

Known Issues and Limitations

Known Issues

Click the following link to receive a dynamically generated list of documents for this specific release and all its fix packs, including known issues and their resolutions, and relevant downloads: <http://www.ibm.com/support/search.wss?tc=SSVNUQ&tc=SSHT2F&atr=SWVersion&atr=8.0> (<http://www.ibm.com/support/search.wss?tc=SSVNUQ&tc=SSHT2F&atr=SWVersion&atr=8.0>).

Known Limitations

In this documentation, you can find the description of IBM MobileFirst Foundation known limitations in different locations:

- When the known limitation applies to a specific feature, you can find its description in the topic that explains this specific feature. You can then immediately identify how it affects the feature.
- When the known limitation is general, that is, applies to different and possibly not directly related topics, you can find its description here.

Globalization

If you are developing globalized apps, the following restrictions apply:

- Partial translation: Part of the product IBM MobileFirst Foundation v8.0, including its documentation, is translated in the following languages: Simplified Chinese, Traditional Chinese, French, German, Italian, Japanese, Korean, Portuguese (Brazil), Russian, and Spanish. User-facing text is translated.
- Bidirectional support: The applications that are generated by IBM MobileFirst Foundation are not fully bidirectional enabled. Mirroring of the graphic user interface (GUI) elements and the control of the text direction are not provided by default. However, no hard dependency exists from the generated applications on this limitation. It is possible for the developers to achieve full bidi compliance by manual adjustments in the generated code.

Although translation into Hebrew is provided for IBM MobileFirst Foundation core functionality, some GUI elements are not mirrored.

- Constraints on adapter names: Names of adapters must be valid names to create a Java class name. In addition, they must be composed only of the following characters:
 - Uppercase and lowercase letters (A-Z and a-z)
 - Digits (0-9)
 - Underscore (_)
- Unicode characters: Unicode characters outside the Basic Multilingual Plane are not supported.
- Language sensitivity and Unicode Normalization Forms: In the following use cases, queries do not consider language sensitivity such as normal matching, accent-insensitive, case-insensitive, and 1-to-2 mapping for the search function to run correctly in different languages, and search on data does not use Normalization Form C (NFC).
 - From the MobileFirst Analytics Console, when you create a custom filter for a custom chart. However, in this console, the message property uses Normalization Form C (NFC) and considers language sensitivity.
 - From the MobileFirst Operations Console, when you search for an application in the Browse Applications page, for an adapter in the Browser Adapters page, for a tag in the Push page, or a device on the Devices page.
 - In the Find functions for the JSONStore API.

IBM MobileFirst Analytics

The IBM MobileFirst Analytics has the following limitations:

- Security analytics (data on requests failing security checks) is not supported.
- In MobileFirst Analytics Console, the format for numbers does not follow the International Components for Unicode (ICU) rules.
- In MobileFirst Analytics Console, the numbers do not use the user's preferred number script.
- In MobileFirst Analytics Console, the format for dates, times, and numbers are displayed according to the language setting of the operating system rather than the locale of Microsoft Internet Explorer.
- When you create a custom filter for a custom chart, the numerical data must be in base 10, Western, or European numerals, such as 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.
- When you create an alert in the Alert Management page of MobileFirst Analytics Console, the numerical data must be in base 10, Western, or European numerals, such as 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.
- The Analytics page of the MobileFirst Operations Console supports the following browsers:
 - Microsoft Internet Explorer version 10 or later
 - Mozilla Firefox ESR or later

- Apple Safari on iOS version 7.0 or later
- Google Chrome latest version
- The Analytics client SDK is not available for Windows.

IBM MobileFirst Application Center mobile client

The Application Center mobile client follows the cultural conventions of the running device, such as the date formatting. It does not always follow the stricter International Components for Unicode (ICU) rules.

IBM MobileFirst Operations Console

The MobileFirst Operations Console has the following limitations:

- Provides only partial support for bidirectional languages.
- The text direction cannot be changed when notification messages are sent to an Android device:
 - If the first letters typed are in a right-to-left language, such as Arabic and Hebrew, the whole text direction is automatically right-to-left.
 - If the first letters typed are in a left-to-right language, the whole text direction is automatically left-to-right.
- The sequence of characters and text alignment do not match cultural fashion in the bidirectional language.
- The numeric fields do not parse numeric values according to the formatting rules of the locale. The console displays formatted numbers but accept as input only *raw* (unformatted) numbers. For example: 1000, not 1 000, or 1,000.

Server Configuration Tool

The Server Configuration Tool has the following restrictions:

- The descriptive name of a server configuration can contain only characters that are in the system character set. On Windows, it is the ANSI character set.
- Passwords that contain single quotation mark or double quotation mark characters might not work correctly.
- The console of the Server Configuration Tool has the same globalization limitation as the Windows console to display strings that are out of the default code page.

You might also experience restrictions or anomalies in various aspects of globalization because of limitations in other products, such as browsers, database management systems, or software development kits in use. For example:

- You must define the user name and password of the Application Center with ASCII characters only. This limitation exists because WebSphere® Application Server (full or Liberty profiles) does not support non-ASCII passwords and user names. See Characters that are valid for user IDs and passwords.
- On Windows:
 - To see any localized messages in the log file that the test server creates, you must open this log file with the UTF8 encoding.
 - These limitations exist because of the following causes:
 - The test server is installed on WebSphere Application Server Liberty profile, which creates log file with the ANSI encoding except for its localized messages for which it uses the UTF8 encoding.
- In Java 7.0 Service Refresh 4-FP2 and previous versions, you cannot paste Unicode characters that are not part of the Basic Multilingual Plane into the input field. To avoid this issue, create the path folder manually and choose that folder during the installation.
- Custom title and button names for the alert, confirm, and prompt methods must be kept short to avoid truncation at the edge of the screen.
- JSONStore does not handle normalization. The Find functions for the JSONStore API do not take account of language sensitivity such as accent insensitive, case insensitive, and 1-to-2 mapping.

Adapters and third-party dependencies

The following known issues pertain to interactions between dependencies and classes in the application server, including the MobileFirst shared library.

Apache HttpClient

IBM MobileFirst Foundation uses Apache HttpClient internally. If you add an Apache HttpClient instance as a dependency to a Java adapter, the following APIs do not work properly in the adapter: `AdaptersAPI.executeAdapterRequest`, `AdaptersAPI.getResponseAsJSON`, and `AdaptersAPI.createJavascriptAdapterRequest`. The reason is that the APIs contain Apache HttpClient types in their signature. The workaround is to use the internal Apache HttpClient but to change the dependency scope in the **pom.xml** provided.

Bouncy Castle cryptographic library

IBM MobileFirst Foundation uses Bouncy Castle itself. It might be possible to use another version of Bouncy Castle in the adapter, but the consequences need to be carefully tested: sometimes, the MobileFirst Bouncy Castle code populates certain static Singleton fields of the `javax.security` package classes and might prevent the version of Bouncy Castle that is inside an adapter from using features that rely on those fields.

Apache CXF implementation of JAR files

CXF is used in the MobileFirst JAX-RS implementation, thus preventing you from adding Apache CXF JAR files to an adapter.

Application Center mobile client: refresh issues on Android 4.0.x

Android 4.0.x WebView component is known to have several refresh issues. Updating devices to Android 4.1.x should provide a better user experience.

If you build the Application Center client from sources, disabling the hardware acceleration at the application level in the Android manifest should improve the situation for Android 4.0.x. In that case, the application must be built with Android SDK 11 or later.

Application Center requires MobileFirst Studio V7.1 for importing and building the Application Center mobile client

To build the Application Center mobile client, you need MobileFirst Studio V 7.1. You can download MobileFirst Studio from the Downloads page (file:///home/travis/build/MFPSamples/DevCenter/_site/downloads). Click the **Previous MobileFirst Platform Foundation releases** tab for the download link. For installation instructions, see Installing MobileFirst Studio in the IBM® Knowledge Center for 7.1 (https://www.ibm.com/support/knowledgecenter/SSHS8R_7.1.0/com.ibm.worklight.installconfig.doc/devenv/t_installing_ibm_worklight_studi.html). For more information about building the Application Center mobile client, see Preparations for using the mobile client ([../appcenter/preparations](https://www.ibm.com/support/knowledgecenter/SSHS8R_7.1.0/com.ibm.worklight.installconfig.doc/devenv/t_installing_ibm_worklight_studi.html)).

Application Center and Microsoft Windows Phone 8.1

Application Center supports the distribution of applications as Windows Phone application package (.xap) files for Microsoft Windows Phone 8.0 and Microsoft Windows Phone 8.1. With Microsoft Windows Phone 8.1, Microsoft introduced a new universal format as app package (.appx) files for Windows Phone. Currently, Application Center does not support the distribution of app package (.appx) files for Microsoft Windows Phone 8.1, but is limited to Windows Phone application package (.xap) files only.

Application Center supports the distribution of app package (.appx) files for Microsoft Windows Store (Desktop applications) only.

Administering MobileFirst applications through Ant or through the command line

The `mfpadm` tool is not available if you download and install only the IBM MobileFirst Foundation Developer Kit. The `mfpadm` tool is installed with the MobileFirst Server with the installer.

Direct Update

Direct Update on Windows is not supported in V8.0.0.

FIPS 140-2 feature limitations

The following known limitations apply when you use the FIPS 140-2 feature in IBM MobileFirst Foundation: * This FIPS 140-2 validated mode applies only to the protection (encryption) of local data that is stored by the JSONStore feature and protection of HTTPS communications between the MobileFirst client and the MobileFirst Server. * For HTTPS communications, only the communications between the MobileFirst client and the MobileFirst Server use the FIPS 140-2 libraries on the client. Direct connections to other servers or services do not use the FIPS 140-2 libraries. * This feature is only supported on the iOS and Android platforms. * On Android, this feature is only supported on devices or simulators that use the x86 or armeabi architectures. It is not supported on Android using armv5 or armv6 architectures. The reason is because the OpenSSL library used did not obtain FIPS 140-2 validation for armv5 or armv6 on Android. FIPS 140-2 is not supported on 64-bit architecture even though the MobileFirst library does support 64-bit architecture. FIPS 140-2 can be run on 64-bit devices if the project includes only 32-bit native NDK libraries. * On iOS, it is supported on i386, x86_64, armv7, armv7s, and arm64 architectures. * This feature works with hybrid applications only (not with native applications). * For native iOS, FIPS is enabled through the iOS FIPS libraries and is enabled by default. No action is required to enable FIPS 140-2. * The use of the user enrollment feature on the client is not supported by the FIPS 140-2 feature. * The Application Center client does not support the FIPS 140-2 feature.

Installation of a fix pack or interim fix to the Application Center or the MobileFirst Server

When you apply a fix pack or an interim fix to Application Center or MobileFirst Server, manual operations are required, and you might have to shut down your applications for some time.

JSONStore supported architectures

For Android, JSONStore supports the following architectures: ARM, ARM v7, and x86 32-bit. Other architectures are not currently supported. Trying to use other architectures leads to exceptions and potential application crashes.

JSON Store is not supported for Windows native applications.

Liberty server limitations

If you use the Liberty server on a 32-bit JDK 7, Eclipse might not start, and you might receive the following error: "Error occurred during initialization of VM. Could not reserve enough space for object heap. Error: Could not create the Java Virtual Machine. Error: A fatal exception has occurred. Program will exit."

To fix this issue, use the 64-bit JDK with the 64-bit Eclipse and 64-bit Windows. If you use the 32-bit JDK on a 64-bit computer, you might configure JVM preferences to **mx512m** and **Xms216m**.

LTPA token limitations

An `SESN0008E` exception occurs when an LTPA token expires before the user session expires.

An LTPA token is associated with the current user session. If the session expires before an LTPA token expires, a new session is created automatically. However, when an LTPA token expires before a user session expires, the following exception occurs:

```
com.ibm.websphere.servlet.session.UnauthorizedSessionRequestException: SESN0008E: A user authenticated as anonymous has attempted to access a session owned by {user name}
```

To resolve this limitation, you must force the user session to expire when the LTPA token expires. * On WebSphere Application Server Liberty, set the `httpSession` attribute `invalidateOnUnauthorizedSessionRequestException` to `true` in the `server.xml` file. * On WebSphere Application Server, add the session management custom property `InvalidateOnUnauthorizedSessionRequestException` with the value `true` to fix the issue.

Note: On certain versions of WebSphere Application Server or WebSphere Application Server Liberty, the exception is still logged, but the session is correctly invalidated. For more information, see APAR PM85141 (<http://www.ibm.com/support/docview.wss?uid=swg1PM85141>).

Microsoft Windows Phone 8

For Windows Phone 8.1 environments, x64 architecture is not supported.

Microsoft Windows 10 UWP apps

Application authenticity feature does not work on MobileFirst Windows 10 UWP apps when the MobileFirst SDK is installed through the NuGet package. As a workaround, developers can download the NuGet package and add the MobileFirst SDK references manually.

MobileFirst Operations Console

Analytics page

Response times in the Analytics page of the MobileFirst Operations Console depend on several factors, such as hardware (RAM, CPUs), quantity of accumulated analytics data, and IBM MobileFirst Analytics clustering. Consider testing your load before you integrate IBM MobileFirst Analytics into production.

Nested projects can result in unpredictable results with the CLI

Do not nest projects inside one another when using the IBM MobileFirst Command Line Interface (CLI). Otherwise, the project that is acted upon might not be the one that you expect.

Previewing Cordova web resources with the Mobile Browser Simulator

You can preview your web resources with Mobile Browser Simulator, but not all MobileFirst JavaScript APIs are supported by the simulator. In particular, the OAuth protocol is not fully supported. However, you can test calls to adapters with `WLResourceRequest`.

Physical iOS device required for testing extended app authenticity

The testing of the extended app authenticity feature requires a physical iOS device, because an IPA cannot be installed on an iOS simulator.

Support of Oracle 12c by MobileFirst Server

The installation tools of the MobileFirst Server (Installation Manager, Server Configuration Tool, and Ant tasks) support installation with Oracle 12c as a database.

The users and tables can be created by the installation tools but the database, or databases, must exist before you run the installation tools.

Support for push notification

Non-secured push is supported in Cordova (on iOS and Android).

Updating cordova-ios platform

To update the cordova-ios platform of a Cordova app, you must uninstall and reinstall the platform by completing the following steps:

1. Navigate to the project directory for the app by using the command-line interface.

2. Run the `cordova platform rm ios` command to remove the platform.
3. Run the `cordova platform add ios@version` command to add the new platform to the app, where version is the version of the Cordova iOS platform.
4. Run the `cordova prepare` command to integrate the changes.

The update fails if you use the `cordova platform update ios` command.

WKWebView support for iOS Cordova applications

App notifications and Direct Update features might not work well in iOS Cordova apps with WKWebView.

This limitation is due to the defect `file:// url XMLHttpRequests` are not allowed in WKWebViewEngine in **cordova-plugin-wkwebview-engine**.

To circumvent this issue, run the following command in your Cordova project: `cordova plugin add https://github.com/apache/cordova-plugins.git#master:wkwebview-engine-localhost`

Executing this command would run a local web server in your Cordova application, you can then host and access your local files instead of using the file URI scheme (`file://`) to work with local files.

Note: This Cordova plug-in is not published to the Node package manager (npm).

cordova-plugin-statusbar does not work with Cordova application loaded with cordova-plugin-mfp.

cordova-plugin-statusbar will not work with Cordova application loaded with cordova-plugin-mfp.

To circumvent this issue, the developer will have to set `CDVViewController` as the root view controller. Replacing the code snippet in the `wlInitDidCompleteSuccessfully` method as suggested below in the **MFPAppdelegate.m** file of the Cordova iOS project.

Existing code snippet:

```
(void)wlInitDidCompleteSuccessfully
{
    UIViewController* rootViewController = self.window.rootViewController;
    // Create a Cordova View Controller
    CDVViewController* cordovaViewController = [[CDVViewController alloc] init] ;
    cordovaViewController.startPage = [[WL sharedInstance] mainHtmlFilePath];
    // Adjust the Cordova view controller view frame to match its parent view bounds
    cordovaViewController.view.frame = rootViewController.view.bounds;
    // Display the Cordova view [rootViewController addChildViewController:cordovaViewController];
    [rootViewController.view addSubview:cordovaViewController.view];
    [cordovaViewController didMoveToParentViewController:rootViewController];
}
```

Recommended code snippet with workaround for the limitation:

```
(void)wlInitDidCompleteSuccessfully
{
    // Create a Cordova View Controller
    CDVViewController* cordovaViewController = [[CDVViewController alloc] init] ;
    cordovaViewController.startPage = [[WL sharedInstance] mainHtmlFilePath];
    [self.window setRootViewController:cordovaViewController];
    [self.window makeKeyAndVisible];
}
```

Raw IPv6 address not supported in Android applications

During the configuration of **mfpclient.properties** for your native Android application, if your MobileFirst Server is on a host with IPv6 address, then use a mapped host name for the IPV6 address to configure the **wlServerHost** property in **mfpclient.properties**. Configuring the **wlServerHost** property with raw IPv6 address fails the application's attempt to connect to the MobileFirst Server.