V 6.2

youi-tv run

PAGE CONTENTS

`youi-tv run` generates, builds, deploys the app on the target platform, and launches the app on the target device, all in one command. While you can do the same thing by running `youi-tv build` and then following our instructions for deploying and launching an app, this command is a handy shortcut. It's really useful for testing your app on multiple target platforms.

Note `youi-tv run` is currently not supported on Roku.

Here's a more detailed look at the phases of `youi-tv run`:

> **generate:** builds a framework of files for your project, and depending on the type of target platform, also builds the related IDE project file — for example, a project you can open in Xcode or Visual Studio. This phase is only run if the required project files for the requested target platform haven't been generated yet.
>
> **build:** builds a debug or release binary executable for your application
>
> **deploy:** deploys the app to the target platform for testing during the development cycle
>
> **launch:** launches the app on the target device

Usage

Prerequisite: Connect a device to your development environment, or start an emulator. See also Deploying You.i Engine One Apps.

Execute the following from a shell in your project folder:

```
youi-tv run -p <platform> [options]
```

where `platform` is the target platform. When you execute `youi-tv run --help`, the description for `--platform` displays the target platforms that are valid from your current development environment.

For example:

`youi-tv run -p ps4 --device 10.100.xx.xx`

Options

General Options

Argument	Description
`-h, --help`	Output help information.
`-V, --version`	Output the CLI version number.

Options Related to Generating Your Project

Argument	Description

Argument	Description
`-b, --build_directory <directory>`	Where to place generated project files. Default: `<app>;/build/<platform>`, or `<app>/youi/build/<platform>` for React Native apps.
`--bundler_configuration <file>` **	Pass a configuration file to the Metro bundler, for example, `path/to/config.js`
`--clean`	Remove old project files before generating.
`-c, --config <configuration>`	The configuration type to send to the generator: `Debug` or `Release`. Default: `Debug`. Required only for generators that don't support multiple configurations. Applied to the generate stage or the build stage, depending on the generator.
`-d, --define <name>=<value>`	Pass a defined variable and its value to CMake. You can specify multiple instances of this option.
`--dev` **	Enable warnings. If you specify this option, you must also specify the `--local`option.
`--directory <directory>` **	Add JavaScript files from the specified directory, and create a bundle for each one.
`--file <file>` **	The path to the app's root JavaScript file. This option overrides any value set in the project's package.json file.
`-g, --generator <generator>`	The generator to use. If not specified, the default generator for the target platform is used. You.i TV officially supports the following generators for each target platform (note that we've listed them exactly as you need to specify them on the command line): **Android:** `AndroidStudio` (default) **OSX, iOS, tvOS:** `Xcode`(default) **Linux, webOS3, webOS4:** `Ninja`(default if installed), `Unix Makefiles` (default if Ninja isn't installed) **PS4:** `Visual Studio 16 2019 -AORBIS` (default). See also the PS4 notes in Supported Target Platforms. **Tizen-NaCl:** `Eclipse CDT4 - Ninja` (default if Ninja is installed), `Eclipse CDT4 - Unix Makefiles` (default if Ninja isn't installed) **win64, UWP:** `Visual Studio 16 2019`
`--inline` **	Compile JavaScript bundles directly into source code instead of fetching them from a yarn server.
`--iterate` **	Rebundle only those JavaScript bundles that need to be updated due to changes to their source JavaScript file. Note that a change to a JavaScript file that is a dependency of the source JavaScript file will not cause the file to be rebundled. If omitted, all JavaScript bundles are deleted and then recreated.
`--local` **	Package JavaScript bundles with the app, instead of fetching them from a yarn server. To deploy the production version of your Cloud Solution app on AWS, you **must** use the `--local` build argument.
`--minify` **	Minify the bundle file by excluding information such as newline characters. By default, minification is disabled, to speed up production builds for testing purposes.

Argument	Description
`-p, --platform <platform>`	Required. The target platform. Supported platforms are: android, ios, linux, osx, ps4, tizen-nacl, tvos, uwp, webos3, webos4, win64 When you execute `youi-tv run --help`, the description for `--platform` displays the target platforms that are valid from your current development environment.
`--ram_bundle` **	Package JavaScript bundles as "Random Access Modules" for offline use. To learn more, see Optimizing Start-up Performance Using RAM Bundling.
`--remote <IP_address>` **	Default if `--local` is not used. When building from your local development platform, localhost is used; otherwise, your development platform's IP address is used. Override this behavior by specifying the IP address for the Metro bundler; JavaScript bundles are then fetched from that bundler.
`--sourcemap_output <file>` **	Create a source map for the resulting bundle with the specified file name, for example: `/tmp/index.map`. This option overrides any value set in the project's package.json file.
`--youi_version <version>`	Generate against a specific version of the You.i Engine. Can be a path, such as `/path/to/5.0`, or semantic version, such as `5.0.1`.

** Applicable to React Native builds only

Options Related to Building Your App

Argument	Description
`-a, --arg <build_argument>`	Custom argument to pass to CMake when building. These arguments are passed down to the native build system. For example, '-j8' passed to Makefiles will run the build with 8 threads. Multiple arguments can be specified.
`-t, --target <target>`	`CopyAssets`: Copies the assets from the project's AE assets directory to the location required by the specified target platform for execution.</li><li> `CleanAssets`: Cleans up the assets which were copied by the CopyAssets target.</li><li> `Package`: Packages the application for the specific platform. This target is only available for platforms which require a packaged application: iOS, tvOS, PS4, webOS 3, and webOS 4.

Options Related to Deploying and Launching Your App

Argument	Description
`--arch <architecture>`	Android only (optional): Override the connected device's architecture. Useful only for apps that have been built into split Android Package Kits (APKs).

Argument	Description
`--device <IP_or_device-ID>`	Android: Optional iOS: Optional
`--pin <VS_pin>`	UWP only: Specify the Visual Studio pin for connecting to Xbox One. Required only the first time you run the command.