# Title: "No. 13 is unlucky" - RCA for input/text area fields having a change of UI for L/C/A/D forms

## Deployment: 46.8 on April 13, 2018

Deployment:46.8 had a UI breakage with L/C/A/D forms, bulk update and lead convert forms, failing to render the FreshSales form elements such as text and textarea box. The other fields had worked perfect as it used to. Something like shown below.

![](data:text/html;base64,)

Interestingly, this happened in *staging*environment as well but we figured that out only post production deployment as it was all fine till 4PM. This resulted in our suspicion towards commits that came just before the deployment and we started reverting the code commits one by one. To our surprise, none of the commits before or after had caused this issue (when reset HEAD in local and frontend build). This led to the next part of the story where we wanted to confirm if the form service addon had an update, but this was outrightly rejected since we have locked freshdesk/ember-form addon to commit hash which means no new commits in the branch would still cause any adverse effect. Scanning the freshdesk/ember-form package.json, one suspicious part was where "ember-form-for" addon that's being consumed by the freshdesk/ember-form addon:

![](data:text/html;base64,)

We then digged into the production build logs of npm (frontend/build\_log\_npm.txt) which had the history of installations of the ember addons consumed by the app. Interestingly, the previous build had v2.0.0-beta1 (Previous evening build) and the current build had v2.0.0-beta2. It was almost confirmed that this was the reason why the UI broke, but to get enough data points and quickly resolve this issue, we locked the ember-form-for version in freshdesk/ember-form package.json to v2.0.0-beta1 and did a fresh frontend build. The UI was back up to old state. What caused the whole mumbo jumbo was that little caret ('^') symbol. This ('^') symbol was all enough to break FreshSales UI.

**Why this happened?**

ember-form-for had a recent tag release at 4:26PM IST (Apr 13th, 2018) that had the following changes. If you wonder, how this change affected us, in freshdesk/ember-form addon, we override form-field.hbs which intern render our form elements for our needs. Since beta2 in parent repo had changes to the way the classes are referred with an '\_' and the previous versions not having this '\_' caused the breakage. Apparently we override with inputClasses (< 2.0.0beta1) while the parent addon (ember-form) had changed the property name to something like **\_inputClasses.** As a result the classes that we set from config/environment.js were never used (inputClasses is now undefined).

**Parent ember addon:**[**https://github.com/martndemus/ember-form-for**](https://github.com/martndemus/ember-form-for)

![](data:text/html;base64,)![](data:text/html;base64,)

**Here's the overrider in freshdesk/ember-form repo:**

![](data:text/html;base64,)

**Version release of ember-form-for addon:**

![](data:text/html;base64,)

**Future plans:**

Ideally, all the addons that we consume should be version locked and not take such risks of allowing addons to update on their own during deployments/build. We had faced similar problems 2 years back with jQuery version that auto upgraded to 3.x version which caused the entire app to break. At the same time, make use of package-lock.json/yarn-lock.json and consume from a cached node\_modules / bower\_components so that frontend build will also be faster and no versions get upgraded automatically.

**PS: If you wonder why this din't happen in dev? Read more:**

In our dev setup, we run *ember build --watch* most of the times, or after we pull frontend code we just do an *ember build*. If there is a change in package.json, ember build prompts an error asking to do an npm install. Despite this, we wouldn't be still facing this issue since when an npm install is performed, only the changes that your package.json had, will be installed new if your node\_modules flat structure did not contain that version of the addon. In this case, there was no new package.json changes for freshdesk/ember-form addon. Which means we will still be consuming the last addon version that was installed inside node\_modules (If it was 2 years back, it would remain that unless you manually deleted node\_modules  ). [Refer here for more inputs on the npm installation algo.](https://docs.npmjs.com/cli/install#algorithm)

Why this happens in a stack/production is now simple to understand. In stack, we do fresh *npm install, bower install and ember build --prod* which has a fresh build ready to deploy. This is also one of the reasons why our frontend builds are slower. As a result, every npm install will be a fresh install of addons and if an addon contains a '^' then it's bound to take the last updated tag release from the npm repo. This is one of the reasons why we were unable to replicate this issue despite doing multiple commit resets (without deleting node\_modules). But when we did a clean installation of node\_modules in local, even the commits that went to production last night faced this problem - which was the initial clue for debugging this issue.

You might also wana checkout this blog from Azer Koçulu which shares a similar experience <https://www.theregister.co.uk/2016/03/23/npm_left_pad_chaos/>

#### Thanks for all the help to [Shaik Buden Saheb](https://confluence.freshworks.com/display/~shaikb) who found this issue while testing Deal forms. Big thanks to [praveenkumar.varadan@freshworks.com](https://confluence.freshworks.com/display/~praveenkumar.varadan@freshworks.com),[Bhagirath Goud](https://confluence.freshworks.com/display/~bhagirath.goud), [Robin Thomas](https://confluence.freshworks.com/display/~robin.thomas) and [Amritendu Mondal](https://confluence.freshworks.com/display/~amritendu.mondal) who pitched in to debug, validate, run test cases, UI automation tests and ensure version locking fix still didn't break the app post production deployment.