

Using CLI to create, build, and manage MobileFirst project artifacts

fork and edit tutorial (<https://github.com/MobileFirst-Platform-Developer-Center/DevCenter/>) | report issue (<https://github.com/MobileFirst-Platform-Developer-Center/DevCenter/issues/new>)

Overview

This tutorial covers the following topics regarding the IBM MobileFirst Platform Foundation CLI:

- Installation of the Command Line Interface (CLI)
- Creating MobileFirst projects and native API
- Creating hybrid applications
- Creating test servers and the build-and-deploy flow
- Creating and testing adapters
- Using the Export command
- Importing CLI-generated projects into MobileFirst Studio

Before continuing with this tutorial, you must first register

(https://www14.software.ibm.com/webapp/iwm/web/signup.do?source=swg-worklight&S_PKG=ov1268&S_CMP=web_dw_rt_swd) in order to download the CLI installer.
URL: https://www14.software.ibm.com/webapp/iwm/web/signup.do?source=swg-worklight&S_PKG=ov1268&S_CMP=web_dw_rt_swd

Installation of the Command Line Interface (CLI)

1. Extract the `mobilefirst-cli-installer-7.0.0.zip` archive and navigate to the `mobilefirst-cli-installer-7.0.0` folder.
2. Select the appropriate installer for your operating system and run it.
3. Click **Install** to begin the installation.



4. Accept the Software License Agreement.
5. Select the installation folder.



6. Accept the Pre-Installation Summary by clicking **Install** and then select **Done** when installation is

complete.



Creating MobileFirst projects and native API

The CLI installer adds the installation folder to your `path`, so that IBM MobileFirst Platform Foundation CLI commands can be run from any directory.

Create a MobileFirst project from the CLI by using either of these commands:

`mobilefirst create`, or `mfp create`

Example:

```
~/projects $ mfp create MyProject
```

A MobileFirst Project was successfully created at /Users/cli-user/projects/MyProject

Helpful Commands

`mobilefirst help` – Shows how to use all the commands.

```
~/projects $ mfp help
```

NAME

mobilefirst

SYNOPSIS

mfp <command> [options]

DESCRIPTION

IBM MobileFirst Platform Command Line Interface (CLI)

Global Commands

config Set your configuration preferences
console Opens the MobileFirst console
create Creates a new MobileFirst project
create-server Creates a new Liberty server
help Displays the command help
info Generates information about your environment
logs Displays the path to the local test server logs
restart Restarts the local test server
run Starts the local test server
start Starts the local test server
status Displays the status of the server
stop Stops the local test server

Project-Level Commands

adapter add Creates a new adapter
adapter call Call an adapter's procedure on the MobileFirst Server
add adapter Creates a new adapter
add api Creates a new Native API
add environment Adds a platform specific environment
add feature Adds optional features
add hybrid Creates a new hybrid application
add skin Creates MobileFirst skins
bd Builds and deploys MobileFirst resources
build Builds MobileFirst resources
build-config Configures an application's build settings
deploy Deploys the set of MobileFirst resources
export Creates a zip file of the project
invoke Invokes a procedure **for** a specified adapter
preview Displays a preview of the current application or environment
remove feature Removes optional features

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

Example usage

```
$ mfp create MyProject
$ cd MyProject
$ mfp add api MyiOS --environment ios
$ mfp add adapter MySQLAdapter --type sql
$ cd MySQLAdapter
$ mfp build
$ mfp deploy
```

`mobilefirst info` - Gives the OS release, system memory, node.js version, and MobileFirst CLI version.

```
~/projects $ mfp info
OS: darwin x64
Release: 14.1.0
System Memory: 4303 MB free out of 16384 MB
Node: v0.10.30
MobileFirst CLI: 7.0.0.00.*
Current directory: /Users/cli-user/projects
...
```

You can use the *interactive* or *direct* options to add native API. First navigate to the MobileFirst project directory.

Interactive Mode:

```
mobilefirst add api
```

Direct Mode:

```
mobilefirst add api --environment ios , or -e for environment
```

Example: (direct mode)

```
~/projects $ cd MyProject
~/projects/MyProject $ mfp add api -e ios MyiOSApi
A new iOS API was added at /Users/cli-user/projects/MyProject/apps/MyiOSApi
```

Creating hybrid applications

Add a hybrid application from the CLI

`mobilefirst add hybrid` – Creates a new hybrid application, which is generated into the `/apps` folder of the current project.

```
~/projects $ cd MyProject
~/projects/MyProject $ mfp add hybrid MyHybridApp
A new hybrid app was added at /Users/cli-user/projects/MyProject/apps/MyHybridApp
```

`mobilefirst add environment` – Adds an environment to your application. You can use this command in the project directory if the project contains only one app. Otherwise, select which app to add the environment to, or call the command from inside the directory of the hybrid app to which the environment should be added.

```
~/projects/MyProject $ mfp add environment
[?] To which hybrid app would you want to add environments? (Use the arrow keys.)
> MyHybridApp
  OtherHybridApp
```

By using the interactive mode, you can select one or more environments to add.

```
~/projects/MyProject/apps/MyHybridApp $ mfp add environment
[?] What environments you want to add to the hybrid app? (Press <space> to select)
> ☐ iPhone
☐ iPad
☐ Android phone and tablets
☐ BlackBerry 6 and 7 (deprecated)
☐ BlackBerry 10
☐ Windows Phone 8
☐ Windows 8 desktop and tablets
(Move up and down to reveal more choices.)
```

To add environments in direct mode:

```
~/projects/MyProject $ mfp add environment iphone,android --app MyHybridApp
A new android environment was added at /Users/cli-user/projects/MyProject/apps/MyHybridApp/android
A new iphone environment was added at /Users/cli-user/projects/MyProject/apps/MyHybridApp/iphone
```

Application skins

To add an application skin from the CLI, use these commands:

`mobilefirst add skin` – Interactive mode prompts for the platform to target and name. The current working directory must be under an existing hybrid application, and at least one environment must already be added to this app.

```
~/projects/MyProject/apps/MyHybridApp $ mfp add skin
[?] What platform do you want to target? (Use arrow keys)
> Android phone and tablets
  iPhone
[?] What do you want to be your skin name suffix? Your skin folder name will be 'platform.<suffix>'
A new android skin was added at /Users/cli-user/projects/MyProject/apps/MyHybridApp
```

To add a skin in direct mode:

```
mfp add skin [--environment|-e android|blackberry|blackberry10|iphone|ipad ]
```

Optional features

To add optional features to a hybrid application from the CLI, use these commands:

`mobilefirst add feature` – Interactive mode prompts for the features to add.

```
~/projects/MyProject/apps/MyHybridApp $ mfp add feature
[?] What feature you want to install in this application? NOTE: Features you have already installed are not shown:
  FIPS 140-2
  IBM Tealeaf SDK
> JSONStore
A new jsonstore Feature was added at /Users/cli-user/projects/MyProject/apps/MyHybridApp<br />
```

Direct:

```
~/projects/MyProject $ cd apps/MyHybridApp/
~/projects/MyProject/apps/MyHybridApp $ mfp add feature jsonstore
A new jsonstore Feature was added at /Users/cli-user/projects/MyProject/apps/MyHybridApp
```

To remove optional features from a hybrid application by using the CLI, use these commands:

`mobilefirst remove feature` – Interactive mode prompts for the features to remove.

```
~/projects/MyProject/apps/MyHybridApp $ mfp remove feature
[?] What feature you want to uninstall from this application? NOTE: Features you have already uninstalled are not shown: (Use arrow keys)
> JSONStore
A jsonstore Feature was removed from /Users/cli-user/projects/MyProject/apps/MyHybridApp
```

To remove a feature direct mode:

`mobilefirst remove feature [fips|jsonstore|tealeaf]`

Creating test servers and the build-and-deploy flow

The instance of the Liberty development server is created on the default user directory. For example:

`/Users/cli-user/.ibm/mobilefirst/7.0.0/server`

Server commands

`mfp start` - Starts the server. If the project is not deployed, this command runs the build-all-and-deploy flow.

`mfp stop` - Stops the server.

`mfp run` - Verbose server mode which outputs server events to a console or log.

`mfp status` - Gives the status of the server.

`mfp build` - Builds the project. When you are not in the root directory of the project, this command builds the current directory and subdirectories.

`mfp deploy` - Deploys anything that you built in the root directory of the project. The server is automatically created and started as needed. When you are in the adapter folder, this command deploys that adapter.

`mfp bd` - Builds and deploys (a combination of both commands in a single command).

`mfp preview [--noshell|-n]` - Displays a preview of the current application or environment in your default browser.

`mfp restart` - Restarts the local test server.

`mfp console` - Opens the MobileFirst Console in your default web browser.

Creating and testing adapters

`mfp adapter add` - interactive mode

```
~/projects/MyProject $ mfp adapter add
[?] What do you want to name your MobileFirst Adapter? MyHttpAdapter
[?] What type of adapter would you like? HTTP
[?] Create procedures for offline JSONStore? No
[?] Create procedures for USSD enablement? No
A new HTTP adapter was added at /Users/cli-user/projects/MyProject/adapters/MyHttpAdapter
```

`mfp adapter add --type http`, or `-t` for type. The types of adapters that can be added are HTTP, SQL, Cast Iron, SAP Netweaver Gateway, JMS, SAP Jco, and Java.

```
~/projects/MyProject $ mfp adapter add --type http TestAdapter
A new http Adapter was added at /Users/cli-user/projects/MyProject/adapters/TestAdapter
```

`mfp adapter call` - interactive mode

```
~/projects/MyProject $ mfp adapter call
[?] Which adapter do you want to use? MyHttpAdapter
[?] Which endpoint do you want to use? MyHttpAdapter/getStories
[?] Enter the comma-separated parameters: "world"
[?] How should the procedure be called? GET
Calling GET '/MyProject/adapters/MyHttpAdapter/getStories?params=["world"]'
Response:
{
  "statusCode": 200,
  "errors": [],
  "isSuccessful": true,
  "statusReason": "OK",
  "rss": {
    "feedburner": "http://rssnamespace.org/feedburner/ext/1.0",
    "channel": {
      "pubDate": "Thu, 26 Mar 2015 12:36:00 EDT",
      "title": "CNN.com - World",
      "description": "CNN.com delivers up-to-the-minute news and information on the latest top stories, weath
er, entertainment, politics and more.",
    }
  }
}
```

To invoke and adapter in direct mode:

`mfp invoke [:[\\"\\"| --file|-f]]`

```
~/projects/MyProject $ mfp invoke MyHttpAdapter:getStories \"world\"
{
  "statusCode": 200,
  "errors": [],
  "isSuccessful": true,
  "statusReason": "OK",
  "rss": {
    "feedburner": "http://rssnamespace.org/feedburner/ext/1.0",
    "channel": {
      "pubDate": "Thu, 26 Mar 2015 12:36:00 EDT",
      "title": "CNN.com - World",
      "description": "CNN.com delivers up-to-the-minute news and information on the latest top stories, weath
er, entertainment, politics and more.",
    }
  }
}
```

Editing files

BYOE - Bring Your Own Editor: Use your favorite text editor or IDE to develop from the CLI.


```
~/projects/MyProject $ vim adapters/MyHttpAdapter
" =====
" Netrw Directory Listing                      (netrw v140)
" /Users/cli-user/projects/MyProject/adapters/MyHttpAdapter
" Sorted by      name
" Sort sequence: [V]$, \<core\%(\\.d\\+\\)\|=>, \.h$, \.c$, \.cpp$, *, \.o$, \.obj$, \.info$, \.swp$, \.bak$, \~$
" Quick Help: <F1>:help  -:go up dir  D:delete  R:rename  s:sort-by  x:exec
" =====
../
.swm
.swn
.swo
MyHttpAdapter-impl.js
MyHttpAdapter.xml
filtered.xsl*
.MyHttpAdapter.xml.swp
.swp
~
~
"adapters/MyHttpAdapter" is a directory
```

Using the Export command

Exporting a project in interactive mode

By using the `export` command, you can create a compressed file, which contains the entire MobileFirst project, or the optimized hybrid assets to use in a native application.

`mfp export` - Running this command in the project root folder prompts you to enter the path and name of the compressed file to export to.

The resulting compressed file is intended to be shared with other MobileFirst developers. The compressed file contains source artifacts and everything another developer would need to build the missing artifacts.

```
~/projects/MyProject $ mfp export
[?] Where do you want to export the project? /Users/cli-user/Desktop
[?] What do you want to name your zip project? MyProject
Project successfully exported to /Users/cli-user/Desktop/MyProject.zip
```

Exporting a project in direct mode

`mfp export []` - If run from the root folder of the project, direct mode takes one additional argument, which is the full path to the compressed file to create.

```
~/projects/MyProject $ mfp export /Users/cli-user/Documents/MyProject.zip
Project successfully exported to /Users/cli-user/Documents/MyProject.zip
```

Exporting hybrid assets in interactive mode

1. First run the `mfp build` command to make sure that the app is built.
2. Run the hybrid asset export within the environment folder of the existing hybrid app. Interactive mode asks whether to include the native libraries, the path, and file name of the compressed file to export to.

The native libraries are built files. Native application IDEs need to ensure that the hybrid assets can be used in a native app.

```
~/projects/MyProject/apps/MyHybridApp/iphone $ mfp export
[?] Would you like to include the native libraries? Yes
[?] Where do you want to export the project? /Users/cli-user/Documents
[?] What do you want to name your zip project? MyiOSHybridApp
Project successfully exported to /Users/cli-user/Documents/MyiOSHybridApp.zip
```

Exporting hybrid assets in direct mode

`mfp export [] [-i | --includeNativeLibs]` - In direct mode, provide the full path to export hybrid assets. If the arguments `-i` or `--includeNativeLibs` are supplied, the native libraries are included.

```
~/projects/MyProject/apps/MyHybridApp/iphone $ mfp export /Users/cli-user/Documents/MyiOSHybridApp.zip -i
Project successfully exported to /Users/cli-user/Documents/MyiOSHybridApp.zip
```

Importing CLI-generated projects into MobileFirst Studio

From Eclipse, select **File > Import > Existing Projects into Workspace**.

Import

Select

Create new projects from an archive file or directory.



Select an import source:

- ▼ General
 - Archive File
 - Existing Projects into Workspace**
 - File System
 - Preferences
- ▶ Android
- ▶ C/C++
- ▶ CVS
- ▶ EJB
- ▶ Git
- ▶ Install
- ▶ Java EE
- ▶ Maven
- ▶ Plug-in Development
- ▶ Remote Systems
- ▶ Run/Debug
- ▶ Tasks



< Back

Next >

Cancel

Finish

Select root directory of the projects to import



projects



Search

Name

Size

Date Modified

Kind

▶ MyProject

--

Today, 2:28 PM

Folder

Cancel

Open

Click to working sets



The CLI can also open an existing MobileFirst Studio Eclipse project. By using the command line, navigate to a workspace that was created by Eclipse.

```
~/Users/cli-user/Documents/eclipse-workspace/MyProject $ ls
drwxr-xr-x  4 cli-user  staff  136 Mar 26 13:06 adapters
drwxr-xr-x  4 cli-user  staff  136 Mar 26 12:43 apps
drwxr-xr-x 11 cli-user  staff  374 Mar 26 14:38 bin
-rwxrwxr-x  1 cli-user  staff  408 Mar 26 10:59 build.properties
drwxr-xr-x  2 cli-user  staff   68 Mar 26 12:31 components
drwxr-xr-x  4 cli-user  staff  136 Mar 26 12:31 externalServerLibraries
drwxr-xr-x  5 cli-user  staff  170 Mar 26 12:31 server
drwxr-xr-x  2 cli-user  staff   68 Mar 26 12:31 services
```

For more information, see the topics about the Command Line Interface in the user documentation.