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

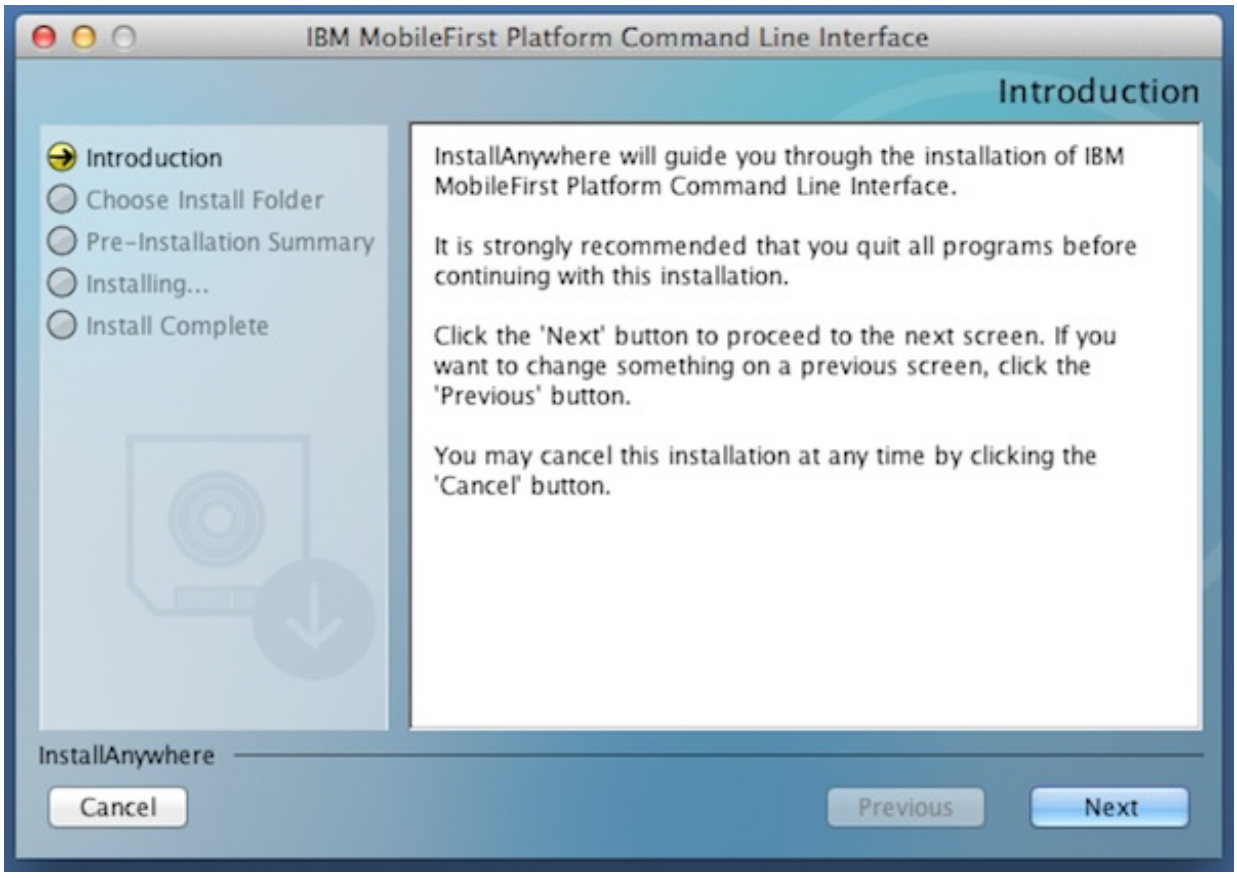
Topics covered in this tutorial:

- 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

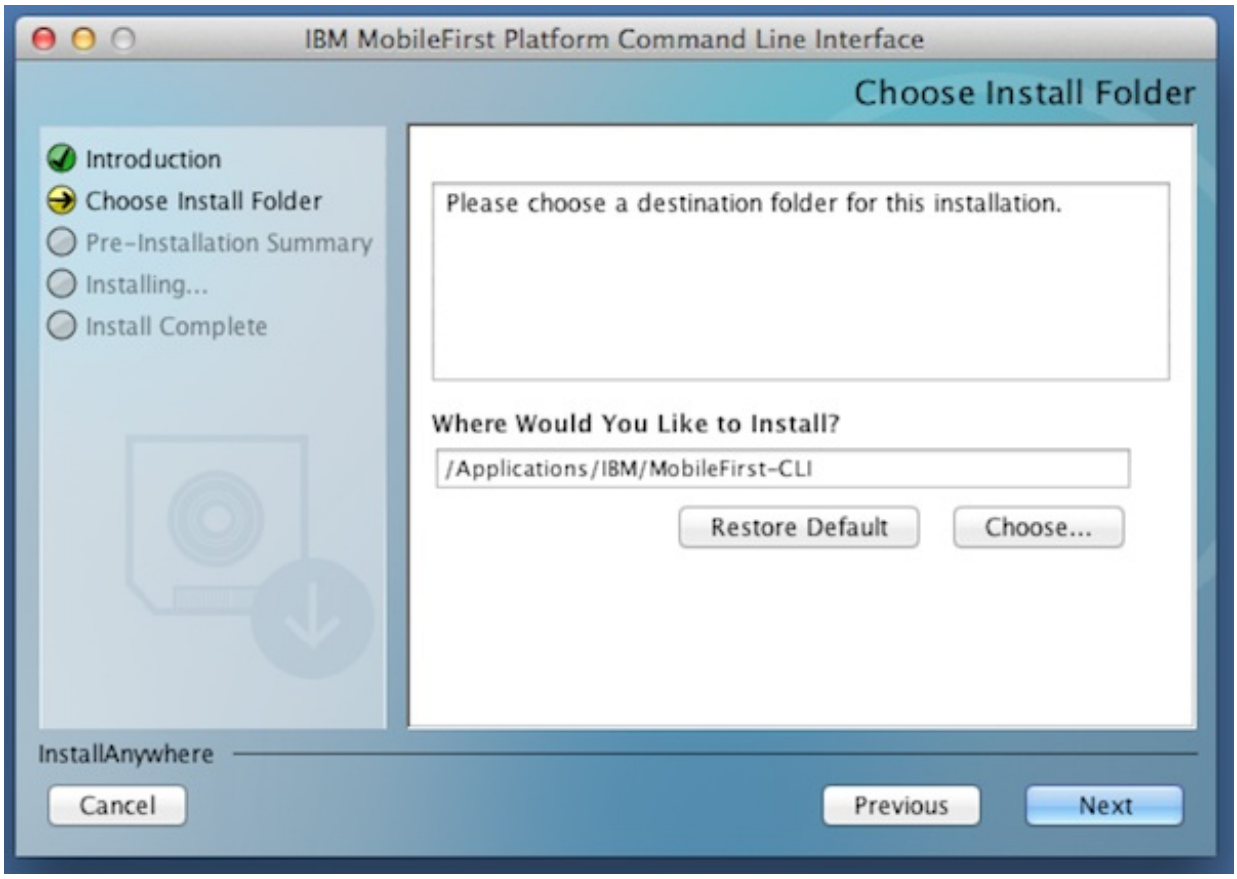
Installation of the Command Line Interface (CLI)

To get the CLI installer, follow through this page first (https://www14.software.ibm.com/webapp/iwm/web/signup.do?source=swg-worklight&S_PKG=ov1268&S_CMP=web_dw_rt_swd).

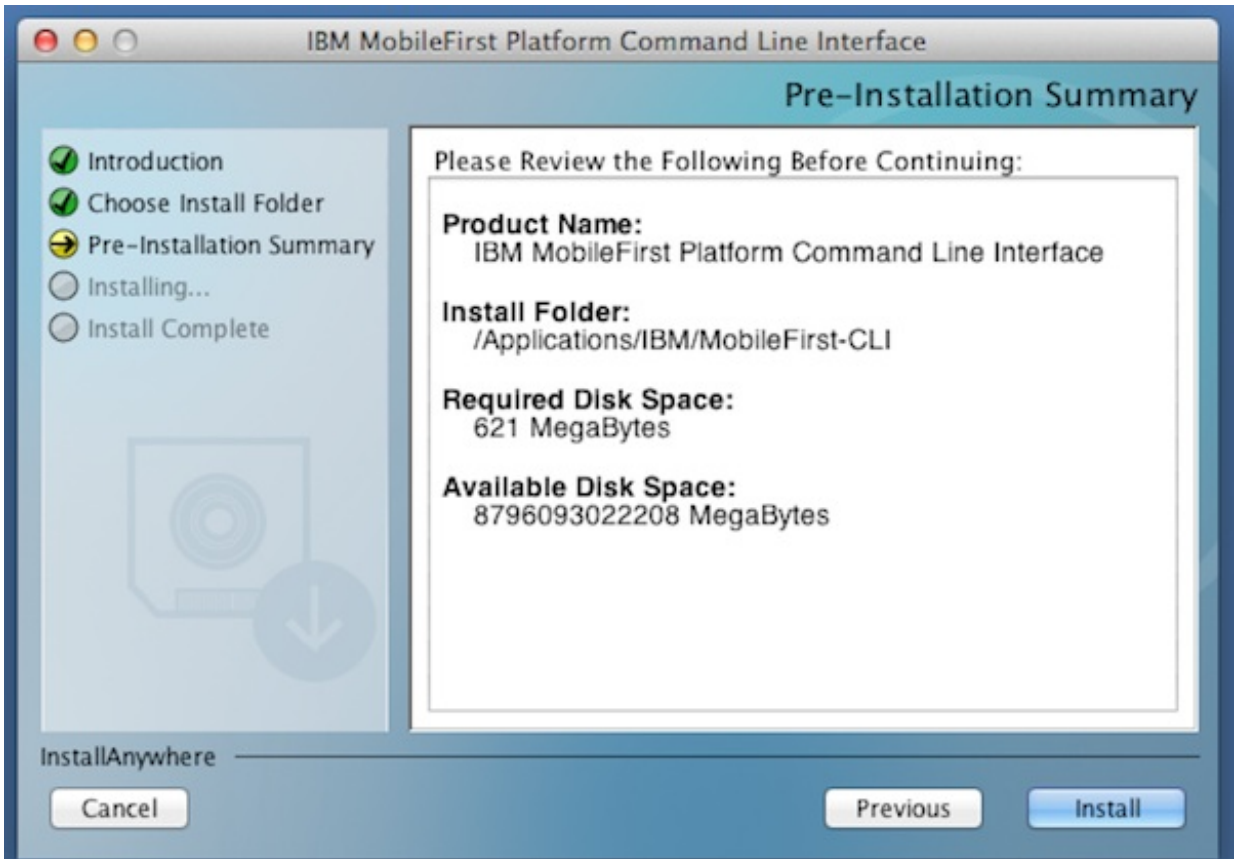
1. Extract the **mobilefirst-cli-installer-6.3.zip** archive and navigate to the **mobilefirst-cli-installer-6.3** folder.
2. Select the appropriate installer for your operating system and run it.
3. Click the **Install** icon to begin the installation.



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 CLI commands can be run from any directory.

Create a MobileFirst project from the CLI by using either of these commands:
`mobilefirst create projectName`, or `mfp create projectName`
Example:

```
~/CLIexample $ mfp create projectName
A MobileFirst Project was successfully created at /Users/CLIUser/CLIexample/pro
jectName
```

Helpful Commands

`mobilefirst help` – Shows all command usage:

```
~/CLIexample $ mfp help
NAME
     mobilefirst -- IBM MobileFirst Platform Command Line Interface (CLI).
SYNOPSIS
     mfp <command> [options]
DESCRIPTION
     Command-line interface to create and manage the applications for IBM Mobil
eFirst Platform.
Global Commands
     config [%lt;setting%gt;] [<value>]
     This command allows you to set your configuration preferences.
     console
     This command opens the MobileFirst console in your default browser for the
     current working directory of your project.
     create [%lt;name%gt;]
     This command creates a new MobileFirst project in the current working
     directory.
     create-server
     This command creates a new Liberty server in your default folder.
     The server is configured to work as a MobileFirst local test server.
     help [%lt;command%gt;]
     Displays the syntax summary or command help.
     status
     This command shows the status (running or stopped) of the local test
     server.
     stop
     This command stops the local test server.
Project-Level Commands
     add adapter [<name> --type|-t <adaptertype> [--jsonstore|-j] [--ussd|-u]]
     This command creates a new adapter that is generated into the adapters
     folder of the current project.
     add api [<name> --environment|-e <environment>]
     This command creates a new Native API that is generated into the apps
     folder of the current project.
     add environment [<type>[,...] [--app|-a <app>]]
     This command adds a platform specific environment to a hybrid application.
     You can run mfp add environment (without any arguments). Prompts then
     request the environments that will be generated.
     ...
Command-line Flags/Options
     -v, --version     prints out this utility's version
     -d, --debug       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 OS release, system memory, node.js version, IBM MobileFirst CLI version

```
~/CLIexample $ mfp info
OS: darwin x64
Release: 13.4.0
System Memory: 3984MB free out of 16384MB
Node: v0.10.30
MobileFirst CLI: 6.3.0.00
...
```

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

Interactive:
`mobilefirst add api`

Direct:
`mobilefirst add api --environment ios apiName`, or `-e` for environment

Example: (direct mode)

```
~/CLIexample $ cd projectName
~/CLIexample/projectName $ mfp add api -e ios apiName
A new iOS API was added at /Users/CLIUser/CLIexample/projectName/apps/apiName
```

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.

```
~/CLIexample $ cd projectName
~/CLIexample/projectName $ mfp add hybrid myApp
A new hybrid app was added at /Users/CLIUser/CLIexample/projectName/apps/myApp
```

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

```
~/CLIexample/myProject $ mfp add environment
[?] Which hybrid app would you want to add environments? (Use arrow keys)
❯ myApp
  myOtherApp
```

Interactive mode allows you to select one or more environments to add.

```
~/CLIexample/projectName/apps/myApp $ mfp add environment
[?] :What environments do you want to add to the hybrid app? (Press <space> to
select)
❯⬡ iPhone
 ⬡ iPad
 ⬡ Android phone and tablets
 ⬡ BlackBerry 6 and 7
 ⬡ BlackBerry 10
 ⬡ Windows Phone 8
 ⬡ Windows 8 desktop and tablets
(Move up and down to reveal more choices)
```

To add environments in direct mode:

```
~/CLIexample/myProject $ mfp add environment iphone,android --app myOtherApp
A new android Environment was added at /Users/CLIUser/CLIexample/myProject/apps
/myOtherApp/android
A new iphone Environment was added at /Users/CLIUser/CLIexample/myProject/apps/
myOtherApp/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.

```
~/CLIexample/projectName/apps/myApp $ mfp add skin
[?] What platform do you want to target? Android phone and tablets
[?] What do you want to be your skin name suffix? Your skin folder name will be
'platform.<suffix>' tablets
A new android Skin was added at /Users/CLIUser/CLIexample/projectName/apps/myAp
p
```

Direct mode:
`mfp add skin [--environment|-e android|blackberry|blackberry10|iphone|ipad skin-name]`

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.

```
~/CLIexample/projectName/apps/myHybrid $ mfp add feature
[?] What feature do you want to install in this application? NOTE: Features you
have already installed are not shown: (Use arrow keys)
  FIPS 140-2
  IBM Tealeaf SDK
❯ JSONStore
A new jsonstore feature was added at /Users/CLIUser/CLIexample/projectName/apps
/myHybrid
```

Direct:

```
~/CLIexample/myProject $ cd apps/myApp/
~/CLIexample/myProject/apps/myApp $ mfp add feature jsonstore
A new jsonstore Feature was added at /Users/CLIUser/CLIexample/myProject/apps/m
yApp
```

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.

```
~/CLIexample/projectName/apps/myHybrid $ mfp remove feature
[?] What feature do 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/CLIUser/CLIexample/projectName/apps
/myHybrid
```

`mobilefirst remove feature [fips|jsonstore|tealeaf]` – direct mode

Editing files

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

- Example - `vim myAdapterName`

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/CLIUser/.ibm/mobilefirst/6.3.0/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 that outputs server events to 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 will be 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 [environments-name --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 add adapter` - interactive mode

```
~/CLIexample/projectName/apps/myHybrid $ mfp add adapter
[?] 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/CLIUser/CLIexample/projectName/adapters/
myHttpAdapter
```

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

```
~/CLIexample/projectName/apps/myHybrid $ mfp add adapter --type http testAdapte
r
A new http Adapter was added at /Users/CLIUser/CLIexample/projectName/adapters/
testAdapter
```

`mfp invoke` - interactive mode

```
~/CLIexample/projectName/adapters/myHttpAdapter $ mfp invoke
[?] Which adapter do you want to use? myHttpAdapter
[?] Which procedure do you want to invoke? (Use arrow keys)
 ❯ getStories
   getStoriesFiltered
```

After a procedure is selected, you are prompted to enter parameters:

```
~/CLIexample/projectName/adapters/myHttpAdapter $ mfp invoke
[?] Which adapter do you want to use? myHttpAdapter
[?] Which procedure do you want to invoke? getStories
[?] Enter the comma-separated parameters: "world"
Invoking %s...
Arguments:
[
  "world"
]
Invocation result:
{
  "statusCode": 200,
  "errors": [],
  "isSuccessful": true,
  "statusReason": "OK",
  "rss": {
    "feedburner": "http://rssnamespace.org/feedburner/ext/1.0",
    "channel": {
      "pubDate": "Thu, 30 Oct 2014 16:30:19 EDT",
      "title": "CNN.com - World",
...
```

`mfp invoke adapterName:function [parameter1[,parameter2...]]` - direct mode

```
~/CLIexample/projectName/adapters/myHttpAdapter $ mfp invoke myHttpAdapter:getS
tories \"world\"
{<
  "statusCode": 200,
  "errors": [],
  "isSuccessful": true,
  "statusReason": "OK",
  "rss": {
    "feedburner": "http://rssnamespace.org/feedburner/ext/1.0",
    "channel": {
      "pubDate": "Tue, 04 Nov 2014 09:45:37 EST",
      "title": "CNN.com - World",
...
```

Using the Export command

Interactive Mode project export

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.

```
~/CLIexample/projectName $ mfp export
[?] Where do you want to export the project? /Users/CLIUser/Desktop
[?] What do you want to name your zip project? projectName
Project successfully exported to /Users/CLIUser/Desktop/projectName.zip
```

Direct Mode project export

`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.

```
~/CLIexample/projectName $ mfp export /Users/CLIUser/Desktop/myProject.zip
Project successfully exported to /Users/CLIUser/Desktop/myProject.zip
```

Interactive Mode hybrid export

First make sure that the app is built by using the `mfp build` command.
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, which native application IDEs need to ensure the hybrid assets can be used in a native app.

```
~/CLIexample/projectName/apps/myHybrid/iphone $ mfp export
[?] Would you like to include the native libraries? Yes
[?] Where do you want to export the project? /Users/CLIUser/Desktop
[?] What do you want to name your zip project? myProjectName.zip
Project successfully exported to /Users/CLIUser/Desktop/myProjectName.zip
```

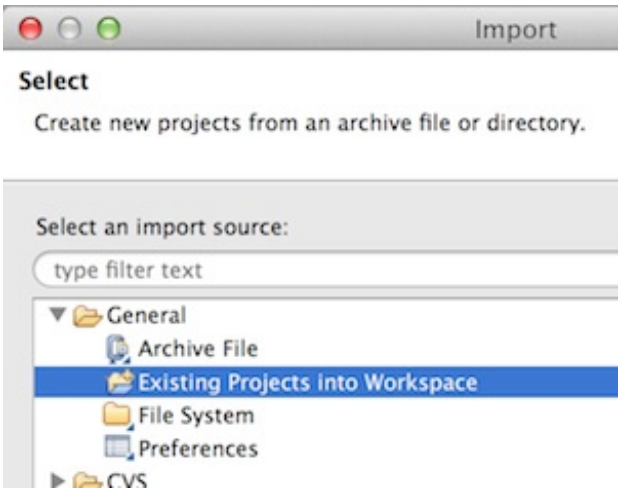
Direct Mode hybrid export

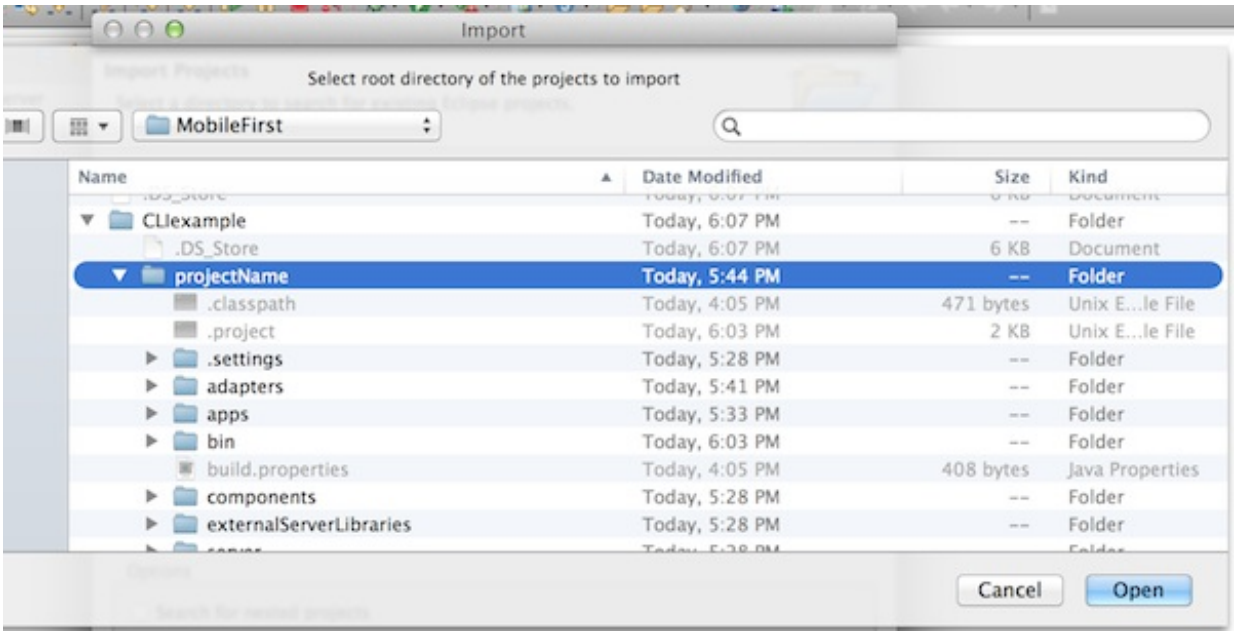
`mfp export [path to zip file] [-i | --includeNativeLibs]` - In direct mode, provide the full path to export hybrid assets. If the arguments `-i` or `--includeNativeLibs` are supplied, the native libraries are included.

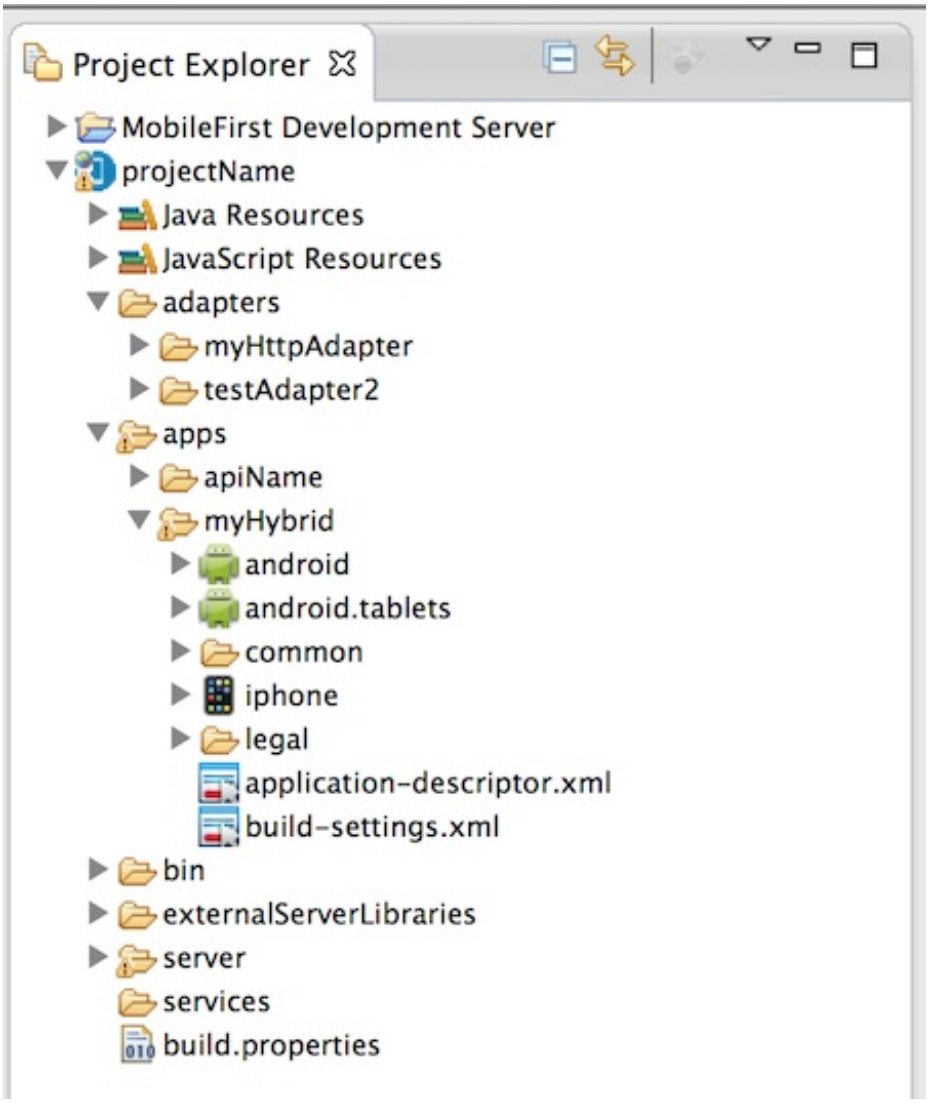
```
~/CLIexample/projectName/apps/myHybrid/iphone $ mfp export /Users/CLIUser/Deskt
op/projectName.zip --includeNativeLibs
Project successfully exported to /Users/CLIUser/Desktop/projectName.zip
```

Importing CLI-generated projects into MobileFirst Platform Studio

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







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

```
~/Users/CLIUser/Desktop/EclipseWorkspace/projectName $ ls
adapters    bin     externalServerLibraries services
apps      components    server
```

> For more information about the Command Line Interface, see the IBM MobileFirst Platform Foundation user documentation.