

# Electron Releases

Electron frequently releases major versions alongside every other Chromium release. This document focuses on the release cadence and version support policy. For a more in-depth guide on our git branches and how Electron uses semantic versions, check out our [Electron Versioning](#) doc.

## Timeline

Electron	Alpha	Beta	Stable	Chrome	Node	Supported
2.0.0	--	2018-Feb-21	2018-May-01	M61	v8.9	🚫
3.0.0	--	2018-Jun-21	2018-Sep-18	M66	v10.2	🚫
4.0.0	--	2018-Oct-11	2018-Dec-20	M69	v10.11	🚫
5.0.0	--	2019-Jan-22	2019-Apr-24	M73	v12.0	🚫
6.0.0	--	2019-May-01	2019-Jul-30	M76	v12.4	🚫
7.0.0	--	2019-Aug-01	2019-Oct-22	M78	v12.8	🚫
8.0.0	--	2019-Oct-24	2020-Feb-04	M80	v12.13	🚫
9.0.0	--	2020-Feb-06	2020-May-19	M83	v12.14	🚫
10.0.0	--	2020-May-21	2020-Aug-25	M85	v12.16	🚫
11.0.0	--	2020-Aug-27	2020-Nov-17	M87	v12.18	🚫
12.0.0	--	2020-Nov-19	2021-Mar-02	M89	v14.16	🚫
13.0.0	--	2021-Mar-04	2021-May-25	M91	v14.16	🚫
14.0.0	--	2021-May-27	2021-Aug-31	M93	v14.17	🚫
15.0.0	2021-Jul-20	2021-Sep-01	2021-Sep-21	M94	v16.5	✅
16.0.0	2021-Sep-23	2021-Oct-20	2021-Nov-16	M96	v16.9	✅
17.0.0	2021-Nov-18	2022-Jan-06	2022-Feb-01	M98	v16.13	✅
18.0.0	2022-Feb-03	2022-Mar-03	2022-Mar-29	M100	v16.13	✅
19.0.0	2022-Mar-31	2022-Apr-26	2022-May-24	M102	TBD	✅

### Notes:

- The `-alpha.1`, `-beta.1`, and `stable` dates are our solid release dates.
- We strive for weekly alpha/beta releases, but we often release more than scheduled.
- All dates are our goals but there may be reasons for adjusting the stable deadline, such as security bugs.

### Historical changes:

- Since Electron 5, Electron has been publicizing its release dates ([see blog post](#)).
- Since Electron 6, Electron major versions have been targeting every other Chromium major version. Each Electron stable should happen on the same day as Chrome stable ([see blog post](#)).

- Since Electron 16, Electron has been releasing major versions on an 8-week cadence in accordance to Chrome's change to a 4-week release cadence ([see blog post](#)).

...info Chrome release dates

Chromium has the own public release schedule [here](#).

...

## Version support policy

...info

Beginning in September 2021 (Electron 15), the Electron team will temporarily support the latest **four** stable major versions. This extended support is intended to help Electron developers transition to the [new 8-week release cadence](#), and will continue until the release of Electron 19. At that time, the Electron team will drop support back to the latest three stable major versions.

...

The latest three *stable* major versions are supported by the Electron team. For example, if the latest release is 6.1.x, then the 5.0.x as well as the 4.2.x series are supported. We only support the latest minor release for each stable release series. This means that in the case of a security fix, 6.1.x will receive the fix, but we will not release a new version of 6.0.x.

The latest stable release unilaterally receives all fixes from `main`, and the version prior to that receives the vast majority of those fixes as time and bandwidth warrants. The oldest supported release line will receive only security fixes directly.

## Breaking API changes

When an API is changed or removed in a way that breaks existing functionality, the previous functionality will be supported for a minimum of two major versions when possible before being removed. For example, if a function takes three arguments, and that number is reduced to two in major version 10, the three-argument version would continue to work until, at minimum, major version 12. Past the minimum two-version threshold, we will attempt to support backwards compatibility beyond two versions until the maintainers feel the maintenance burden is too high to continue doing so.

## End-of-life

When a release branch reaches the end of its support cycle, the series will be deprecated in NPM and a final end-of-support release will be made. This release will add a warning to inform that an unsupported version of Electron is in use.

These steps are to help app developers learn when a branch they're using becomes unsupported, but without being excessively intrusive to end users.

If an application has exceptional circumstances and needs to stay on an unsupported series of Electron, developers can silence the end-of-support warning by omitting the final release from the app's `package.json` `devDependencies`. For example, since the 1-6-x series ended with an end-of-support 1.6.18 release, developers could choose to stay in the 1-6-x series without warnings with `devDependency` of `"electron": 1.6.0 - 1.6.17`.