topic in the WebSphere Application Server product documentation.

6. Start the Liberty Server by running the following command inside your **bin** folder

```
./server start <serverName>
```

7. Go to the MobileFirst Analytics Console.

```
http://localhost:9080/analytics/console
```

For more information about administering WebSphere Application Server Liberty, see the [Administering Liberty from the command line](http://ibm.biz/knowctr#SSAW57_8.5.5/com.ibm.websphere.wlp.nd.multiplatform.doc/ae/twlp_admin_script.html) (http://ibm.biz/knowctr#SSAW57_8.5.5/com.ibm.websphere.wlp.nd.multiplatform.doc/ae/twlp_admin_script.html) topic in the WebSphere Application Server product documentation.

Installing MobileFirst Analytics on Tomcat

Ensure that you already have the MobileFirst Analytics WAR files. For more information on the installation artifacts, see [Installing MobileFirst Server to an application server](#) ([../appserver](#)). The **analytics-ui.war** and **analytics-service.war** files are found in the **\analytics** folder. For more information about how to download and install Tomcat, see [Apache Tomcat](http://tomcat.apache.org/) (<http://tomcat.apache.org/>). Ensure that you download the version that supports Java 7 or higher. For more information about which version of Tomcat supports Java 7, see [Apache Tomcat Versions](http://tomcat.apache.org/whichversion.html) (<http://tomcat.apache.org/whichversion.html>).

1. Add **analytics-service.war** and the **analytics-ui.war** files to the Tomcat **webapps** folder.
2. Uncomment the following section in the **conf/server.xml** file, which is present, but commented out, in a freshly downloaded Tomcat archive.

```
<Valve className="org.apache.catalina.authenticator.SingleSignOn"/>
```

3. Declare the two war files in the **conf/server.xml** file, and define a user registry.

```
<Context docBase="analytics-service" path="/analytics-service"></Context>
<Context docBase="analytics" path="/analytics"></Context>
<Realm className="org.apache.catalina.realm.MemoryRealm"/>
```

The **MemoryRealm** recognizes the users that are defined in the **conf/tomcat-users.xml** file. For more information about other choices, see Apache Tomcat Realm Configuration HOW-TO (<http://tomcat.apache.org/tomcat-7.0-doc/realm-howto.html>).

4. Add the following sections to the **conf/tomcat-users.xml** file to configure a **MemoryRealm**.
 - Add the security roles.

```
<role rolename="analytics_administrator"/>
<role rolename="analytics_infrastructure"/>
<role rolename="analytics_support"/>
<role rolename="analytics_developer"/>
<role rolename="analytics_business"/>
```

- Add a few users with the roles you want.

```
<user name="admin" password="admin" roles="analytics_administrator"/>
<user name="support" password="demo" roles="analytics_support"/>
<user name="business" password="demo" roles="analytics_business"/>
<user name="developer" password="demo" roles="analytics_developer"/>
<user name="infrastructure" password="demo" roles="analytics_infrastructure"/>
```

- Start your Tomcat Server and go to the MobileFirst Analytics Console.

```
http://localhost:8080/analytics/console
```

For more information about how to start the Tomcat Server, see the official Tomcat site. For example, Apache Tomcat 7 (<http://tomcat.apache.org/tomcat-7.0-doc/introduction.html>), for Tomcat 7.0.

Installing MobileFirst Analytics on WebSphere Application Server

For more information on initial installation steps for acquiring the installation artifacts (JAR and EAR files), see Installing MobileFirst Server to an application server (../appserver). The **analytics.ear**, **analytics-ui.war**, and **analytics-service.war** files are found in the **\analytics** folder.

The following steps describe how to install and run the Analytics EAR file on WebSphere Application Server. If you are installing the individual WAR files on WebSphere Application Server, follow only steps 2 - 7 on the **analytics-service** WAR file after you deploy both WAR files. The class loading order must not be altered on the analytics-ui WAR file.

1. Deploy the EAR file to the application server, but do not start it. . For more information about how to install an EAR file on WebSphere Application Server, see the Installing enterprise application files with the console (http://ibm.biz/knowctr#SSAW57_8.5.5/com.ibm.websphere.nd.multiplatform.doc/ae/trun_app_instwiz.html) topic in the WebSphere Application Server product documentation.
2. Select the **MobileFirst Analytics** application from the **Enterprise Applications** list.



3. Click **Class loading and update detection**.



4. Set the class loading order to **parent last**.

General Properties

Class reloading options

☒ Override class reloading settings for Web and EJB modules

Polling interval for updated files
 Seconds

Class loader order

- ☐ Classes loaded with parent class loader first
- ☒ Classes loaded with local class loader first (parent last)

WAR class loader policy

- ☒ Class loader for each WAR file in application
- ☐ Single class loader for application

Apply OK Reset Cancel

5. Click **Security role to user/group mapping** to map the admin user.

Enterprise Applications

[Enterprise Applications](#) > [MobileFirst Platform Analytics](#) > **Security role to user/group mapping**

Security role to user/group mapping

Each role that is defined in the application or module must map to a user or group from the domain user registry. accessIds: The accessIds are required only when using cross realm communication in a multi domain scenario. For all other scenarios the accessId will be determined during the application start based on the user or group name. The accessIds represent the user and group information that is used for Java Platform, Enterprise Edition authorization when using the WebSphere default authorization engine. The format for the accessIds is user:realm/uniqueUserID, group:realm/uniqueGroupID. Entering wrong information in these fields will cause authorization to fail. AllAuthenticatedInTrustedRealms: This indicates that any valid user in the trusted realms be given the access. AllAuthenticated: This indicates that any valid user in the current realm be given the access.

Map Users... Map Groups... Map Special Subjects ▾

Select	Role	Special subjects	Mapped users	Mapped groups
<input type="checkbox"/>	analytics_administrator	None	admin	
<input type="checkbox"/>	analytics_infrastructure	None		
<input type="checkbox"/>	analytics_support	None		
<input type="checkbox"/>	analytics_developer	None		
<input type="checkbox"/>	analytics_business	None		

OK Cancel

6. Click **Manage Modules**.

Enterprise Applications

Enterprise Applications > MobileFirst Platform Analytics

Use this page to configure an enterprise application. Click the links to access pages for further configuring of the application or its modules.

Reports Operations Service Policies Routing Policies Configuration

General Properties

Name
MobileFirst Platform Analytics

Application reference validation
Issue warnings

Detail Properties

- Target specific application status
- Startup behavior
- Application binaries
- Class loading and update detection
- Request dispatcher properties
- Security role to user/group mapping
- JASPI provider
- Custom properties
- View Deployment Descriptor
- Last participant support extension

References

- Shared library references
- Shared library relationships

Modules

- Manage Modules
- Display module build ids

Web Module Properties

- Session management
- Context Root For Web Modules
- Environment entries for Web modules
- Initialize parameters for servlets
- JSP and JSF options
- Virtual hosts

Enterprise Java Bean Properties

- Default messaging provider references

Client Module Properties

- Client module deployment mode

Database Profiles

- SQLJ profiles and pureQuery bind files

Apply OK Reset Cancel

7. Select the **analytics** module and change the class loader order to **parent last**.

Enterprise Applications

Enterprise Applications > MobileFirst Platform Analytics > Manage Modules > analytics-service.war

Use this page to configure an instance of a deployed web module in the application. This page contains deployment-specific information for a web module and session management settings.

Configuration

General Properties

URI
analytics-service.war

Alternate deployment descriptor

Starting weight
10000

Class loader order
Classes loaded with local class loader first (parent last)

Additional Properties

- View Module Class Loader
- Custom properties
- Target specific application status
- View Deployment Descriptor
- Session Management
- Web Module Proxy Configuration

Apply OK Reset Cancel

8. Enable **Administrative security** and **application security** in the WebSphere Application Server administration console:
- Log in to the WebSphere Application Server administration console.
 - In the **Security > Global Security** menu, ensure that **Enable administrative security** and **Enable application security** are both selected. Note: Application security can be selected only after **Administrative security** is enabled.
 - Click **OK** and save changes.

9. Start the MobileFirst Analytics application and go to the link in the browser: `http://<hostname>:<port>/analytics/console`.

Installing MobileFirst Analytics with Ant tasks

Ensure that you have the necessary WAR and configuration files: **analytics-ui.war** and **analytics-service.war**. For more information on the installation artifacts, see Installing MobileFirst Server to an application server (`../appserver`). The **analytics-ui.war** and **analytics-service.war** files are found in the **MobileFirstPlatformServer\analytics**.

You must run the Ant task on the computer where the application server is installed, or the Network Deployment Manager for WebSphere® Application Server Network Deployment. If you want to start the Ant task from a computer on which MobileFirst Server is not installed, you must copy the file **/MobileFirstServer/mfp-ant-deployer.jar** to that computer.

Note: The **mfserverinstall_dir** placeholder is the directory where you installed MobileFirst Server.

1. Edit the Ant script that you use later to deploy MobileFirst Analytics WAR files.
 - Review the sample configuration files in Sample configuration files for MobileFirst Analytics (`../installation-reference/#sample-configuration-files-for-mobilefirst-analytics`).
 - Replace the placeholder values with the properties at the beginning of the file.

Note: The following special characters must be escaped when they are used in the values of the Ant XML scripts:

- The dollar sign (\$) must be written as \$\$, unless you explicitly want to reference an Ant variable through the syntax `${variable}`, as described in the Properties (<http://ant.apache.org/manual/properties.html>) section of the Apache Ant Manual.
- The ampersand character (&) must be written as &, unless you explicitly want to reference an XML entity.
- Double quotation marks (") must be written as ", except when it is inside a string that is enclosed in single quotation marks.

2. If you install a cluster of nodes on several servers:
 - You must uncomment the property **wl.analytics.masters.list**, and set its value to the list of host name and transport port of the master nodes. For example:
`node1.mycompany.com:96000,node2.mycompany.com:96000`
 - Add the attribute **mastersList** to the **elasticsearch** elements in the tasks **installanalytics**, **updateanalytics**, and **uninstallanalytics**.

Note: If you install on a cluster on WebSphere Application Server Network Deployment, and you do not set the property, the Ant task computes the data end points for all the members of the cluster at the time of installation, and sets the **masternodes** JNDI property to that value.

3. To deploy the WAR files, run the following command: `ant -f configure-appServer-analytics.xml install`
You can find the Ant command in **mfserverinstall_dir/shortcuts**. This installs a node of MobileFirst Analytics, with the default type master and data, on the server, or on each member of a cluster if you install on WebSphere Application Server Network Deployment.
4. Save the Ant file. You might need it later to apply a fix pack or perform an upgrade. If you do not want to save the passwords, you can replace them by "*****" (12 stars) for interactive prompting.

Note: If you add a node to a cluster of MobileFirst Analytics, you must update the `analytics/masternodes` JNDI property, so that it contains the ports of all the master nodes of the cluster.

Installing MobileFirst Analytics Server on servers running previous versions

Although there is no option to upgrade previous versions of the MobileFirst Analytics Server, when you install MobileFirst Analytics Server V8.0.0 on a server that hosted a previous version, some properties and analytics data need to be migrated.

For servers previously running earlier versions of MobileFirst Analytics Server update the analytics data and the JNDI properties.

Migration of server properties used by previous versions of MobileFirst Analytics Server

If you install MobileFirst Analytics Server V8.0.0 on a server that was previously running an earlier version of MobileFirst Analytics Server, you must update the values of the JNDI properties on the hosting server.

Some event types were changed between earlier versions of MobileFirst Analytics Server and V8.0.0. Because of this change, any JNDI properties that were previously configured in your server configuration file must be converted to the new event type.

The following table shows the mapping between old event types and new event types. Some event types did not change.

Old event type	New event type
AlertDefinition	AlertDefinition
AlertNotification	AlertNotification
AlertRunnerNode	AlertRunnerNode
AnalyticsConfiguration	AnalyticsConfiguration
CustomCharts	CustomChart
CustomData	CustomData
Devices	Device
MfpAppLogs	AppLog
MfpAppPushAction	AppPushAction
MfpAppSession	AppSession
ServerLogs	ServerLog
ServerNetworkTransactions	NetworkTransaction
ServerPushNotifications	PushNotification
ServerPushSubscriptions	PushSubscription
Users	User
inboundRequestURL	resourceURL
mfpAppName	appName
mfpAppVersion	appVersion

Analytics data migration

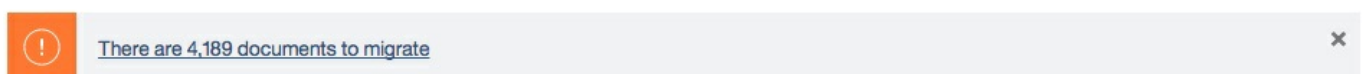
The internals of the MobileFirst Analytics Console were improved, which required changing the format in which the data is stored. To continue to interact with the analytics data that was already collected, the data must be migrated into the new data format.

When you first view the MobileFirst Analytics Console after you upgrade to V8.0.0, no statistics are rendered in the MobileFirst Analytics Console. Your data is not lost, but it must be migrated to the new data format.

An alert is displayed on every page of the MobileFirst Analytics Console that reminds you that documents must be migrated. The alert text includes a link to the **Migration** page.

The following image shows a sample alert from the **Overview** page of the **Dashboard** section:

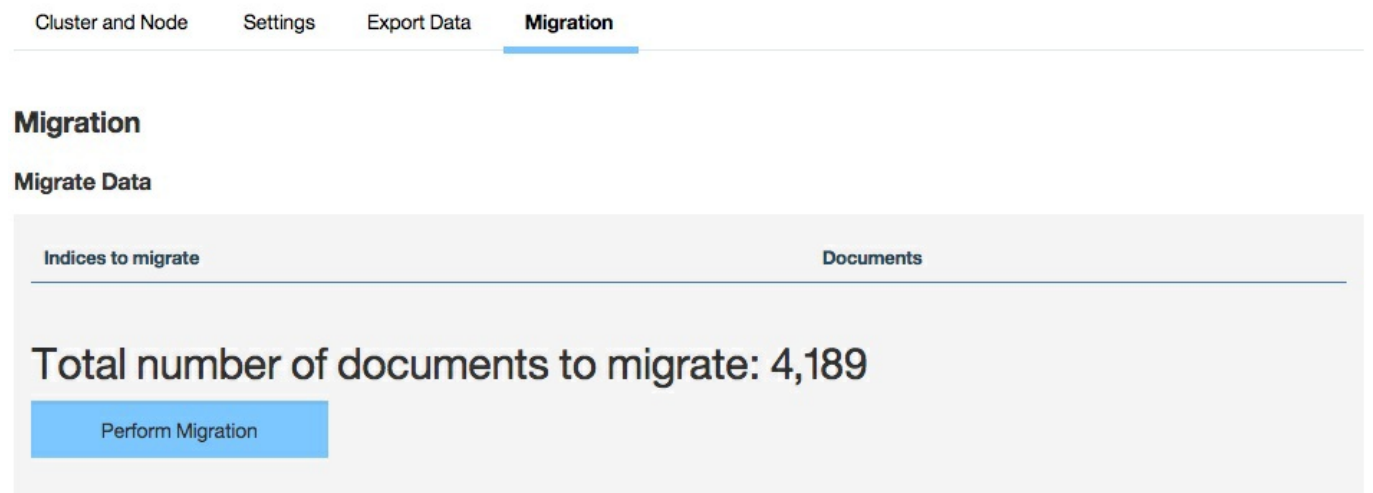
Overview



Migration page

You can access the Migration page from the wrench icon in the MobileFirst Analytics Console. From the **Migration** page, you can see how many documents must be migrated, and which indices they are stored on. Only one action is available: **Perform Migration**.

The following image shows the **Migration** page when you have documents that must be migrated:



Note: This process might take a long time, depending on the amount of data you have, and it cannot be stopped during migration.

The migration can take approximately 3 minutes to migrate 1 million documents on a single node with 32G of RAM, with 16G allocated to the JVM, with a 4-core processor. Documents that are not migrated are not queried, so they are not rendered in the MobileFirst Analytics Console.

If the migration fails while in progress, retry the migration. Retrying the migration does not remigrate documents that were already migrated, and your data integrity is maintained.