To be honest I had to do some research into the inner workings of a dictionary for this project. My resources where from a very good book about .NET that I own, some snippets from Stack Overflow, and the actual source code for a .NET dictionary which I didn’t have time to fully dissect. It was a very interesting project and I actually learned quite a lot about dictionaries such as the fact that they use Hash tables so they are very fast.

I basically implemented a list with some generic key value pairs. They are strongly typed on instantiation and take any type of variable. I used Equals instead of the (==) operator and this does seem to compare objects as well as other variable types.

I added some try catch blocks that will throw an error from a custom error class but I did not implement the actual logging class for the errors which I would normally do in a situation when I have more time.

I added a custom Equals method that accounts for string case sensitivity but I probably could have done a better implementation for this.

I did look up some optimizing improvements you can make for a dictionary but since this project is implemented with a list it would really be some list optimizations I could have done. Some optimizations you can make for a list is setting the size of the list and using a for loop instead of foreach.

Some optimizations for a dictionary include increasing size instead of decreasing size. This allows for less hash collisions. You can also divide the dictionary into smaller pieces, read data from a dictionary in chunks. Shorting keys also helps. There are a handful of other dictionary optimization techniques that I read about here. –

<https://www.dotnetperls.com/optimization>

Here are some of the resources I used

<http://www.dotnetframework.org/default.aspx/4@0/4@0/DEVDIV_TFS/Dev10/Releases/RTMRel/ndp/clr/src/BCL/System/Collections/Generic/Dictionary@cs/1305376/Dictionary@cs>

<https://www.codeproject.com/Tips/126403/CustomDictionary-Class>

<http://stackoverflow.com/questions/6264794/create-dictionary-item-on-the-fly-with-operator>

<http://programmingwithmosh.com/csharp/csharp-collections/>