# Intro

Notes on migrating from a solution based on the “official SPA template” as it was on December 3th, 2012, to a Breeze version of that solution.

I am not trying to explain anything; I am merely recording the steps. I do begin this essay with some questions arising from the migration and testing.

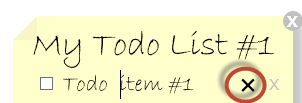
# Questions

## copyright

The template currently generates an AssemblyInfo with a Microsoft copyright. That doesn’t seem right. But if it is, should the Breeze template have an IdeaBlade copyright? Even though the whole thing is a mashup of different parts? We don’t really care … but this sample template was given an IdeaBlade copyright.

## second “x”

When editing a TodoItem, why does a second ‘x’ appear as shown here:



This seems to mean “clear entry” but it sure is confusing relative to the other, gray ‘x’ which means “delete”.

## both the original and the breeze solution share the database [???]

Although the database file name of the breeze solution is different from the original and the connection string in the breeze solution’s Web.config clearly references that solution’s own copy of the database, when I run both the original and the migrated breeze solutions, I discover that they are actually sharing the same database. Changes in one are made to the other.

However, when I run a solution on its own, it talks to its own database exclusively.

I am chagrined to admit that I cannot explain how this is possible. File date examination reveals that the migrated breeze solution is somehow updating the original solution’s db.

# Change

I made an exact copy of the official template solution and then …

* Prefixed the name of the solution, project folder, and csproj with “Breeze” so it now reads “BreezeMvcApplication7” (edited the .sln file in a text editor as part of this process).
* Globally changed every mention of “MvcApplication7” to “BreezeMvcApplication7”; this mostly affected namespaces.
* Made the same changes in the project properties, thus changing the assembly and default namespace names
* Changed the product name, company, and copyright in AssemblyInfo as described above.
* Changed port to 7902 in “Project | Properties | Web ! Local IIS Web Server | Project Url” so can run official template version and this Breeze template version side-by-side w/o conflict.
* Renamed the database in App\_Data to match “BreezeMvcApplication7” although, as noted above, this seems to make no difference at all. It’s a mystery to me.
* ***BundleConfig.cs*** – add a breeze bundle

bundles.Add(new ScriptBundle("~/bundles/breeze").Include(

"~/Scripts/q.js",

"~/Scripts/breeze.debug.js"));

* Include this breeze bundle in the @section scripts just before the “todo” bundle
* ***WebApiConfig.cs*** – comment out or delete “config.EnableQuerySupport();” because it conflicts with Breeze’s ODataActionFilter.   
  **Note that Breeze should do this itself for its controller**

# Delete

* Controllers/TodoController
* Controllers/TodoListController
* Models/TodoItemDto
* Models/TodoListDto
* Scripts/app/todo.model.js
* Scripts/app/todo.datacontext.js

# Add

* Breeze NuGet Pkg: “Breeze.MVC4WebApi” from NuGet official package source

**FAIL**: Something is very wrong with the project and its interaction with the Breeze NuGet pkg via the visual NuGet tool.

The Breeze pkg install works fine for a new MVC 4 Empty. But it hung every time I tried it (3x) when applied to **this solution**, always freezing during breeze “install.ps1”. There is no indication of what the problem is. I had to use Task Manager to kill VS every time. Not sure how to debug.

However, I was able to install it from Package Manager Console! So I soldiered on. Will revisit.

* Controllers/BreezeTodoController
* Models/TodoContextProvider
* Scripts/app/todo.model.js [Breeze version]
* Scripts/app/todo.datacontext.js [Breeze version]