FAKES FRAMEWORK

* Fakes framework helps us to test our unit in isolation.

It generally uses two concepts:

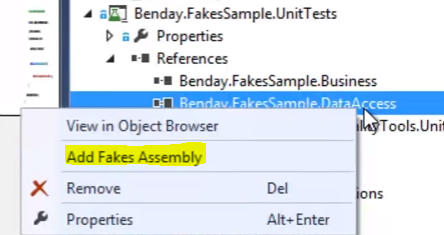
**SHIMS**: A shim modifies the compiled code of your application at run time so that instead of making a specified method call, it runs the shim code that your test provides. Shims can be used to replace calls to assemblies that you cannot modify, such .NET assemblies.

**STUBS**: A stub replaces a class with a small substitute that implements the same interface. To use stubs, you have to design your application so that each component depends only on interfaces, and not on other components. (By "component" we mean a class or group of classes that are designed and updated together and typically contained in an assembly.)

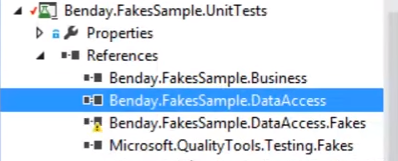
**Steps to write Fakes Unit test**:

* Identify the dll we want to fake. This is the layer next to our testing unit and since we don’t want to test any dependency(Unit test) so we fake that layers dll.

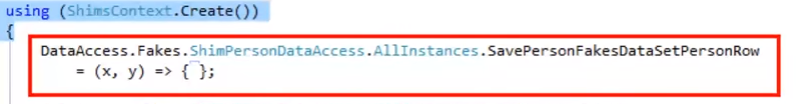
Right click the dll and click Add Fake Assembly.



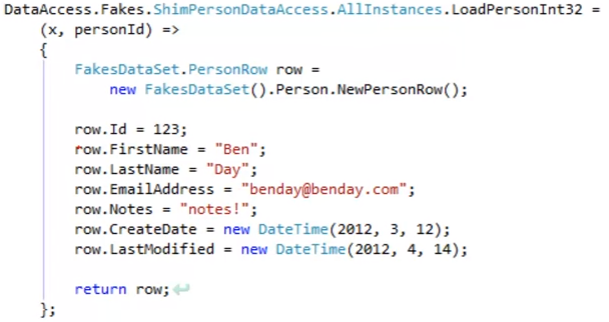
* Doing this will add some fakes assemblies to the unit testing project.



* Now lets say we have a method SavePersonFakesDataSetPersonRow(it just saves data doesn’t return anything) method in the assembly we don’t wanna depend and we want to fake it.
* By fake syntax we do like this:



* This class **ShimPersonDataAccess** is a class automatically generated by shim framework.
* Now lets say we have a method LoadPersonInt32(), this method takes int as input argument and will return a person object, then inside ShimsContext we fake it like this :



* By doing this we now don’t call the actual implementation, we call these implementations which we have provided In fakes.

For Stub Implimentation, we use to stub the interfase.

If we have a IPerson interface then we need to do like this:

