

RPG Maker MV/MZ Cook Tool

Version R4.0 – User's Manual

Intro

This tool allows developers to compile their game's source code to binary files that NW.js understands. This adds a layer of protection to the game, preventing people from modifying or stealing the source code. It can also be used to re-enforce the game's encryption.

Please note: Compiling the code will break the cross-platform capability. Meaning that in order to publish to other platforms, you'll have to re-compile the code, using the same version of the NW.js SDK for the platform you also want to publish to.

Getting Started

Preparing the game

RPG Maker MV

In order for the game to work with the compiled code, you'll need to edit the HTML and rpg_managers.js files.

In the index.html file, replace the <script> portion with this:

```
<script>nw.window.get().evalNWBin(null, "www/js/libs/pixi.bin");</script>
<script>nw.window.get().evalNWBin(null,
"www/js/libs/pixi-tilemap.bin");</script>
<script>nw.window.get().evalNWBin(null,
"www/js/libs/pixi-picture.bin");</script>
<script>nw.window.get().evalNWBin(null, "www/js/libs/fpsmeter.bin");</script>
<script>nw.window.get().evalNWBin(null, "www/js/libs/lz-string.bin");</script>
<script>nw.window.get().evalNWBin(null, "www/js/libs/iphone-inline-
video.browser.bin");</script>
<script>nw.window.get().evalNWBin(null, "www/js/rpg_core.bin");</script>
<script>nw.window.get().evalNWBin(null, "www/js/rpg_managers.bin");</script>
<script>nw.window.get().evalNWBin(null, "www/js/rpg_objects.bin");</script>
<script>nw.window.get().evalNWBin(null, "www/js/rpg_scenes.bin");</script>
<script>nw.window.get().evalNWBin(null, "www/js/rpg_sprites.bin");</script>
<script>nw.window.get().evalNWBin(null, "www/js/rpg_windows.bin");</script>
<script>nw.window.get().evalNWBin(null, "www/js/plugins.bin");</script>
<script>nw.window.get().evalNWBin(null, "www/js/main.bin");</script>
```

In the rpg_managers.js, you'll need to edit two parts. In line 2805 replace it with this:

```
this.loadScript(plugin.name + '.bin');
```

Then, replace lines 2826 until the ending bracket with this code:

```
PluginManager.loadScript = function(name) {
    nw.window.get().evalNWBin(null, "www/js/plugins/" + name);
};
```

Make sure to change the file extension with the one you want.

RPG Maker MZ

RPG Maker MZ requires a bit more work for the code to work. Mainly due to a different structure.

First, Open the index.html file and change the line:

```
<script type="text/javascript" src="js/main.js"></script>
```

To

```
<script>nw.window.get().evalNWBin(null, "js/main.bin");</script>
```

Make sure to change the .bin to the file extension you want.

Then, open the main.js file (found on the js folder). Change the following:

- In the scriptUrls the file extensions from JS to whatever you like.
- Change lines 50 to 55 (testXhr) to this:

```
testXhr() {  
    this.xhrSucceeded = true;  
}
```
- Change lines 57 to 72 (loadScripts) to this:

```
loadMainScripts() {  
    try {  
        for (const url of scriptUrls) {  
            nw.window.get().evalNWBin(null, url);  
        }  
        PluginManager.setup($plugins);  
    }  
    catch {  
        this.printError("Failed to load", url);  
    }  
    window.addEventListener("load",  
this.onWindowLoad.bind(this));  
    window.addEventListener("error",  
this.onWindowError.bind(this));  
}
```

To finish up, open rmmz_managers.js. Do the following:

- Change lines 3065 to 3075 (PluginManager.loadScripts) to this:

```
PluginManager.loadScript = function(filename) {  
    nw.window.get().evalNWBin(null, this.makeUrl(filename));  
};
```
- In the line 3077 (PluginManager.makeUrl) change ".js" to the file extension you want.

Compiling the project

Now that the project's ready, it's time to compile. First, make sure that the path to the NW.js SDK is set. Next, click on Add and pick the project's folder. If you want to edit some details, use the "Project Settings" and "Edit Metadata" to adjust them. You can add more projects you want to compile if you need to. When ready, click on "Compile". The tool will start compiling the game's source code, using the NW.js SDK. After a few minutes, your project is ready.

Project Settings

The "Project Settings" button will open up the settings for the selected project. These allow you to adjust some settings for the compiler.

Name	Function
Delete Javascript files after compilation	Removes the JS files from the project's folder. Recommended for deployment.
Package the game to a package.nw file	Compresses the game files into one archive called package.nw. This is helpful for reducing the game's size at the cost of startup time (since it has to decompress the archive first).
Compression Mode	Adjust the compression ratio. Optimal gives the best compression but it requires more time to decompress. No compression gives the fastest startup time, but the game's size won't change.
Remove source files after packaging	After packaging, the game files are deleted. Recommended for deployment, if you want to compress the game files.
File Extension	Sets the file extension for the binary files.

Edit Metadata

The "Edit Metadata" button will open up the Game Package Metadata Editor. This allows you to adjust the settings that are related to NW.js and the game. Stuff such as the game's resolution, the location of the index.html file, Chromium Flags and more are available.

Tool License

MIT License

Copyright (c) 2019 Studio ACE

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR

COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.