Migrating existing adapters to work under MobileFirst Server V8.0.0

Overview

Starting with v8.0 of MobileFirst Server, adapters are Maven projects. Learn how to upgrade adapters that were developed under earlier versions of MobileFirst Server.

This page describes the steps to take to migrate adapters that were developed to work with MobileFirst Server V6.2 or later so that they work with MobileFirst Server v8.0
To start, study the changes in adapter APIs that are described in Deprecated features and API elements and Server-side API changes in v8.0 (../../product-overview/release-notes/deprecated-discontinued/).

- Under certain conditions, existing adapters work as-is with MobileFirst Server v8.0. See Using older adapters as-is under MobileFirst Server V8.0.
- In most cases, you need to upgrade the adapters. For Java™ adapters, see Migrating Java adapters to Maven projects for MobileFirst Server v8.0. For JavaScript adapters, see Migrating JavaScript adapters to Maven projects for MobileFirst Server v8.0.

Using older adapters as-is under MobileFirst Server v8.0

An existing adapter can be deployed as-is under MobileFirst Server v8.0, unless it matches any of the following criteria:

Adapter type	Condition
Java	Uses the PushAPI or SecurityAPI interfaces

Adapter type	Condition

JavaScript	• Was built using IBM Worklight V6.2 or earlier.
	• Uses a connection type that is not HTTP or SQL.
	• Contains procedures with securityTest customization
	• Contains procedures that use the user identity to connect to the back end
	• Uses any of the following APIs:
	  ◦ WL.Device.*
	  ◦ WL.Geo.*
	  ◦ WL.Server.readSingleJMSMessage
	  ◦ WL.Server.readAllJMSMessages
	  ◦ WL.Server.writeJMSMessage
	  ◦ WL.Server.requestReplyJMSMessage
	  ◦ WL.Server.getActiveUser
	  ◦ WL.Server.setActiveUser
	  ◦ WL.Server.getCurrentUserIdentity
	  ◦ WL.Server.getCurrentDeviceIdentity
	  ◦ WL.Server.createEventSource
	  ◦ WL.Server.createDefaultNotification
	  ◦ WL.Server.getUserNotificationSubscription
	  ◦ WL.Server.notifyAllDevices
	  ◦ WL.Server.notifyDeviceToken
	  ◦ WL.Server.notifyDeviceSubscription
	  ◦ WL.Server.sendMessage
	  ◦ WL.Server.createEventHandler
	  ◦ WL.Server.setEventHandlers
	  ◦ WL.Server.setApplicationContext
	  ◦ WL.Server.fetchNWBusinessObject
	  ◦ WL.Server.createNWBusinessObject
	  ◦ WL.Server.deleteNWBusinessObject
	  ◦ WL.Server.updateNWBusinessObject
	  ◦ WL.Server.getBeaconsAndTriggers
	  ◦ WL.Server.signSoapMessage
	  ◦ WL.Server.createSQLStatement

Migrating Java adapters to Maven projects for MobileFirst Server v8.0

1. Create a Maven adapter project with the archetype **adapter-maven-archetype-java**. When setting the parameter **artifactId** use the adapter name and for the parameter **package** use the same package as the one in the existing Java adapter. For more information, see Creating Java adapters (../../adapters/creating-adapters).
2. Overwrite the adapter-descriptor file (**adapter.xml**) under **src**/**main**/**adapter-resources** in the created project from the existing Java adapter. For more details about the descriptor, see The Java

adapter-descriptor file (../../adapters/java-adapters/#the-adapter-resources-folder).

3. Remove all the files under **src**/**main**/**java** in the created project from the existing Java adapter, then copy all the Java files under the old adapter's **src** folder, but preserve the same folder structure. Copy all the non-Java files under the **src** folder of the old adapter to the **src**/**main**/**resources** of the new adapter. By default, **src**/**main**/**resources** does not exist, so if the adapter contains non-Java files, create it. For the changes in Java adapter APIs, see Server-side API changes in v8.0.

   The following diagrams illustrate the structure of adapters up to v7.1 and Maven adapters, starting from v8.0:

```
├── adapters
│   └── RSSAdapter
│       ├── RSSAdapter.xml
│       ├── lib
│       └── src
│           └── com
│               └── sample
│                   ├── RSSAdapterApplication.java
│                   └── RSSAdapterResource.java
```

   New structure of a Java adapter:

```
├── pom.xml
├── src
│   └── main
│       ├── adapter-resources
│       │   └── adapter.xml
│       └── java
│           └── com
│               └── sample
│                   ├── RSSAdapterApplication.java
│                   └── RSSAdapterResource.java
```

4. Using either of the following methods, add any JAR files that are not in the Maven repository:
   - Add the JAR files to a local repository, as described in Guide to installing third-party JARs (https://maven.apache.org/guides/mini/guide-3rd-party-jars-local.html), then add them to **dependencies** element.
   - Add the JAR files to the dependencies element by using the **systemPath** element. For more information, see Introduction to the Dependency Mechanism (https://maven.apache.org/guides/introduction/introduction-to-dependency-mechanism.html).

Migrating JavaScript adapters to Maven projects for MobileFirst Server v8.0

1. Create a Maven adapter project with the archetype **adapter-maven-archetype-http or adapter-maven-archetype-sql**. When setting the parameter **artifactId** use the adapter name. For more information, see Creating JavaScript adapters (../../adapters/creating-adapters).
2. Overwrite the adapter-descriptor file (**adapter.xml**) under **src**/**main**/**adapter-resources** in the created project from the existing JavaScript adapter. For details about the descriptor, see The JavaScript adapter-descriptor file (../../adapters/javascript-adapters/#the-adapter-resources-folder).
3. Overwrite the JavaScript files **src**/**main**/**adapter-resources**/**js** in the created project from the

existing JavaScript adapter JavaScript files.

Last modified on