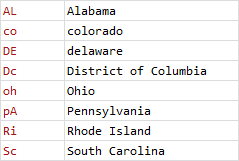
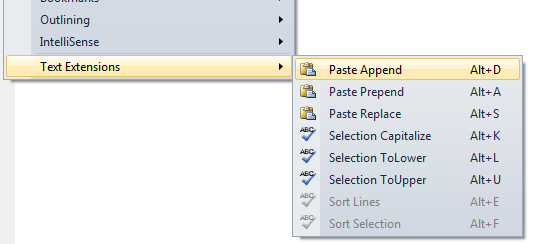
These commands are essentially copy and paste type commands that can be done against large regions of selected code blocks in the Visual Studio editor found under the “Edit” menu. Useful shortcuts have been made close to the copy and paste commands for easier keyboard accessibility during their use.

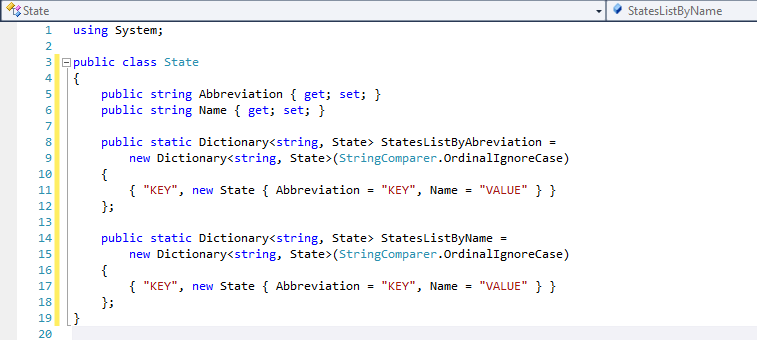
**States Example**

Say you have an Excel document with a list of states in random order. The document contains one column which holds the state abbreviations, and it contains another column which holds the state name ("AL", "Alabama"). These states and abbreviations were also pulled distinctly from a database table that contained data entry problems and the results had text formatting issues, improper capitalization, etc. The problem of time consumption is formed when a developer needs to create a dictionary in a C# class, of these states and abbreviations that is organized and tidy. The developer needs to keep the code concise and organized without spending a whole lot of time on it. Here is a sample of what might be seen:  
  


For me, I can look at something like this and not be frustrated at all, because I made a Visual Studio plugin just for reasons such as this. I just haven’t released it to the general population yet as I haven’t taken time to expand upon it. Using text extensions gives me the following options under the Edit menu in Visual Studio once the plugin has been installed by opening or double clicking the following file in the Text Extensions project directory “TextExtensions\TextExtensionsVsi\TextExtensions.vscontent”:

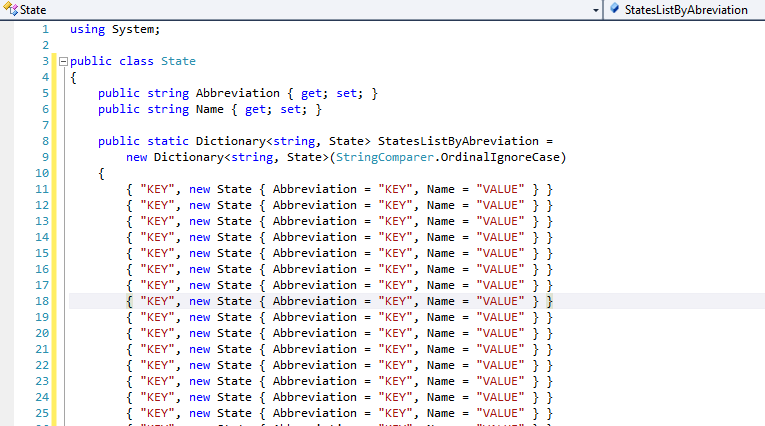


I will show you the basic use for Text Extensions momentarily with a created sample class for states:



Here we are saying that we have two dictionaries for lookups where the dictionary keys can be called against with any state or state name capitalization. So if we used the data to populate these classes from the excel sheet it would be totally fine to use the data as is.

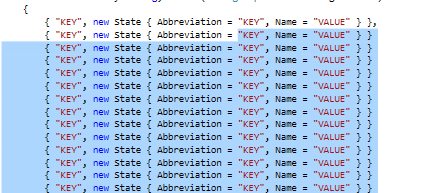
However, if we wanted to use the values in the class dictionaries in dropdowns and in other useful ways, we would at least want the data in the State class portion in our example to have proper capitalization. Also, if we had them in the correct order in the dictionary, then we would not have to sort the results of the dictionary values as well. This would require us to have the dictionary data instantiated with the states already sorted.

First we will want to create/duplicate as many dictionary entries that we need for the States class:  
  


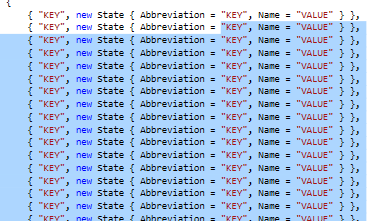
Now, we realize that we need a comma at the end of each line, because computers do exactly what we tell them to. This is simple if we type one comma at the top, select the comma, and then copy it using [ctrl + c].

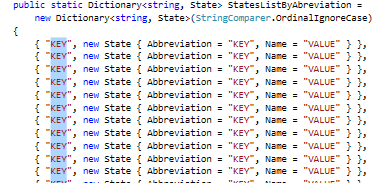


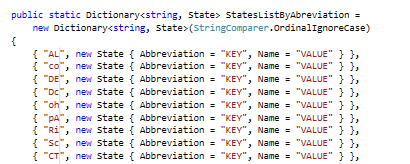
Lazily select the code block that you wish to paste the comma against.



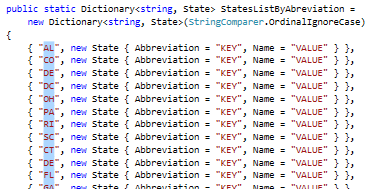
Press [alt + d] on the keyboard and it will keep your results selected as well as it adds the comma to the end of each line using the Paste Append command. If a mistake is made, it is okay. Text Extensions is using an Undo context.

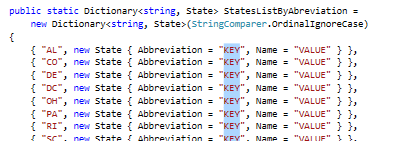


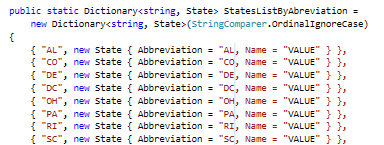
Now we wish to add in the state abbreviations for the dictionary. Copy these from the Excel document first and then highlight the “KEY” columns while holding [alt] key down to get into column selection mode in Visual Studio.  
  


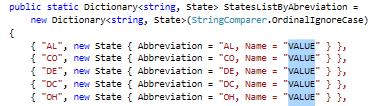
With your multi-line abbreviation column data selected from excel, it is time to press [alt + s] for an “inline” Paste Replace command. During the selection, the alt button never has to be released.  


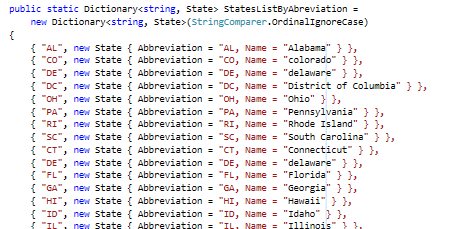
Quickly reselect the abbreviations again and press [alt + u] to make everything in the selection upper case using the Selection Capitalize command:

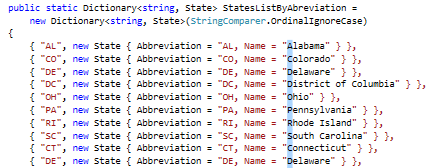


Press [ctrl + c] to copy again and then select the Abbreviation string values as shown while holding down the [alt] key for column selection mode:  
  


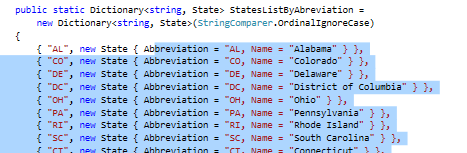
Press [alt + s] for the Paste Replace command.  
  


Go back to the Excel document and now select the state name column and copy the results. Swap back to the Visual Studio instance and column select the name string contents:  
  


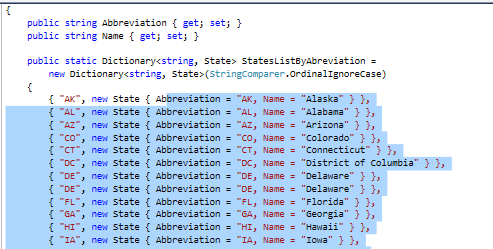
Press [alt + s] for the inline Paste Replace command. We now have our entire dictionary filled out. However, the first initial in some of the states is not capitalized.  
  


Column select, the first initial in each of the states and press [alt + u] to selection capitalize the state names.  
  


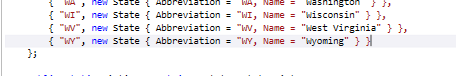
Now lazy select all of the state dictionary definitions again.



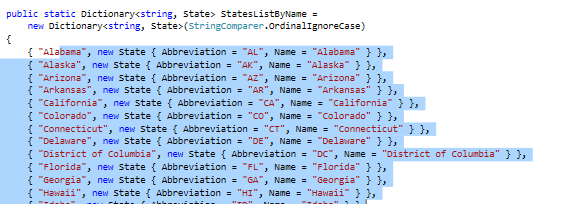
Press [alt + e] to sort all of the lines of code using the Sort Lines command.



Remove the comma from the end of the dictionary (or leave it, this is a personal choice).



For StateListByName dictionary, copy and paste in a similar manner using the Text Extensions commands. However, use the state name for the dictionary key. This way, the results are ordered by the state name key, and not the abbreviation key.

  
  
Welcome to cleaner coding, maintainability, and coding outside the box. Contact me at [ryanmauldin@msn.com](mailto:ryanmauldin@msn.com) with questions/ideas if needed.