

Migrating to Meteor 1.10.2

Most of the new features in Meteor 1.10.2 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 is a breaking change for those using the Flow syntax.

Flow syntax not supported

The `babel-compiler` package, used by both `ecmascript` and `typescript`, no longer supports stripping Flow type annotations by default, which may be a breaking change if your application (or Meteor package) relied on Flow syntax.

If you still need Babel's Flow plugins, you can install them with npm and then enable them with a custom `.babelrc` file in your application's (or package's) root directory:

```
{
  "plugins": [
    "@babel/plugin-syntax-flow",
    "@babel/plugin-transform-flow-strip-types"
  ]
}
```

Migrating from a version older than 1.10?

If you're migrating from a version of Meteor older than Meteor 1.10, there may be important considerations not listed in this guide (which specifically covers 1.10 to 1.10.2). Please review the older migration guides for details:

- 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)