React Native for Windows + macOS   0.72

Docs   APIs   Blog   Resources   Samples   Support

THE BASICS (WINDOWS)

Publishing a React Native Windows App to the Microsoft Store

Edit

Steps to Publish a React Native Windows App to the Microsoft Store

These are the steps to follow if you are looking to publish a React Native Windows app as a third party to the Microsoft Store. Since React Native Windows apps are Universal Windows Platform (UWP) apps, you can also see Publish Windows apps - UWP applications | Microsoft Docs for all kinds of documentation on the UWP app publishing process.

1. Sign-up for a developer account through the Microsoft Partner Center (Register as an app developer – Windows app development). For more guidance on the sign-up process see here: Opening a developer account - UWP applications | Microsoft Docs.

2. Sign into the Partner Center. You should now be able to reserve a name for your app (Create your app by reserving a name - UWP applications | Microsoft Docs).

3. Open your app's solution in Visual Studio.

4. If you haven't already, generate visual assets for your app. You can generate visual assets from a base image in Visual Studio by opening your app's

⚛️ React Native for Windows + macOS   0.72

5. Right-click on your app project and select `Publish > Associate App with the Store…` . After you've signed-in (make sure you sign-in with the same account as the one used for the Partner Center) and selected your app, Visual Studio will update your `Package.appxmanifest` file with the new Publisher (a GUID) and the app name. You'll also get a new `Package.StoreAssociation.xml` file in your project that should not be checked in (Packaging MSIX apps - MSIX | Microsoft Docs).

6. Set your solution configuration to 'Release'. Right-click on your app project and choose `Publish > Create App Packages` . If you have already associated your project with an app in the Microsoft Store, you should have an option to 'Create Packages for the Associated Store App'. Fill out the form according to your app's information. This process should create an `.appxupload` , which you will upload to Partner Center later (Packaging MSIX apps - MSIX | Microsoft Docs).

7. Run your app through the latest version of the Windows App Certification Kit, and verify that it passes all tests (Windows App Certification Kit - UWP applications | Microsoft Docs).

8. Select your newly reserved app name from the 'Overview' section in Partner Center to begin adding information to your app.

9. Locate the 'Start a Submission' button on the current webpage. Hitting this button will open up the form for your first submission to the app store.

10. The app submission form will contain many questions about your app's age ratings, properties, pricing, etc. Click on each of the sections to fill in the information for your app (App submissions - UWP applications | Microsoft Docs):

- Pricing and Availability (Set app pricing and availability - UWP applications | Microsoft Docs)

- Properties (Enter app properties - UWP applications | Microsoft Docs)

- Age Ratings (Age ratings - UWP applications | Microsoft Docs)

- Store Listings (Create app Store listings - UWP applications | Microsoft Docs)

- Submission Options (Manage submission options - UWP applications | Microsoft Docs)

React Native for Windows + macOS   0.72

- Successfully passes WACK.

- Has personalized visual assets.

- Has a Display Name that matches the app name you have reserved.

- Created from the same developer account as the account where the submission is being made (Upload app packages - UWP applications | Microsoft Docs).

12. Once your submission form is fully filled out, you can submit your app to the Microsoft Store. It usually takes a couple of hours to run through certification and validation, and then possibly another couple hours for it to actually appear in the app store. (The app certification process - UWP applications | Microsoft Docs).

‹ Setup Continuous Integration Pipeline for an RNW App

Native Modules ›

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