

Migrating to Meteor 2.2

Most of the new features in Meteor 2.2 are either applied directly behind the scenes (in a backwards compatible manner) or are opt-in. For a complete breakdown of the changes, please refer to the changelog.

The above being said, there are a few breaking changes that you might need to apply migration for.

Running MongoDB on Windows

`meteor-tool` has been updated and you might need to install the new Visual C++ Redistributable for Visual Studio 2019 to run MongoDB 4.4.4 on Windows. [read more](#)

MongoDB `useUnifiedTopology`

`mongo` package is now using `useUnifiedTopology` as `true` by default otherwise the new driver was producing a warning (see details below). It's important to test your app with this change.

Cordova 10

`cordova` plugins and main libraries were updated from 9 to 10. It's important to test your app with these changes.

Typescript 4.2.2

`typescript` was updated to 4.2.2, make sure you read the breaking changes.

Migrating from a version older than 2.0?

If you're migrating from a version of Meteor older than Meteor 2.0, there may be important considerations not listed in this guide (which specifically covers 2.0 to 2.2). Please review the older migration guides for details:

- [Migrating to Meteor 2.0 \(from 1.12\)](#)
- [Migrating to Meteor 1.12 \(from 1.11\)](#)
- [Migrating to Meteor 1.11 \(from 1.10.2\)](#)
- [Migrating to Meteor 1.10.2 \(from 1.10\)](#)
- [Migrating to Meteor 1.10 \(from 1.9.3\)](#)
- [Migrating to Meteor 1.9.3 \(from 1.9\)](#)
- [Migrating to Meteor 1.9 \(from 1.8.3\)](#)
- [Migrating to Meteor 1.8.3 \(from 1.8.2\)](#)

- Migrating to Meteor 1.8.2 (from 1.8)
- Migrating to Meteor 1.8 (from 1.7)
- Migrating to Meteor 1.7 (from 1.6)
- Migrating to Meteor 1.6 (from 1.5)
- Migrating to Meteor 1.5 (from 1.4)
- Migrating to Meteor 1.4 (from 1.3)
- Migrating to Meteor 1.3 (from 1.2)