React 18 & React Native

> ⚠ **CAUTION**
>
> This documentation is still **experimental** and details are subject to changes as we iterate. Feel free to share your feedback on the discussion inside the working group for this page.
>
> Moreover, it contains several **manual steps**. Please note that this won't be representative of the final developer experience once the New Architecture is stable. We're working on tools, templates and libraries to help you get started fast on the New Architecture, without having to go through the whole setup.

This page describes how to use React 18 with React Native using the React Native's New Architecture.

> **tl;dr:** The first version of React Native compatible with React 18 is **0.69.0**. In order to use the new features in React 18 including automatic batching, `startTransition`, and `useDeferredValue`, you must migrate your React Native app to the New Architecture.

React 18 and the React Native New Architecture

React 18 introduces several new features including:

- Automatic batching
- New Strict Mode behaviors
- New hooks (`useId`, `useSyncExternalStore`)

It also includes new concurrent features:

- `startTransition`
- `useTransition`
- `useDeferredValue`

- Full Suspense support

The concurrent features in React 18 are built on top of the new concurrent rendering engine. Concurrent rendering is a new behind-the-scenes mechanism that enables React to prepare multiple versions of your UI at the same time.

Previous versions of React Native built on the old architecture **cannot** support concurrent rendering or concurrent features. This is because the old architecture relied on mutating the native trees, which doesn’t allow for React to prepare multiple versions of the tree at the same time.

Fortunately, the New Architecture was written bottom-up with concurrent rendering in mind, and is fully compatible with React 18. This means, in order to upgrade to React 18 in your React Native app, your application needs to be migrated to the React Native's New Architecture including Fabric Native Components and Turbo Native Modules.

React 18 enabled by default

Starting in React Native 0.69, React 18 is **enabled by default** when you enable the New Architecture.

This means you’re able to use the new features in React 18 as soon as you migrate. Since the new concurrent features are opt-in by using features like `startTransition` or `Suspense`, we expect React 18 to work out-of-the-box with minimal changes for users who migrate to the New Architecture or create a new app with the New Architecture enabled.

However, if you do hit any issues, we provide an option to opt-out of the new root in React 18. Opt-ing out means your app will run in React 17 mode, and none of the features of React 18 will be accessible.

Opt-ing out of React 18 on Android

On Android, you will be able to override the `isConcurrentRootEnabled` in your ActivityDelegate (in the `MainActivity` file), and enable/disable Concurrent React.

Java Kotlin

```
public class MainActivity extends ReactActivity {

  public static class MainActivityDelegate extends ReactActivityDelegate {
    public MainActivityDelegate(ReactActivity activity, String mainComponentName) {
      super(activity, mainComponentName);
    }

    @Override
    protected ReactRootView createRootView() {
      ReactRootView reactRootView = new ReactRootView(getContext());
      // If you opted-in for the New Architecture, we enable the Fabric Renderer.
      reactRootView.setIsFabric(BuildConfig.IS_NEW_ARCHITECTURE_ENABLED);
      return reactRootView;
    }

+   @Override
+   protected boolean isConcurrentRootEnabled() {
+     // If you opted-in for the New Architecture, we enable Concurrent Root (i.e. React 18).
+     // More on this on https://reactjs.org/blog/2022/03/29/react-v18.html
+     return BuildConfig.IS_NEW_ARCHITECTURE_ENABLED;
+   }
  }
}
```

Opt-ing out of React 18 on iOS

On iOS, you'll have access to the `concurrentRootEnabled` method on your `AppDelegate.mm` file. You should change the returned value to `false` (or `NO`) to disable the feature.

```
/// This method controls whether the `concurrentRoot`feature of React18 is turned on or off.
///
/// @see: https://reactjs.org/blog/2022/03/29/react-v18.html
/// @note: This requires to be rendering on Fabric (i.e. on the New Architecture).
/// @return: `true` if the `concurrentRoot` feature is enabled. Otherwise, it returns `false`.
- (BOOL)concurrentRootEnabled
{
  // Switch this bool to turn on and off the concurrent root
```

```
    return true;
  }
```

Users on React Native 0.69 not yet migrated to the New Architecture

Note: Users on React Native 0.69, but still on the Old Architecture will still use React 17 mode even if React 18 is installed in the `package.json` file.

Overriding the `isConcurrentRootEnabled` method will have no effect on your app.

Is this page useful? 👍 👎

Edit this page

Last updated on ***Jun 21, 2023***