

Using MobileFirst Developer CLI to Manage MobileFirst Artifacts

Overview

IBM MobileFirst Platform Foundation provides a Command Line Interface (CLI) tool - `mfpdev` - to easily manage MobileFirst client and server artifacts.

By using the CLI, you can manage Cordova-based applications that uses the MobileFirst Cordova plug-in, and Native applications that uses the MobileFirst Native SDK.

You can also create, register, and manage MobileFirst adapters to either local or remote MobileFirst Server instances, and administer projects from the command line or via REST services, or from the MobileFirst Operations Console.

For more information regarding SDK integration in native and Cordova applications, see the tutorials in the Adding the MobileFirst Platform Foundation SDK category ([../..adding-the-mfpf-sdk/](#))

In this tutorial you will learn how to install the `mfpdev` Command Line Interface (CLI) and how to use it to manage MobileFirst Server instances, applications and adapters.

Learn more about the MobileFirst CLI in the "MobileFirst Platform Command Line Interface" topic in the user documentation

Jump to

- Prerequisites
- Installing the mfpdev CLI
- Interactive and Direct modes
- Managing MobileFirst Server instances
- Managing Applications
- Managing and Testing Adapters
- Helpful commands
- Update and Uninstall the Command Line Interface

Prerequisites

The Command Line Interface is available as a NPM package at NPM registry (<https://www.npmjs.com/>). You must have node.js installed in your environment in order to install NPM packages.

Access nodejs.org (<https://nodejs.org>) and follow the instructions there to install node.js.

To confirm that node.js is installed properly, execute the command `node -v`.

```
$ node -v
v4.2.3
```

Note: Minimum supported node.js version is 4.2.3

TODO: Confirm if node.js 5.0+ will be supported at GA

Installing the mfpdev CLI

To install the Command Line Interface execute the command:

```
$ npm install -g mfpdev-cli
```

To confirm the installation execute the command `mfpdev` without any arguments and it will print the help text.

mfpdev

NAME

IBM MobileFirst Platform Foundation Command Line Interface (CLI).

SYNOPSIS

mfpdev <command> [options]

DESCRIPTION

The IBM MobileFirst Platform Foundation Command Line Interface (CLI) is a command-line for developing MobileFirst applications. The command-line can be used by itself, or in conjunction with the IBM MobileFirst Platform Foundation Operations Console. Some functions are available from the command-line only and not the console.

For more information and a step-by-step example of using the CLI, see the IBM Knowledge Center for your version of IBM MobileFirst Platform Foundation at

<https://www.ibm.com/support/knowledgecenter>.

ADAPTERS

adapter build Builds a MobileFirst Adapter
adapter create Creates a MobileFirst Adapter

APPLICATIONS

app preview Previews your app.
app pull Pulls application artifacts from a server
app push Pushes application artifacts to a server
app register Registers an application or applications to a server

GLOBAL

config Sets your configuration preferences.
info Displays information about your environment.

SERVERS

server add Adds a new server definition.
server clean Clean apps and adaptors in an existing server.
server console Opens the MobileFirst console.
server edit Edits an existing server definition.
server info Provides information for all servers or a specific server.
server remove Removes an existing server definition.

COMMAND-LINE FLAGS/OPTIONS

-v, --version Prints out this utility's version
-d, --debug Debug mode produces debug log output
-dd, --ddebug Debug mode produces verbose log output
--no-color Suppresses use of color in command output

EXAMPLE USAGE

```
$ mfpdev register
$ mfpdev pulls
$ mfpdev push
```

Interactive and Direct modes

All commands can be executed in interactive or direct mode. In the interactive mode, the parameters required for the command will be prompted and some default values will be used. In direct mode, the parameters must be provided with the command being executed.

Example:

`mfpdev server add` in interactive mode:

```
$ mfpdev server add
? Enter the name of the new server definition: mydevserver
? Enter the fully qualified URL of this server: http://mydevserver.example.com:9080
? Enter the MobileFirst Server administrator login ID: admin
? Enter the MobileFirst Server administrator password: *****
? Save the admin password for this server?: Yes
? Enter the context root of the MobileFirst administration services: mfpadmin
? Enter the MobileFirst Server connection timeout in seconds: 30
Verifying server configuration...
The following runtimes are currently installed on this server: mfp
Server profile 'mydevserver' added successfully.
```

The same command in direct mode would be

```
mfpdev server add mydevserver --url http://mydevserver.example.com:9080 --login admin --password admin
--setdefault
```

To find what is the right syntax for a command in direct mode use `mfpdev help <command>`

Managing MobileFirst Server instances

You can use `mfpdev server <option>` command to manage the instances of MobileFirst Server that are in use. There must be always at least one server instance listed as the default instance. The default server is always used when another one is not specified.

List server instances

To list all the MobileFirst Server instances available to be used, execute the command

```
$ mfpdev server info
```

By default, a local server profile is created automatically and used as the current default by the CLI

Add a new server instance

If you are using another local or remote MobileFirst Server instance you can add it to the list of instances available to be used with the command:

```
$ mfpdev server add
```

Follow the interactive prompt to provide a name to the server, the server URL and user/password credentials.

Edit server instances

If you want to edit the details of a registered server instance, execute the command

```
$ mfpdev server edit
```

And follow the interactive prompt to select the server to be edited and provide the information to be updated
To set a server as the default one execute

```
$ mfpdev server edit <server_name> --setdefault
```

Remove server instances

To remove a server instance from the list of registered servers, execute the command:

```
$ mfpdev server remove
```

And select the server from the interactive list

Open server console

To open the console of the default server registered execute the command

```
$ mfpdev server console
```

To open the console of another server, not the default one, inform the server name as a parameter of the command

```
$ mfpdev server console <server_name>
```

Remove apps and adapters from a server

To remove all apps and adapters registered in a server execute the command

```
$ mfpdev server clean
```

And select the server to clean form the interactive prompt. This will put the server instance in a clean state without any app or adapter deployed.

Managing applications

The command `mfpdev app <option>` can be used to manage applications created with the MobileFirst Platform SDK.

Register an application in a server instance

An application created with MobileFirst Platform SDK, must be registered in a MobileFirst Server when it is ready to be executed.

To register an app, run the following command from the root folder of the app project.

```
$ mfpdev app register
```

This command can be executed from the root of a Cordova, Android, iOS or Windows application.

It will use the default server and runtime to execute the following tasks:

- Register an application with a server
- Generate a default client properties file for the application
- Put the server information into the client properties file

For a Cordova application, this command will update the config.xml file For an iOS application, this command will update the file mfpclient.plist For an Android or Windows application, this command will update the file mfpclient.properties

To register an app to a server and runtime that is not the default one, use the syntax

```
$ mfpdev app register <server> <runtime>
```

For Cordova Windows platform, the `-w <platform>` argument must be added to the command. The `<platform>` argument is a comma separated list of the windows platforms to be registered. Valid values are `windows`, `windows8` and `windowsphone8`

```
$ mfpdev app register -w windows8
```

Config an application

When an application is registered, server related attributes are added to the app configuration file.

To change the value of those attributes, use the following command

```
$ mfpdev app config
```

This command will interactively present a list of attributes that can be changed and prompt for the new value of the attribute.

The attributes available will vary for each platform (Android, iOS, Windows).

Preview a Cordova application

A Cordova application can be previewed using a browser. This is a way to preview an app faster then using the platform specific emulators and simulators.

To preview a Cordova application execute the following command form the app root

```
$ mfpdev app preview
```

You will be prompted to select what platform to preview and what type of preview to be used. There are two options of preview, MBS and Browser. MBS - Mobile Browser Simulator - will simulate a mobile device on browser. Browser - Simple Browser Rendering - will present the www resources of the cordova application as a usual browser web page.

For more details about the preview options see the [UPDATE- Tutorial about preview](#) ([../url/to/be/updated](#))

Update web resources for Direct Update

The web resources of a cordova app, like .html, .css and .js files inside www folder can be updated without the need to reinstall the app at the mobile device. This is possible with the Direct Update feature provided by MobileFirst Platform.

For more details about how Direct Update works see the tutorial [Using Direct Update in Cordova applications \(../direct-update\)](#)

When you want to send a new set of web resources to be updated in a cordova application, execute the command

```
$ mfpdev app webupdate
```

This command will package the updated web resources to a .zip file and upload it to the default MobileFirst Server registered. The packaged web resources can be found at the [cordova-project-root-folder]/mobilefirst/ folder.

To upload the web resources to different server instance, inform the server name and runtime as part of the command

```
$ mfpdev app webupdate <server_name> <runtime>
```

You can use the --build parameter to generate the .zip file with the packaged web resources without uploading it to a server.

```
$ mfpdev app webupdate --build
```

To upload a package that was previously built, use the --file parameter

```
$ mfpdev app webupdate --file mobilefirst/com.ibm.test-android-1.0.0.zip
```

There is also the option to encrypt the content of package using the --encrypt parameter

```
$ mfpdev app webupdate --encrypt
```

Pull and Push the MobileFirst Application configuration

After a MobileFirst Application is registered in a MobileFirst server, it is possible to change some of the application configurations using the MobileFirst Server Console and then pull those configurations from the server to the application with the following command

```
$ mfpdev app pull
```

It is also possible to change the application configurations locally and push the changes to the MobileFirst Server with the command

```
$ mfpdev app push
```

Managing and Testing Adapters

It is possible to manage MobileFirst Adapters with the command `mfpdev adapter <option>`.

To learn more about MobileFirst Adapters see the tutorials at the [Adapters \(../adapters/\)](#) category.

Create a MobileFirst Adapter

```
$ mfpdev adapter create
```

Build a MobileFirst Adapter

```
$ mfpdev adapter build
```

Deploy a MobileFirst Adapter

```
$ mfpdev adapter deploy
```

Execute a MobileFirst Adapter from the command line

```
$ mfpdev adapter call
```

Helpful commands

- `mfpdev config`
- `mfpdev help`
- `mfpdev info`
- `mfpdev -v`

Update and Uninstall the Command Line Interface

To update the command line interface execute the command:

```
npm update -g mfpdev-cli
```

To uninstall the command line interface execute the command:

```
npm uninstall -g mfpdev-cli
```

Tutorials to follow next

to be updated...