Product ⌄   Solutions ⌄   Resources ⌄   Open Source ⌄   Enterprise ⌄   Pricing

Sign in   Sign up

electron / **rcedit**   Public

🔔 Notifications   Fork 190   ☆ Star 1.7k

<> Code   Issues 30   Pull requests 2   ▶ Actions   Security   Insights

main ⌄

dependabot[bot]   chore: bump actions/checkout ...   ⋯   ✓   e36b688 · yesterday   🕐 184 Commits

📁 .github	chore: bump actions/checkout from 4.2...	yesterday
📁 src	chore: restore help version format (#12...	last year
📄 .editorconfig	Add .editorconfig with two-space inden...	9 years ago
📄 .gitignore	feat!: build with CMake and release wit...	last year
📄 .releaserc.json	chore: migrate default branch to main (...	last year
📄 CMakeLists.txt	feat!: build with CMake and release wit...	last year
📄 LICENSE	Initial commit.	11 years ago
📄 README.md	feat!: build with CMake and release wit...	last year

📖 README   Code of conduct   ⚖ MIT license

rcedit

![Continuous Integration passing]

Command line tool to edit resources of exe file on Windows.

Executables

Prebuilt binaries can be found in the [GitHub releases](https://github.com/electron/rcedit/releases).

Building

To build you need CMake 3.15+ and Visual Studio 2015 or above.

1. Clone the repository
2. Create a build directory: `cmake -E make_directory build`
3. Change to the build directory: `cd build`
4. Make the CMake project: `cmake ..`
5. Build: `cmake --build .`

Docs

Show help:

```
$ rcedit -h
```

Set version string:

```
$ rcedit "path-to-exe-or-dll" --set-version-string "Comments" "This
```

About

Command line tool to edit resources of exe

`windows`   `c-plus-plus`   `rcedit`

📖 Readme
⚖ MIT license
Code of conduct
✴ Activity
Custom properties
☆ 1.7k stars
👁 51 watching
⑂ 190 forks

Report repository

Releases 5

🏷 **v2.0.0** `Latest`
on Nov 16, 2023

+ 4 releases

Packages

No packages published

Contributors 22

+ 8 contributors

Languages

● C++ 99.4%   ● CMake 0.6%

Use this option to change any supported properties, as described in the MSDN documentation [here](#)

Set file version:

```
$ rcedit "path-to-exe-or-dll" --set-file-version "10.7"
```

Set product version:

```
$ rcedit "path-to-exe-or-dll" --set-product-version "10.7"
```

Set icon:

```
$ rcedit "path-to-exe-or-dll" --set-icon "path-to-ico"
```

Set resource string:

```
$ rcedit "path-to-exe-or-dll" --set-resource-string id_number "new s
```

Set [requested execution level](#) (`asInvoker` | `highestAvailable` | `requireAdministrator`) in the manifest:

```
$ rcedit "path-to-exe-or-dll" --set-requested-execution-level "requi
```

Set [application manifest](#):

```
$ rcedit "path-to-exe-or-dll" --application-manifest ./path/to/manif
```

And you can change multiple things in one command:

```
$ rcedit "path-to-exe-or-dll" --set-icon "path-to-ico" --set-file-ve
```

Get version string:

```
$ rcedit "path-to-exe-or-dll" --get-version-string "property"
```

Use the same properties as `--set-version-string` . Use `"FileVersion"` to get the results of `--set-file-version` and `"ProductVersion"` to get the results of `--get-product-version` .

Get resource string:

```
$ rcedit "path-to-exe-or-dll" --get-resource-string id_number
```