Migrating to Meteor 1.10

Most of the new features in Meteor 1.10 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, some breaking changes to note and migration steps for a bug that you might encounter.

Unexpected mongo exit code 62

If you get `Unexpected mongo exit code 62. Restarting.` when starting your local MongoDB, you can either reset your project (`meteor reset`) (if you don't care about your local data) or you will need to update the feature compatibility version of your local MongoDB:

```
1. Downgrade your app to earlier version of Meteor `meteor update --release 1.9.2`
2. Start your application
3. While your application is running open a new terminal window, navigate to the
   app directory and open `mongo` shell: `meteor mongo`
4. Use: `db.adminCommand({ getParameter: 1, featureCompatibilityVersion: 1 })` to
   check the current feature compatibility.
5. If the returned version is less than 4.0 update like this:
   `db.adminCommand({ setFeatureCompatibilityVersion: "4.2" })`
6. You can now stop your app and update to Meteor 1.10.

For more information about this, check out [MongoDB documentation](https://docs.mongodb.com/
```

Cordova upgrade

Cordova has been updated from version 7 to 9. We recommend that you test your features that are taking advantage of Cordova plugins to be sure they are still working as expected.

WKWebViewOnly

WKWebViewOnly is set by default now as true so if you are relying on UIWebView or plugins that are using UIWebView APIs you probably want to set it as false, you can do this by calling `App.setPreference('WKWebViewOnly', false);` in your mobile-config.js. But we don't recommend turning this into false because Apple have said they are going to reject apps using UIWebView.

Windows 32-bit support dropped

Because MongoDB since 3.4 no longer supports 32-bit Windows, Meteor 1.10 has also dropped support for 32-bit Windows. In other words, Meteor 1.10 supports 64-bit Mac, Windows 64-bit, and Linux 64-bit.

Migrating from a version older than 1.9.3?

If you're migrating from a version of Meteor older than Meteor 1.9.3, there may be important considerations not listed in this guide (which specifically covers 1.9.3 to 1.10). Please review the older migration guides for details:

- 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)