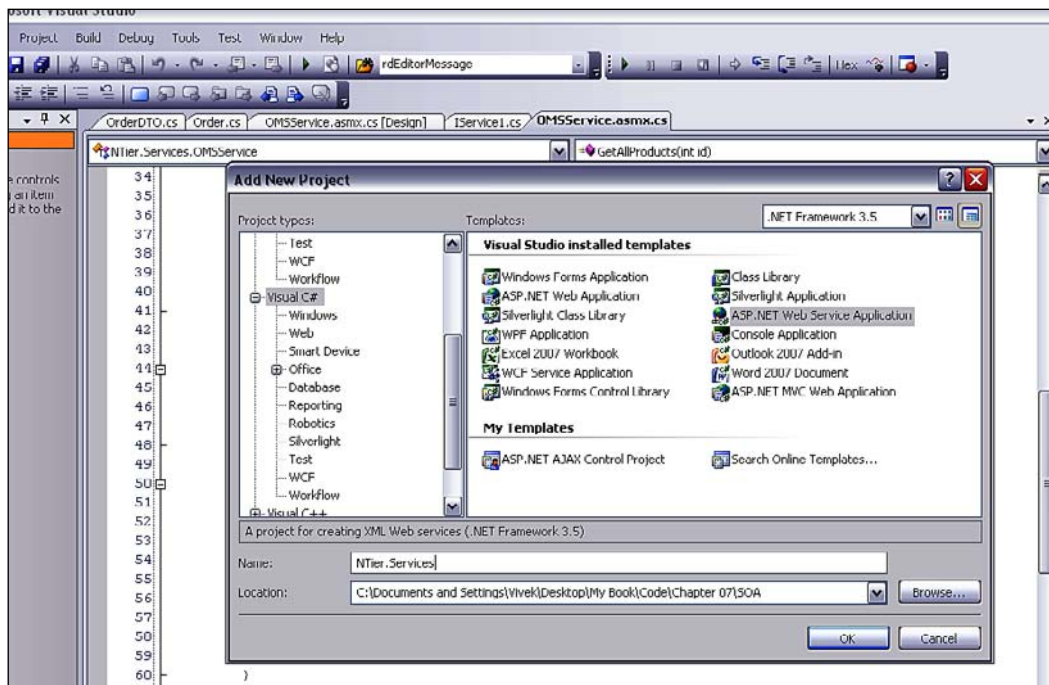


In this example, our main aim is to see how a service interface can provide another level of abstraction between the GUI and BL/DAL tiers, and enable our project to talk to multiple clients (or GUIs) using XML web services. Let's look at a simple example that illustrates how loose coupling can be implemented.

Building a Service Interface

Let us open the same 5-tier project that we created in Chapter 4. Add a new ASP.NET web services project to the solution, and name it `NTier.Services` :



Now right-click on the new `NTier.Services` project, select **Add New Item**, and from that dialog box select the **Web Service** option. Name the file `OMSService.asmx`. The ASMX extension is used to denote web service files in the Microsoft .NET framework, and they are basically placeholders from where our main web service code runs. In an `asmx.cs` file, we can have multiple web methods (denoted by the `[WebMethod]` attribute before the method definition).

We will create a simple web method that returns a list of products in an XML SOAP format, in our newly-created `OMSService.asmx.cs` code file. Note that the VS and the .NET framework make it very easy for us to create and consume web services, as most of the SOAP formatting and other related work is handled by VS, letting us focus on the web methods. Here is how our `FindAllCustomers` method in the Business Layer looks (as seen in Chapter 4):

```
public Collection<Customer> FindAll(LoadStatus loadStatus,
    int start,int pageLength, out int noOfPages)
{
    try
    {
        Collection<CustomerDTO> dtoList =
        CustomerDAL.GetAllCustomers(
        loadStatus,start, out noOfPages);
        foreach(CustomerDTO dto in dtoList)
            Customer customer = new Customer(dto);
            this.Add(customer);
        }
    return this;
}
catch(Exception ex)
{
    //handle exception
    throw;
}
```

This returns a strongly-typed collection of `Customer` objects by calling the DAL method from `CustomerDAL` (for details of these methods, please refer to Chapter 4).

Now, we define a new web method in our `OMSService.asmx.cs` file called `GetAllCustomers()`, and call the BL method as follows:

```
/// <summary>
/// Get all customers
/// </summary>
/// <returns>An array of customer DTOs</returns>
[WebMethod]
public CustomerDTO[] GetAllCustomers()
{
    CustomerCollection customers = new CustomerCollection();
    CustomerDTO[] customerArray;
    //get all customers so set page length =-1 indicating
    full load
```