# BLocal Setup Guide

## BLocal

### Installation

* Reference Localization(.dll)

### Recommended useage

* Check out the different providers that come with the package
* Pick a LocaleProvider, PartProvider, LocalizedValueProvider and NotificationProvider
  + Or roll your own
* Create a repository with instances of said selected providers
* Use this repository as needed

## BLocal.Web (basic)

### Installation

* Reference Localization(.dll)
* Reference Localization.Web(.dll)

### Recommended useage

* See Localization Recommended Useage
* You will need to manually pass your repository to every view
* Reference the Localization.Web from the web.config in your views folder
* Use var local = @Html.Local(\*your repository\*) to set up localization
* Use @local.AnyMethod instead of @Html.AnyMethod to build localized tags

## BLocal.Web (auto-resolve with IoC)

### Setup

* Install Localization.Web
* Set up your different providers with your IoC container
  + Example: kernel.Bind<IPartProvider>().ToConstant(partProvider).InSingletonScope();
* Make sure to set up injection for the LocalizationContext class and all of its dependencies
* If your IoC container does not do this automatically, register it with MVC
  + Example: DependencyResolver.SetResolver(yourResolver);
* Tip: check out MvcLocaleProvider, MvcPartProvider, MSDBValueAndLogProvider, VoidLogger
* Tip: you can set up dependency injection to inject different providers in different scenarios

### Result

* The LocalizationHelper will automatically resolve its repository
* The @Html.Local() method no longer needs its repository argument
* You no longer have to manually pass the repository to all your views
* This means you can now set up localization in the layout without worries

## BLocal.Web (javascript debugging)

### Setup

* Install Localization.Web
* Recommended: setup auto-resolve with IoC
* Create a controller that inherits MvcLocalizationController in your controller namespace
  + Recommended: name it LocalizationController to avoid extra arguments later
* Required scripts
  + jquery-1.9.1.js (or higher), reference this first
  + shortcut.js, reference this second
  + jquery.localization.js, do not reference this manually
* Recommended css
  + jquery.localization.css (or roll your own), reference this yourself
* Use the method @local.JsLink() to set it up
  + Attention: reference jquery.js and shortcut.js before calling this method!
  + Tip: you can do this in your from the layout to avoid redundancy
* Recommended: use the method @local.JsValueSummary() just before </body>
  + Will debug other values when pressing enter while no localized elements are active

### Result

* When debug mode is on (on the localizationcontext), will allow debugging
  + Hover over any localized HTML node and press enter to get started
  + Press escape to close
  + Conditional tags are always visible while in debug mode
  + Some values might not change immediately; reload page to confirm it worked

## BLocal.Web Tips & Tricks

* Use at least the IoC step when possible, it’s usually worth it.
* Set up localization in your layout
* Remember to use @using(local.BeginSubPart()) { } in really big pages
* Make use of @using(local.BeginPart()) { } or var otherLocal = @local.ForPart
  + You can have a static class containing parts that recur throughout your application
* Use the Localization.Web.Manager, it has a lot of cool functions!
* Use VoidLogger in production environments
* Using an EmailNotifier (or similar) can avoid problems in production environments
* MSDBValueProvider has an InsertDummy option which is amazing for development purposes
* If you model-bind the localization, don’t rename your model class too often ☺
* If you are discontent with something, roll your own!
* Stay on the latest version, it will only get better.

## BLocal.Web.Manager

### Setup

* Add configurations to the web.config as demonstrated
* Have fun!