

Microsoft Code Push Evaluation

What Is Code Push ?

Code Push allows you to update the javascript bundle and images (limited to specific components) in your ReactNative-App. Allowing your app to update without deploying to the App. Code Push allows you to change the UI/UX, business logic fix bugs, load hot-fixes.

While CodePush is in beta, it will remain completely free to use, with absolutely no limits but it doesn't offer any official SLA.

In order to ensure that your end users always have a functioning version of your app, the CodePush plugin maintains a copy of the previous update, so that in the event that you accidentally push an update which includes a crash, it can automatically roll back.

What Code are we Pushing ?

A React Native App is composed of Javascript files and images and native code from the bridged sections and in the view layer.

These are bundled together with the packager ios/android.bundle.js + images + native code into an IPA or an APK file.

CodePush allows you to update the ios/android.bundle.js and images, without going through the whole app store process (partial continuous delivery).

CodePush allows you to push client side assets only such as js bundle, images and etc.

CodePush doesn't allow to update native-side code (Java, Objective-C)

Getting started

1.Install the CodePush CLI

To install it, open a command prompt or terminal, and type

```
npm install -g code-push-cli
```

2.Create a CodePush account

To do this, simply type the following command via the CLI and authenticate with either your GitHub or Microsoft account:

```
code-push register
```

3.Register your app with service

```
code-push app add MyApp-iOS ios react-native
```

```
code-push app add MyApp-Android android react-native
```

4.CodePush-ify your app

Add the appropriate CodePush client SDKs to your app, and configure them to query for updates against the app deployment created above.

The following provide details on how to do this:

Run the following command from within your app's root directory:

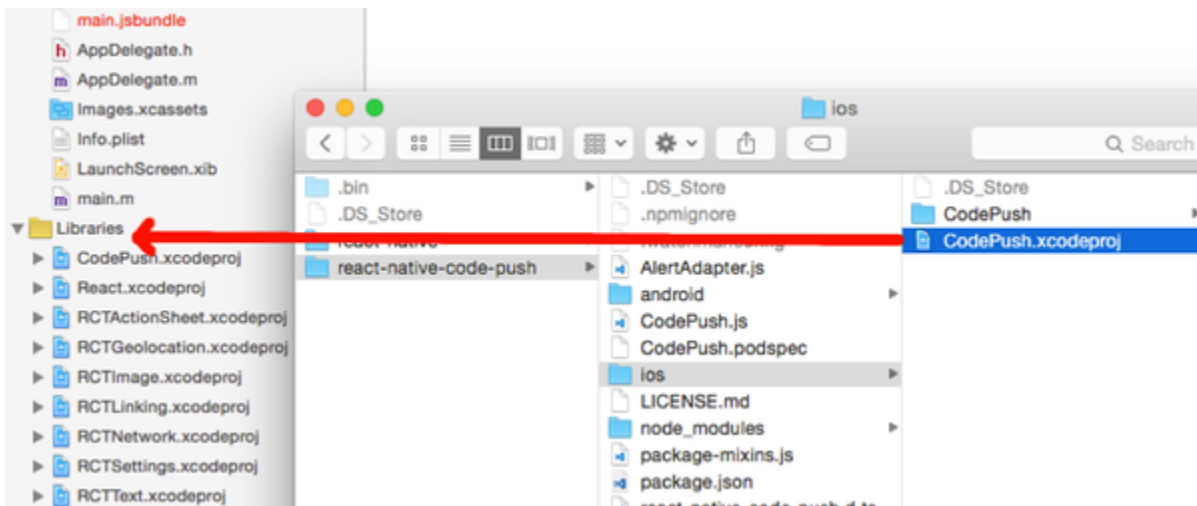
```
npm install --save react-native-code-push@latest
```

iOS Setup

Plugin Installation (IOS-Manual)

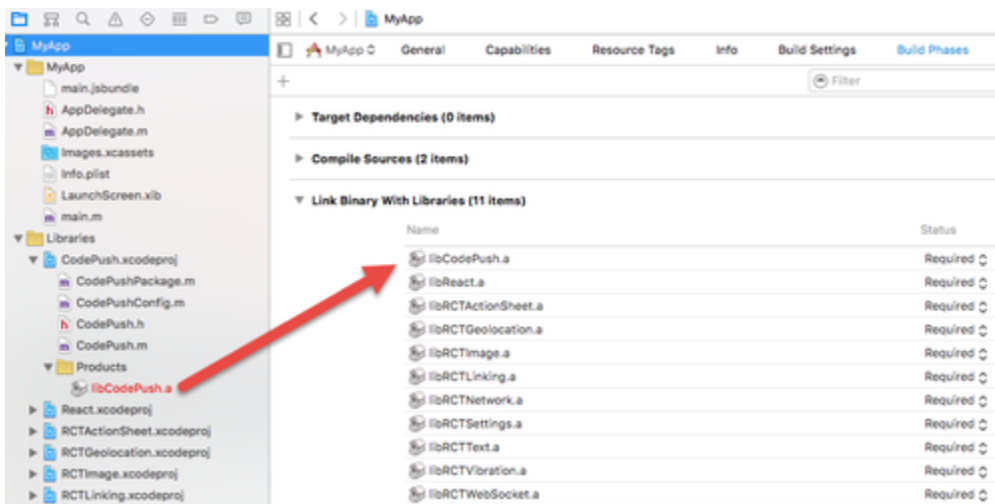
1. Open your app's Xcode project

2. Find the CodePush.xcodeproj file within the node_modules/react-native-code-push/ios directory (or node_modules/react-native-code-push for <=1.7.3-beta installations) and drag it into the Libraries node in Xcode

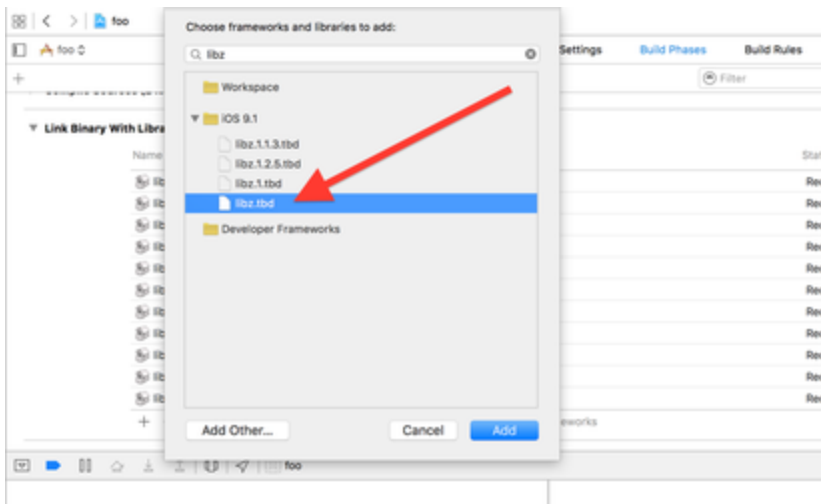


3. Select the project node in Xcode and select the “Build Phases” tab of your project configuration.

4. Drag libCodePush.a from Libraries/CodePush.xcodeproj/Products into the “Link Binary With Libraries” section of your project’s “Build Phases” configuration.



5. Click the plus sign underneath the “Link Binary With Libraries” list and select the libz.tbd library underneath the iOS 9.1 node.



Plugin Configuration (iOS)

1. Open up the AppDelegate.m file, and add an import statement for the CodePush headers:

```
#import <CodePush/CodePush.h>
```

2. Find the following line of code, which loads your JS Bundle from the app binary for production releases:

```
jsCodeLocation = [[NSBundle mainBundle] URLForResource:@"main" withExtension:@"jsbundle"];
```

3. Replace it with this line:

```
jsCodeLocation = [CodePush bundleURL];
```

To let the CodePush runtime know which deployment it should query for updates against, open your app's Info.plist file and add a new entry named CodePushDeploymentKey, whose value is the key of the deployment you want to configure this app against.

You can retrieve this value by running **code-push deployment ls <appName> -k** in the CodePush CLI (the -k flag is necessary since keys aren't displayed by default) and copying the value of the Deployment Key column which corresponds to the deployment you want to use

Name	Deployment Key	Package Metadata
Production	272982eb-b10d-4163-9044-cb940941bbc9	Label: v12 App Version: 1.0.0 Mandatory: Yes Hash: 4d455057570e84e8650d9af8d0dbff9311a2a1cd2fc05ea82aaf2fafbad1e154 Release Time: 5 days ago
Staging	d79d308a-4a12-48be-a479-9fd555143f3f	Label: v18 App Version: 1.0.0 Mandatory: No Hash: daf368a67ddc6ddad3f7bc1cb1571be895733b3ecc0ffbb51aceed8381d6b750 Release Time: 21 hours ago

Android Setup

Plugin Installation (Android- Manual)

1. In your android/settings.gradle file, make the following additions:

```
include ':app', ':react-native-code-push'
```

```
project(':react-native-code-push').projectDir = new File(rootProject.projectDir, '../node_modules/react-native-code-push/android/app')
```

2. In your android/app/build.gradle file, add the :react-native-code-pushproject as a compile-time dependency:

```
...
dependencies {
    ...
    compile project(':react-native-code-push')
}
```

3. In your android/app/build.gradle file, add the codepush.gradle file as an additional build task definition underneath react.gradle:

```
...
apply from: "../../node_modules/react-native/react.gradle"
apply from: "../../node_modules/react-native-code-push/android/codepush.gradle"
...
```

Plugin Configuration (Android)

For React Native >= v0.29

Update the MainActivity.java file to use CodePush via the following changes:

```
...
// 1. Import the plugin class.
import com.microsoft.codepush.react.CodePush;

public class MainActivity extends Application implements ReactApplication {

    private final ReactNativeHost mReactNativeHost = new ReactNativeHost(this) {
        ...
        // 2. Override the getJSBundleFile method in order to let
        // the CodePush runtime determine where to get the JS
        // bundle location from on each app start
        @Override
        protected String getJSBundleFile() {
            return CodePush.getJSBundleFile();
        }

        @Override
        protected List<ReactPackage> getPackages() {
            // 3. Instantiate an instance of the CodePush runtime and add it to the list of
            // existing packages, specifying the right deployment key. If you don't already
            // have it, you can run "code-push deployment ls <appName> -k" to retrieve your key.
            return Arrays.<ReactPackage>asList(
                new MainReactPackage(),
                new CodePush("deployment-key-here", MainActivity.this, BuildConfig.DEBUG)
            );
        }
    };
}
```

```
    }  
    }  
};  
}
```

Releasing Updates

```
code-push release-react <appName> <platform>
```

```
code-push release-react MyApp-iOS ios
```

```
code-push release-react MyApp-Android android
```

Rolling back Updates

A deployment's release history is immutable, so you cannot delete or remove an update once it has been released. However, if you release an update that is broken or contains unintended features, it is easy to roll it back using the rollback command:

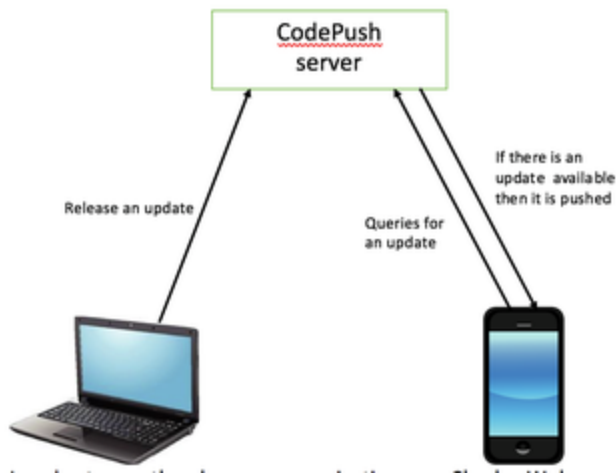
```
code-push rollback <appName> <deploymentName>
```

```
code-push rollback MyApp-iOS Production
```

If you would like to rollback a deployment to a release other than the previous, you can specify the optional `--targetRelease` parameter.

How it works

CodePush is a cloud service that enables Cordova and React Native developers to deploy mobile app updates directly to their users' devices. It works by acting as a central repository that developers can publish certain updates to (e.g. JS, HTML, CSS and image changes), and that apps can query for updates from (using our provided client SDKs). This allows you to have a more deterministic and direct engagement model with your end-users, while addressing bugs and/or adding small features that don't require you to re-build a binary and/or re-distribute it through any public app stores.



In order to see the above communication, use Charles Web debugging proxy. Refer the link <https://www.charlesproxy.com/documentation/> to get started with it

Following are the observations made:

1. Release an update for the app using the command `code-push release-react <appName> <platform>`
2. When the app resumes, it requests for the update. A request for an update is sent to the codePush server with deploymentKey of the app as one of the parameters.

Note: By default, CodePush will check for updates on every app start. If an update is available, it will be silently downloaded, and installed the next time the app is restarted. If you would like your app to discover updates more quickly, you can also choose to sync up with the CodePush server every time the app resumes from the background.

https://outlook.office365.com	URL	https://codepush.azurewebsites.net/updateCheck?deploymentKey=Mk8NU2JRFCzxV8addsvFCxT2Z_U4N15TtT7-X&appVersion=1.0&packageHash=4a82896f...
https://quicken.jamfcloud.com	Status	Complete
https://teams.microsoft.com	Response Code	200 OK
https://codepush.azurewebsites.net	Protocol	HTTP/1.1
reportStatus	SSL	TLSv1.2 (TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384)
updateCheck?deploymentKey=Mk8NU2JRFCzxV8addsvFCxT2Z_U4N15TtT7-X&appVersion=1.0&packageHash=4a82896f...	Method	GET
https://amer-client-ss.msg.skype.com	Kept Alive	No
https://pipe.skype.com	Content-Type	application/json; charset=utf-8
https://quicken.com	Client Address	/192.168.1.9
https://wfbs-svc-nabu-aal.trendmicro.com	Remote Address	codepush.azurewebsites.net/23.101.203.117
http://wfbs-svc30-p.activeupdate.trendmicro.com	Connection	
http://ocsp.apple.com		

3. If there is an update, then an update confirmation dialog appears.

Note: If you would like to display an update confirmation dialog (an "active install"), configure when an available update is installed (e.g. force an immediate restart) or customize the update experience in any other way, refer to the [codePush\(\)](#) API reference for information on how to tweak this default behaviour.



```
06-05 10:18:41.991 18195-18506/com.wixnavigation I/ReactNativeJS: active
06-05 10:18:42.002 18195-18506/com.wixnavigation I/ReactNativeJS: [CodePush] Checking for update.
06-05 10:18:42.037 18195-18226/com.wixnavigation W/unknown:PackagerConnectionSettings: You seem to be running on device. Run 'adb reverse tcp:8081 tcp:8081' to forward the debug server's
port to the device.
06-05 10:18:42.097 18195-18506/com.wixnavigation I/ReactNativeJS: [CodePush] Reporting binary update (1.0)
06-05 10:18:49.024 18195-18506/com.wixnavigation I/ReactNativeJS: [CodePush] Awaiting user action.
```

4. Once the user confirms for the installation, then the package is downloaded and installed immediately.

Note: If an update is available, it will be silently downloaded, and installed the next time the app is restarted (either explicitly by the end user or by the OS). However, you can modify the install behaviour, by leveraging the `installMode` parameter.

	Name	Value
▶ https://outlook.office365.com	URL	https://codepushupdates.azureedge.net
▶ https://quicken.jamfcloud.com	Status	Receiving response body...
▶ https://teams.microsoft.com	Notes	SSL Proxying not enabled for this host: enable in Proxy Settings, SSL locations
▼ https://codepush.azurewebsites.net	Response Code	-
▶ reportStatus	Protocol	HTTP/1.1
▶ updateCheck?deploymentKey=MK8NU2JF	SSL	TLSv1.2 (TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384) [http/1.1]
▶ <unknown>	Method	CONNECT
▶ https://amer-client-ss.msg.skype.com	Kept Alive	No
▶ https://pipe.skype.com	Content-Type	-
▶ https://quicken.com	Client Address	/192.168.1.9
▶ https://wfbs-svc-nabu-aal.trendmicro.com	Remote Address	codepushupdates.azureedge.net/23.65.99.94
▶ https://wfbs-svc30-p.activeupdate.trendmicro	Connection	
▶ http://ocsp.apple.com	▼ Timing	
▶ https://clients4.google.com	Request Start Time	5/6/17 16:28:12
▼ https://codepushupdates.azureedge.net	Request End Time	-
▶ <unknown>		

```

06-05 10:18:41.991 18195-18506/com.wixnavigation I/ReactNativeJS: active
06-05 10:18:42.002 18195-18506/com.wixnavigation I/ReactNativeJS: [CodePush] Checking for update.
06-05 10:18:42.037 18195-18226/com.wixnavigation W/unknown:PackagerConnectionSettings: You seem to be running on device. Run 'adb reverse tcp:8081 t:
port to the device.
06-05 10:18:42.097 18195-18506/com.wixnavigation I/ReactNativeJS: [CodePush] Reporting binary update (1.0)
06-05 10:18:49.024 18195-18506/com.wixnavigation I/ReactNativeJS: [CodePush] Awaiting user action.
06-05 10:19:01.566 18195-18230/com.wixnavigation D/OpenGLRenderer: endAllStagingAnimators on 0xabe442b8 (RippleDrawable) with handle 0xabe71a28
06-05 10:19:01.589 18195-18506/com.wixnavigation I/ReactNativeJS: [CodePush] Downloading package.
06-05 10:19:02.071 18195-18203/com.wixnavigation W/art: Suspending all threads took: 12.892ms
06-05 10:19:03.595 18195-18203/com.wixnavigation W/art: Suspending all threads took: 27.321ms
06-05 10:19:04.594 18195-18203/com.wixnavigation W/art: Suspending all threads took: 22.951ms
06-05 10:19:06.104 18195-18203/com.wixnavigation W/art: Suspending all threads took: 23.551ms
06-05 10:19:06.415 18195-18506/com.wixnavigation I/ReactNativeJS: [CodePush] Installing update.
06-05 10:19:06.614 18195-18505/com.wixnavigation W/unknown:PackagerConnectionSettings: You seem to be running on device. Run 'adb reverse tcp:8081 t:
port to the device.
06-05 10:19:06.614 18195-18225/com.wixnavigation W/unknown:PackagerConnectionSettings: You seem to be running on device. Run 'adb reverse tcp:8081 t:
port to the device.
06-05 10:19:06.671 18195-18507/com.wixnavigation D/ReactNative: [CodePush] Loading JS bundle from "/data/data/com
.wixnavigation/files/CodePush/b78bdd3f8168e13148642508e778f0c8bca0e416546a2c8059d1f0c811b888de/CodePush/index.android.bundle"
06-05 10:19:06.691 18195-18506/com.wixnavigation I/ReactNativeJS: [CodePush] Restarting app
06-05 10:19:06.826 18195-18506/com.wixnavigation I/ReactNativeJS: background
06-05 10:19:06.899 18195-18208/com.wixnavigation I/art: Background sticky concurrent mark sweep GC freed 34061(1779KB) AllocSpace objects, 13(734KB)
paused 3.209ms total 266.452ms
06-05 10:19:06.910 18195-18208/com.wixnavigation W/art: Suspending all threads took: 8.814ms
06-05 10:19:07.059 18195-18195/com.wixnavigation D/ReactNative: [CodePush] Loading JS bundle from "/data/data/com
.wixnavigation/files/CodePush/b78bdd3f8168e13148642508e778f0c8bca0e416546a2c8059d1f0c811b888de/CodePush/index.android.bundle"

```

How does CodePush Authenticate?

Most commands within the CodePush CLI require authentication, and therefore, before you can begin managing your account, you need to login using the GitHub or Microsoft account you used when registering. You can do this by running the following command:

```
code-push login
```

If you need to be able to authenticate against the CodePush service without launching a browser and/or without needing to use your GitHub and/or Microsoft credentials (e.g. in a CI environment), you can run the following command to create an "access key" (along with a name describing what it is for):

```
code-push access-key add "VSTS Integration"
```

After creating the new key, you can specify its value using the `--accessKey` flag of the `login` command, which allows you to perform "headless" authentication, as opposed to launching a browser.

```
code-push login --accessKey <accessKey>
```

How does CodePush prevent application code hijacking; i.e. pushing malicious code to users?

The source code uploaded to the Code Push Servers can only be accessed using your access key. Once uploaded, only meta data about the source code can change, not the code itself.

Muti-deployment Testing

CodePush plugin is configured using specific deployment key (You can retrieve this value by running `code-push deployment ls <appName> -k` in the CodePush CLI). Staging and Production deployments are auto-generated when a CodePush app is created for the first time. By using this, you never release an update to the end users that you haven't been able to validate yourself.

Following steps describes the workflow that can be achieved using Staging and Production deployments:

1. Release a CodePush update to your Staging deployment using the `code-push release-react` command (or `code-push release` if you need more control)

2. Run your staging/beta build of your app, sync the update from the server, and verify it works as expected
3. Promote the tested release from Staging to Production using the `code-push promote` command
4. Run your production/release build of your app, sync the update from the server and verify it works as expected

You can use `--deploymentName` to specify the deployment you want to update.

```
code-push release-react APP_NAME_HERE <platform> --deploymentName Production
```

You could also promote your release from staging to production ([docs](#)):

```
code-push promote APP_NAME_HERE Staging Production
```