Using Direct Update in Cordova applications

Overview

With Direct Update, Cordova applications can be updated "over-the-air" with refreshed web resources, such as changed, fixed or new applicative logic (JavaScript), HTML, CSS or images. Organizations are thus able to ensure that end-users always use the latest version of the application.

In order to update an application, the updated web resources of the application need to be packaged and uploaded to the MobileFirst Server using the MobileFirst Developer CLI, which will then handle updating any application as it attempts to connect.

Supported Cordova platforms: iOS and Android.

> Direct Update updates only the application's web resources. To update native resources a new application version must be submitted to the respective app store.

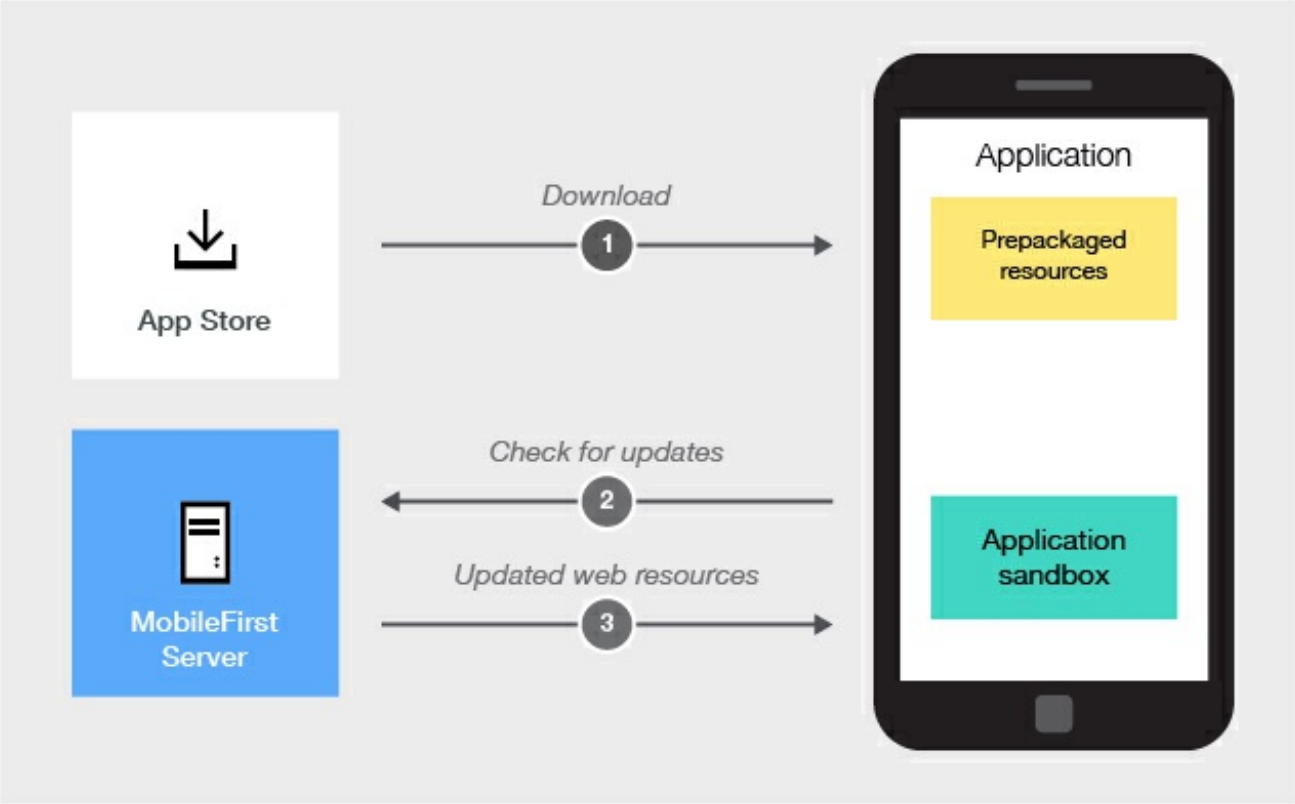
Jump to:

How Direct Update works

The application web resources are initially packaged with the application to ensure first offline availability. Afterwards, the application checks for updates on every request to the MobileFirst Server.

> **Note:** after a Direct Update was performed, it is checked for again after 60 minutes.

After a Direct Update, the application no longer uses the pre-packaged web resources. Instead, it will use the downloaded web resources from the application's sandbox. If the application's cache on the device will be cleared, the original packaged web resources will be used again.



Versioning

A Direct Update applies only to a specific version. In other words, updates generated for an application versioned 2.0 cannot be applied to a different version of the same application.

Creating and deploying updated web resources

Once work on new web resources, such as bug fixes or minor changes and the like, is done, the updated web resources need to be packaged and uploaded to the MobileFirst Server.

1. Open a **Command-line** window and navigate to the root of the Cordova project.
2. Run the command: `mfpdev app webupdate`.

The `mfpdev app webupdate` command packages the updated web resources to a .zip file and uploads it to the default MobileFirst Server running in the developer workstation. The packaged web resources can be found at the **[cordova-project-root-folder]/mobilefirst/** folder.

Alternatives:

- Build the .zip file and upload it to a different MobileFirst Server: `mfpdev app webupdate [server-name] [runtime-name]`. For example:

  ```
  mfpdev app webupdate myQAServer MyBankApps
  ```

- Upload a previously generated .zip file: `mfpdev app webupdate [server-name] [runtime-name] --file [path-to-packaged-web-resources]`. For example:

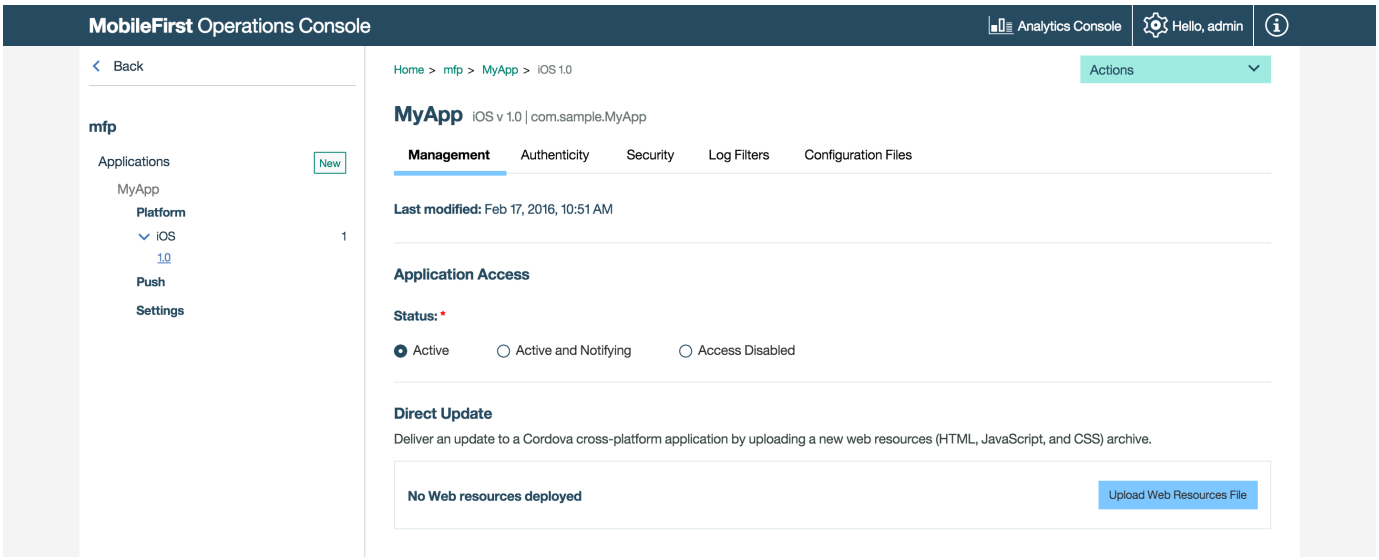
  ```
  mfpdev app webupdate myQAServer MyBankApps --file mobilefirst/ios/com.mfp.myBankApp-1.0.1.zip
  ```

- Manually upload packaged web resources to the MobileFirst Server:

1. Build the .zip file without uploading it:

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

2. Load the MobileFirst Operations Console and click on the application entry.
    1. Click on **Upload Web Resources File** to upload the packaged web resources.



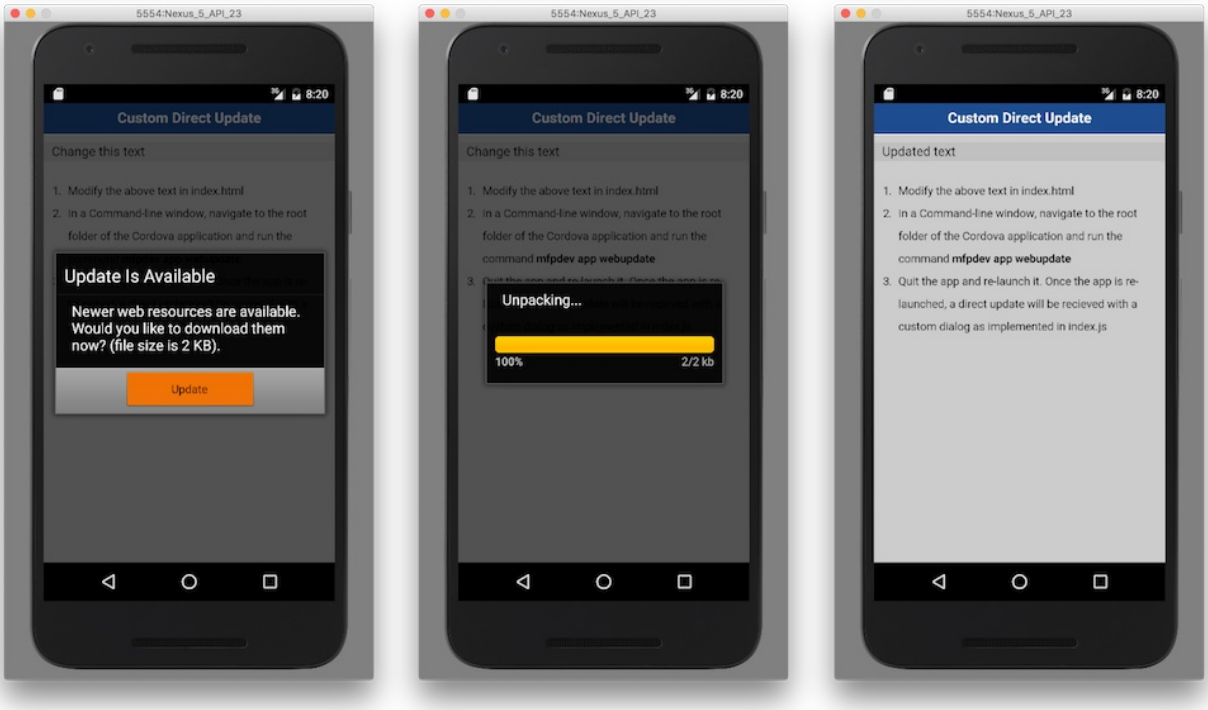
- The packaged web resources can be further protected by placing the web resources in an encrypted .zip file:

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

> Run the command `mfpdev help webupdate` to learn about additional command flags.

User Experience

By default, after a Direct Update is received a dialog is displayed and the user is asked whether to begin the update process. After the user approves a progress bar dialog is displayed and the web resources are downloaded. The application is automatically reloaded after the update is complete.



Direct Update authenticity

Disabled by default, Direct Update authenticity prevents a 3rd-party attacker from altering the web resources that are transmitted from the MobileFirst Server (or from a content delivery network (CDN)) to the client application.

To enable Direct Update authenticity:
Using a preferred tool, extract the public key from the MobileFirst Server keystore and convert it to base64. The produced value should then be used as instructed below:

1. Open a **Command-line** window and navigate to the root of the Cordova project.
2. Run the command: `mfpdev app config` and select the "Direct Update Authenticity public key" option.
3. Provide the public key and confirm.

Any future Direct Update deliveries to client applications will be protected by Direct Update authenticity.

> Refer to the "Direct Update" user documentation for more information about obtaining the public key required for Direct Update authenticity.

Delta and Full Direct Update

Delta Direct Updates enables an application to download only the files that were changed since the last update instead of the entire web resources of the application. This reduces download time, conserves bandwidth, and improves overall user experience.

> ❗ **Important:** A **delta update** is possible only if the client application's web resources are one version behind the application that is currently deployed on the server. Client applications that are more than one version behind the currently deployed application (meaning the application was deployed to the server at least twice since the client application was updated), receive a **full update** (meaning that the entire web resources are downloaded and updated).