

**EXPERIMENTAL**

Hermes on Windows + macOS

[Edit](#)

Hermes

The Hermes engine is an open source JavaScript engine created by Facebook to optimize building and running React Native applications.

To learn more about what it is and how to use it, check out the React Native documentation for it.

Hermes on Windows:

Hermes is experimentally supported on Windows, generally providing better performance characteristics than the default Chakra engine.

Enabling Hermes for new projects

Pass the `--useHermes` flag to `react-native-windows-init` when creating a new project. This will set up your project to use the Hermes engine and to generate bundles as Hermes bytecode instead of JavaScript.



React Native for Windows + macOS 0.72

[Docs](#)[APIs](#)[Blog](#)[Resources](#)[Samples](#)[Support](#)

```
<PropertyGroup Label="Microsoft.ReactNative Experimental Features">
  ...
  <UseHermes>true</UseHermes>
</PropertyGroup>
```



Copy

Disabling Hermes

Set the `UseHermes` property to `false` in the `ExperimentalFeatures.props` file in your project's `windows` directory:

```
<PropertyGroup Label="Microsoft.ReactNative Experimental Features">
  ...
  <UseHermes>false</UseHermes>
</PropertyGroup>
```



Copy

Known limitations

- Hermes dll is not signed by Microsoft.

Hermes on macOS

Hermes is available on React Native for macOS, provided you are using version 0.62 or higher. To learn how to upgrade to the latest version, check out the **Upgrading** section of the [macOS Getting Started guide](#).

After you have upgraded to the latest version of React Native for macOS, install and add the following:

1. Install the npm package `yarn add 'hermes-engine-darwin@^0.4.3'`



React Native for Windows + macOS 0.72

[Docs](#)[APIs](#)[Blog](#)[Resources](#)[Samples](#)[Support](#)

```

pod 'react-native-hermes', :path => '../node_modules/react-native-macos/
pod 'hermes', :path => '../node_modules/hermes-engine-darwin'
pod 'libevent', :podspec => '../node_modules/react-native-macos/third-party-podspecs/libevent'

```

3. Run `pod install`

Be sure to set your target's deployment target to at least 10.14 before running `pod install`

Hermes Inspector

React Native for Windows with *Hermes engine* supports direct JavaScript runtime inspection using tools such as Chrome or [Edge devtools](#), [VSCode debugger](#), [Flipper](#) etc. by implementing an in-process Chrome Debug Protocol server. Please note that it is fundamentally different from "Remote JS Debugging", which loads the JavaScript bundle into a remote Chrome browser session with duplex communication over IPC channels.

We share the implementation (code and design) with other platforms wherever possible. All the external endpoints, APIs and protocols should be identical to *React Native* environments on other platforms. Hence, we expect most tooling available on other platforms to just work on Windows. But, as of now, we have tested only with Chrome and [Edge devtools](#).

Steps to enable direct debugging

1. Initialize React Native Host,
 - Turn on `DeveloperSupport`
 - Turn on `FastRefresh`
 - Turn off `WebDebugger`
 - Turn on `Direct Debugging`



4. Navigate to `edge://inspect` in Edge browser or `chrome://inspect` in Chrome browser
5. Enable **Discover network targets** and **configure** the target discovery settings to include `localhost:8081` (or wherever the metro server is running)
6. Within a few seconds "Hermes React Native" should appear on the page as a remote target
7. Click on the **inspect** link to launch the `devtools` page
8. Click `ctrl+P` to open source files and set break points
9. Alternatively, you can insert `debugger` statements in source code to break on specific locations

In order to break on locations during boot, you can either

- Add statements into the boot sequence to pause the runtime waiting for debugger to connect.

```
debugger;
```



Copy

- Set a break point and refresh the bundle through the Dev Server. The runtime will wait for debugger to attach.

Steps to enable heap profiling

Follow steps 1-7 from above, and then

1. Click on the "Memory" tab in the inspector
2. Heap snapshots and instrumented allocations should be working.

Enable debugging/profiling on release builds



React Native for Windows + macOS 0.72

[Docs](#)[APIs](#)[Blog](#)[Resources](#)[Samples](#)[Support](#)

```
npx react-native run-windows --msbuild EnableHermesInspectorInReleaseFlavor=true
```



Copy

Known Issues

1. CPU Sampling profiler currently doesn't work.

< Metro config for out of tree platforms

Using react-native-windows NuGet packages >

REACT NATIVE DOCS

[Getting Started](#)[Tutorial](#)[Components and APIs](#)[More Resources](#)

REACT NATIVE FOR WINDOWS + MACOS DOCS

[Get Started with Windows](#)[Get Started with macOS](#)[React Native Windows Components and APIs](#)[Native Modules](#)[Native UI Components](#)

CONNECT WITH US ON

[Blog](#)[Twitter](#)[GitHub](#)[Samples](#)