

Yerra Solutions Ltd

Yerra Clearly Web Application

Code Review Document



YERRA CLEARLY
LEGAL SPEND MANAGEMENT

Stephen Pino
4-6-2016

Table of Contents

1.0 Overview & Scope	3
1.1 Breakpointing.....	4
1.2 General flow of an MVC application	5
1.3 ASP.net MVC introduction	6
1.4 High level overview of a generic Solution Explorer	7
Section 1: Yerra_Web	8
1.5 ASP.net Generic Application	8
1.6 Yerra_Web – Project Property files.....	10
1.7 Properties - PublishProfiles_Inspinia.pubxml.....	12
1.8 Properties – AssemblyInfo.cs	14
1.9 References	16
2.0 Visual studio Object Browser	19
2.1 App_Data folder	20
2.2 App_Start folder.....	21
2.2.1 BundleConfig.cs	22
2.2.2 FilterConfig.cs	23
2.2.3 IdentifyConfig.cs	23
2.2.4 RouteConfig.cs.....	24
2.2.5 Regex Expressions	29
2.2.6 Startup.Auth.cs	33
2.3 Code Templates Folder – cs.t4 Class files	35
2.4 Content folder / CSS files & Images	37
2.5 Controllers Folder	41
2.5.1 Archive folder	42
2.5.2 AdminManageUploads.cs	42
2.5.3 ReportsController.cs.....	44
2.5.4 UploadController.cs	45
2.6 Fonts folder	47
2.6.1 Fonts	47
2..6.2 CSS	47
2.6.3 Sass	48
2..6.4 Less	48
2.6.5 Glyphicons	48
2.7 Helpers folder	48
2.7.1 HTMLHelperExtensions.cs	48
2.8 Images folder	49
2.9 Infrastructure folder	50

3.0 Models folder	50
3.1 Yerra_Reporting_BL.cs	51
3.2 Scripts folder	57
3.3 Views folder	58
3.3.1 _ViewStart.cshtml & _Layout_2.cshtml	58
3.4 Global.asax	60
3.5 Packages.config	61
3.6 Web.config	62
3.6.1 What are the Web.Debug.config and Web.release.Config files for?	62
3.7 DbContext class	63
3.8 Section 2 DAL (Data Access Layer)	64
3.9 DAL Properties – AssemblyInfo.cs	64
4.0 DAL References	65
4.1 DAL Scripts folder	65
4.2 DAL Packages.config	66
4.3 DAL PageAccess.cs	66
4.4 DAL Published_InvoicesPartial.cs	68
4.5 DAL RAWDATA_STAGING.cs	69
4.6 DAL ReceivedEmailAttachmentInfo.cs	69
4.7 DAL ROCHE_RAWDATA_STAGING.cs	69
4.8 eMailProcessor	70
4.9 Currency Convertor	70
5.0 Github	72
5.1 Data Quality [DQ] issues	74
5.2 Section 3 Database	77
5.2.1 Database Utilisation	77
5.2.2 Database design	77
5.3 Cross Charging Report	81
5.4 Report spend for each law firm	83
5.5 Report list of all law firms with their total spend	83
5.6 Site YTD spend for 2015	84
5.7 Yerra Clearly Editable report (GUI) – new development	85
5.8 Report title required?	86
5.9 Legal Spend Application – new development	87
6.0 False Flagging – Data does match IPM?	87
6.1 User Manager Application – new development	88
6.2 Truncating data into SQ Ltable	88
6.3 Appendix	90

Date	Version	Author	Comments
08.04.2016	1.0	Stephen Pino	Creation of Technical Document for Source Code review

1.0 Overview & Scope

The aim of this document is to provide a, low level, technical analysis of the Yerra Clearly Web Application looking at its source code flow structure and separation of concerns (SoC). It is aimed at all software developers, in particular junior level. Some knowledge of the .Net framework 4.5 & ASP.Net MVC 5 would be a prerequisite to fully understand this document.

To begin with the Visual Studio GUI and source code development suite provide a “Solution Explorer” a lot like Windows File Explorer. The Yerra Clearly version of the Solution Explorer will be used as the backbone of this code review document. We will analyse each file in the Yerra Clearly Application Solution Explorer (top to bottom) and explain briefly what classes, methods, objects, routes etc. it contains and their respective function, flow structure. We will pay close attention to Models, Views and Controller folders and their respective classes files; also in particular “routing”.
 *This report can be used to update comments in the source code of Yerra Clearly Web Application.
 *Please also refer to Yerra Reporting invoice tool by Mark Barnes (Creator/Lead developer) for a high level overview of the application.

- The report will look at the C# Classes (blueprints) and data models and how they flow and communicate with one another in the source code & database. I have produced a class diagram (see appendix) this is the high level blueprint for the application.
- The report will aim to break Yerra Clearly down to bitesize chunks as the old saying goes - How do you eat an elephant? One bite at a time.
- The Yerra Clearly application is split into 3 sections: **1. Yerra_Web, 2. DAL – meaning Data Access Layer (DAL), 3. SQL Server database.**
- For comparison purposes Yerra Clearly will initially be compared to a generic ASP.net template.
- The report will identify the Files/Classes (blueprints) and use techniques known as ‘debugging’ and ‘break-pointing’ to establish what each bit of the source code performs.
- The Developer feedback will be sort as support. <http://blog.codinghorror.com/learn-to-read-the-source-luke/>



Note: contributions to the document will be made from the Lead contract software developer Yerra Junior software developer, external developers and Yerra Senior analysts.

1.1 Breakpointing

Introduction:

"So What is breakpoint?

When you run the program in Visual Studio IDE, if you want to check the execution of a particular routine or function at that time you can set the breakpoint. Probably, it's set before to run the application. So when the program executes, the breakpoint hits and the program waits there. So that you can step into the program and take a deep look into the code and check the values there itself.

This is the main advantage of breakpoints."

Source: <http://www.c-sharpcorner.com/article/debugging-with-visual-studio/>

Tip! "As a software developer, you probably know the “don’t repeat yourself” (DRY) principle and apply it every day."

Developer comments:

Debug Tracing:

Initial Senior analyst question: "Adam is trying to upload the US data that we receive monthly, we usually upload this in batches of 500-1000 line items however the system is not accepting batches of 500 – the request to validate is timing out after about 20 seconds or so. Do you know why this suddenly can't be uploaded as we've always uploaded batches of this size?"

Thanks,"

Yerra Lead developer: "thanks for your message, this is an error I'm not expecting as nothing has been changed for a while in terms of that part of the module. Would you be able to send me a copy of the file please and I will run a debug trace in the development environment to see what could be causing the issue? Thanks."

Junior developer asks: "With regards to the debug trace (in yellow below) how do you perform this? Is this a console application, that you have wrote separately, that scans the Yerra application for debugging purposes then outputs specific details on the console output screen?"

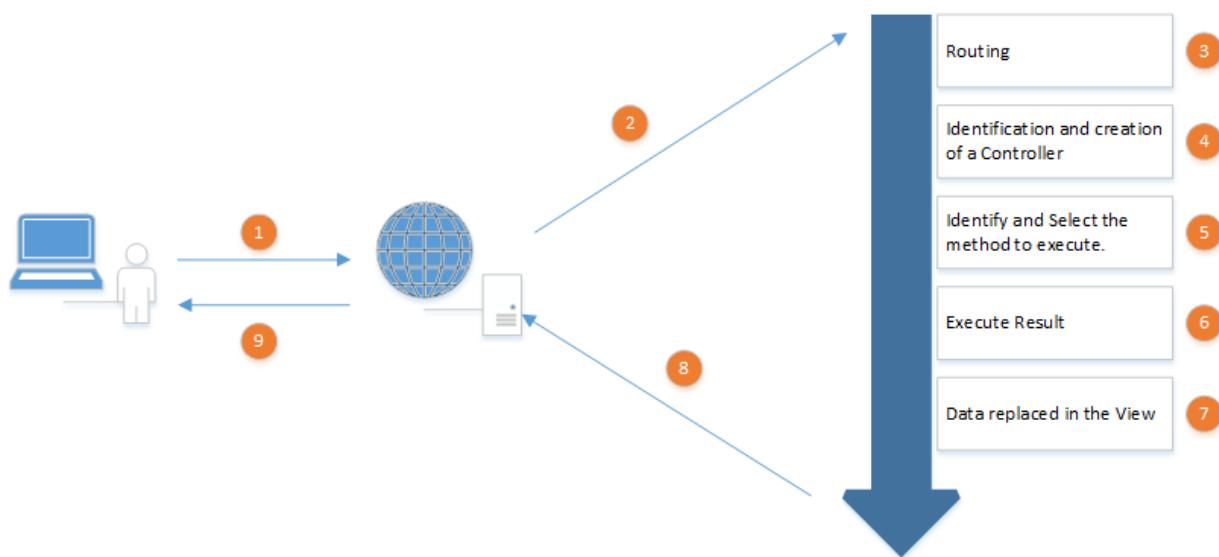
Can you educate me on this please?"

Lead developer: "No worries, you can attach to external processes using the VS Debugger but it is advised that is restricted to internal networks/active directory domains as there is a risk something malicious could also attach to it if exposed."

1.2 General flow of an MVC application

"The flow of an MVC application can get confusing, but I want to visually show you how a simple request moves through the ASP.NET pipeline.

1. A user requests a web page using a browser.
2. The server receives the request.
3. If this is the first time the web application is called, **Route objects** are added to the **RouteTable object**. Once the **route table** is created, we build additional routing objects and perform the routing using the **RouteTable**, which creates the **RouteData**, then the **RequestContext**.
4. Now that we have our Request Context, the **MvcRouteHandler** creates an **MvcHandler**. The **MvcHandler** is what kicks off the start of an MVC Application. It also identifies which controller to use based on the **HTTP request**. The **MvcHandler** creates the controller and calls the **execute method**.
5. Once we have our Controller identified and selected, we need to find the method to execute. This is done by using the **ControllerActionInvoker**. Once found, it executes that method.
6. The action method is executed while **passing in user parameters**, orchestrates response data, and returns a resulting **ActionResult**. Usually, a **ViewResult** is returned by MVC.
7. The data is passed through to either the default View or a specified different View through a **ViewResult**. The data is laid into the **View (HTML)** using **Razor placeholders**. Think of it as a template.
8. Once the data is in the View, it now processes the View, performing replacements by executing code **resulting in plain HTML**.
9. The final View is sent back to the user in their browser."



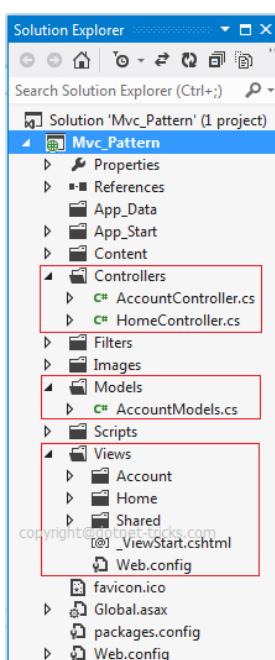
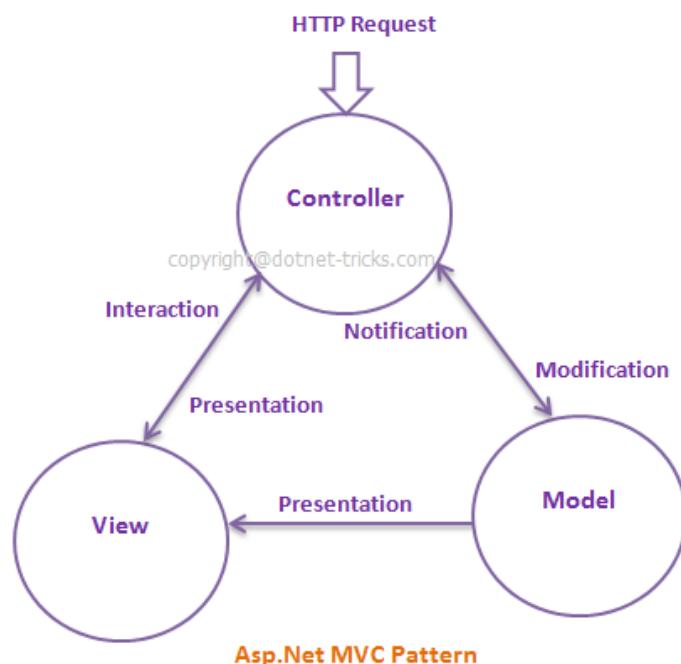
Source: https://blog.udemy.com/asp-net-mvc-tutorial/#1_1

1.3 ASP.net MVC introduction

*"The Model-View-Controller (MVC) pattern was introduced in 1970s. It is a software design pattern that splits an application into three main aspects: Model, View and Controller. Moreover, MVC pattern forces a separation of concerns within an application **for example**, separating data access logic and business logic from the UI."*

Source: <http://www.dotnet-tricks.com/Tutorial/mvc/DIYG060113-Understanding-Model-View-Controller-in-Asp.Net-MVC.html>

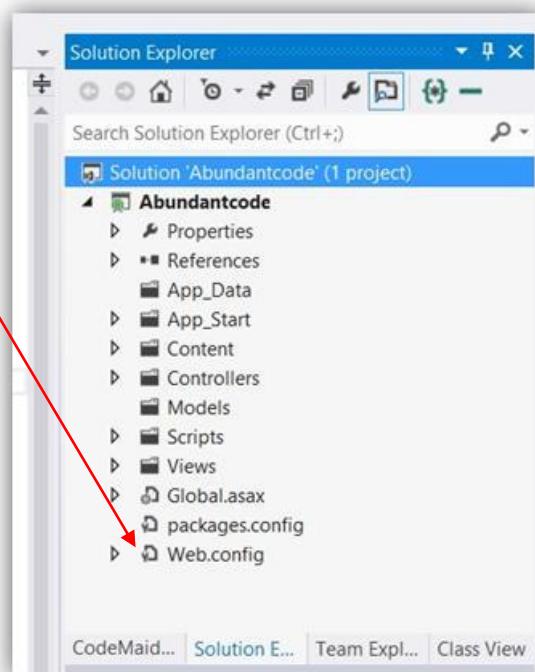
See diagram and image below:



1.4 High level overview of a generic Solution Explorer

- **App_Data** – The developers can add the databases like SQLite or SQL Server Express database file or XML files to this folder
- **App_Start** – This contains the core settings of the ASP.NET MVC Project.
- **Content** – Contains contents like CSS file and images.
- **Controllers** – contains the controller classes of the ASP.NET MVC Project
- **Models** – Contains the ViewModel or domain model classes
- **Views** – This folder can be used to add the views and partial views for the ASP.NET MVC Application. The shared folder contains the files related to the layouts and common views.
- **Scripts** – This folder holds the JavaScript files and libraries for the ASP.NET MVC Application.
- **Web.Config file in Views Folder** – This file contains the configuration that will make the views work with the ASP.NET MVC Project.
- **Global.asax** – This is one of the important file in the ASP.NET MVC Project and is the global application class.
- **Web.Config** – This is the configuration file for the application and is found in the root of the application.

*Overview Sourced from <http://abundantcode.com/files-and-folders-in-asp-net-mvc-4-project/>



Note: Yerra Clearly will be based on a slightly more complex Solution Explorer as seen in Section 1.1.

Source: <http://www.classonnet.in/Views/MVC/Getting%20Started%20with%20MVC.html#/Menu>

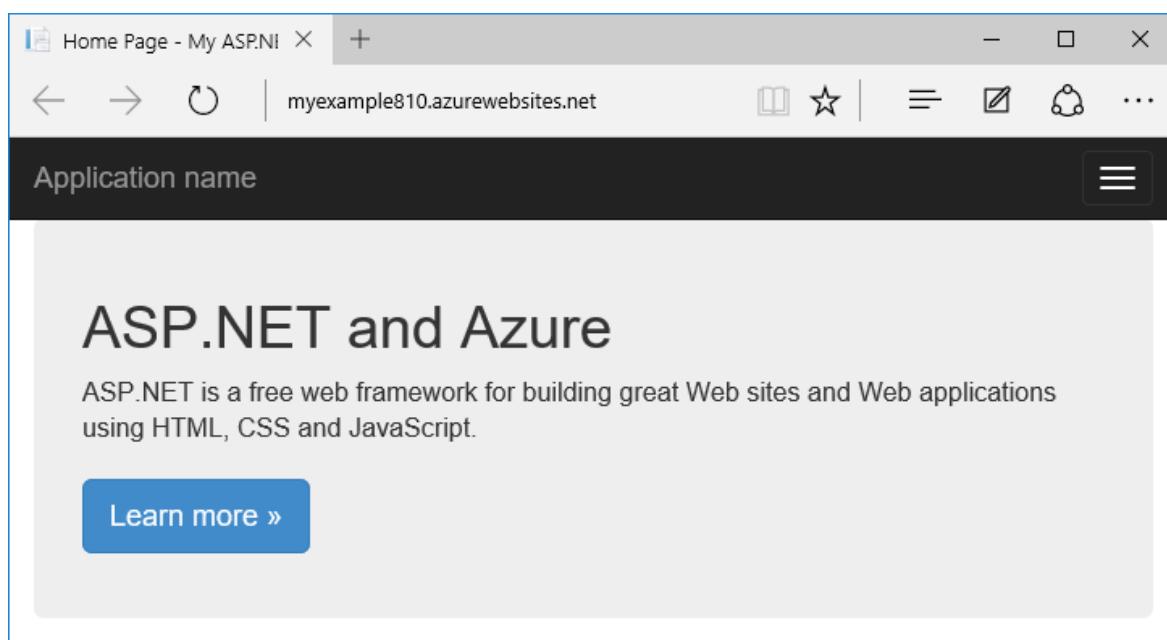
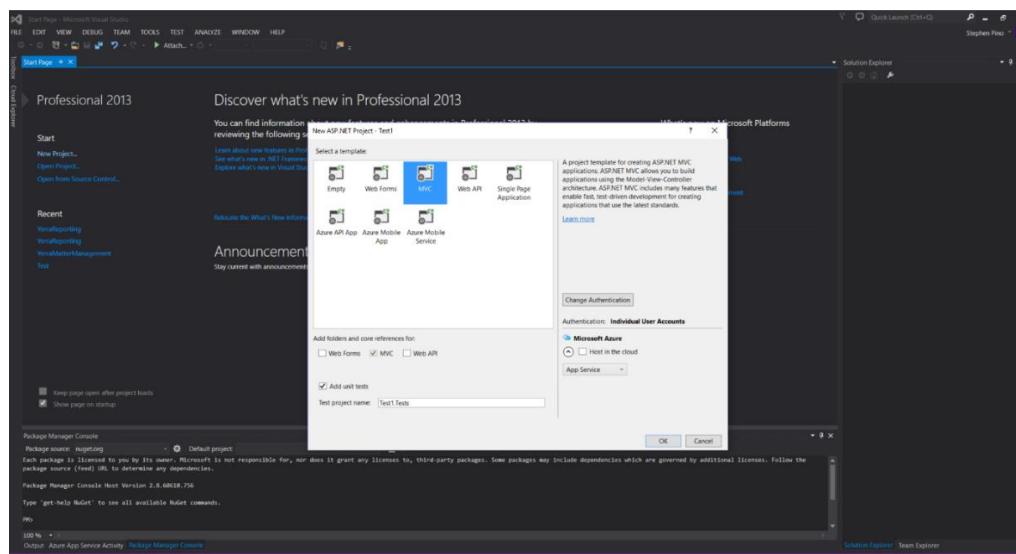
Section 1: Yerra_Web

1.5 ASP.net Generic Application

Introduction:

To begin with here is a picture of the Visual studio development suite and the file- new project solution screen allowing anyone to create a generic website template for a ASP.net MVC 5 application. All you would do at this stage is select MVC and give the project a name and click ok. A generic solution for a webpage is provided to the developer. You will notice the blank solution explorer to the right of the Visual Studio suite and an example of this as ‘active’ is on page 2 of this report on the left. Below also shows the generic webpage template when runningin Microsoft Edge web browser and in debug mode.

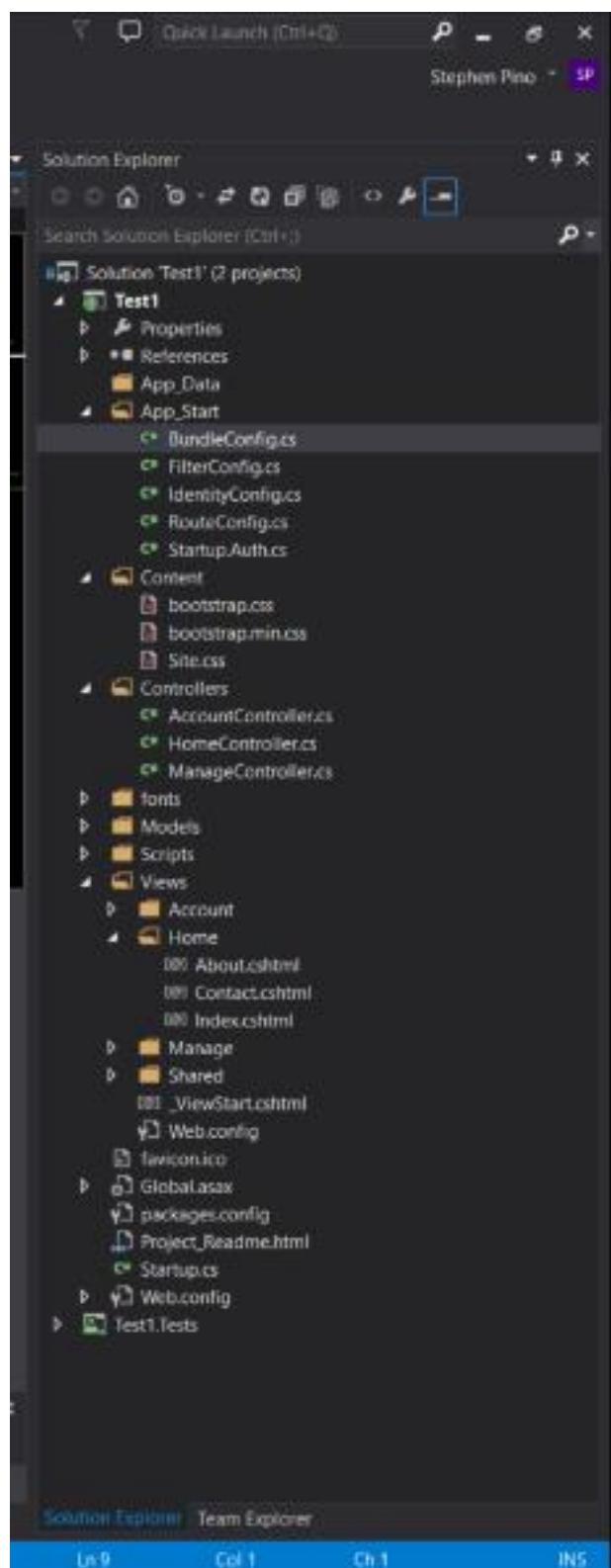
***Note:** This Microsoft website link is a good starting point for creating web applications in .NET
<https://azure.microsoft.com/en-gb/documentation/articles/web-sites-dotnet-get-started/>



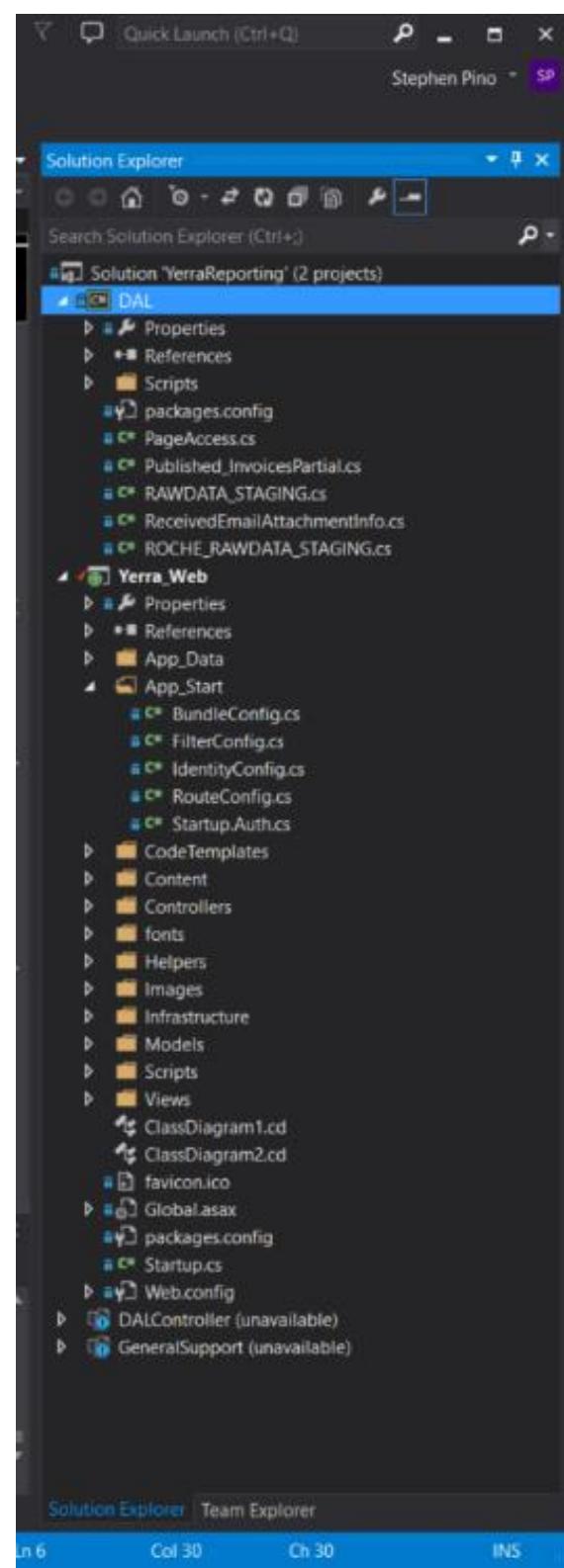
Solution Explorer comparison:

Below is a comparison between a generic Solution Explorer and Yerra Clearly Solution Explorer. This enables us to see how the application has been developed from a generic stand point & SoC.

Generic Solution Explorer:



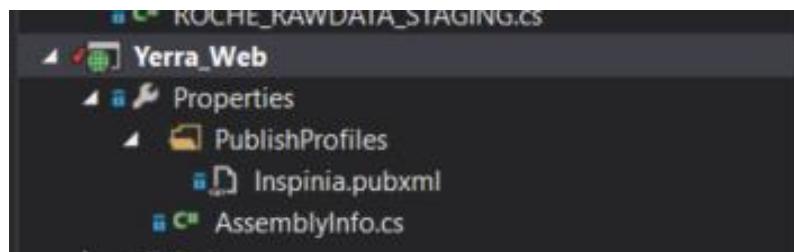
Yerra Clearly Solution Explorer:



1.6 Yerra_Web – Project Property files

Introduction:

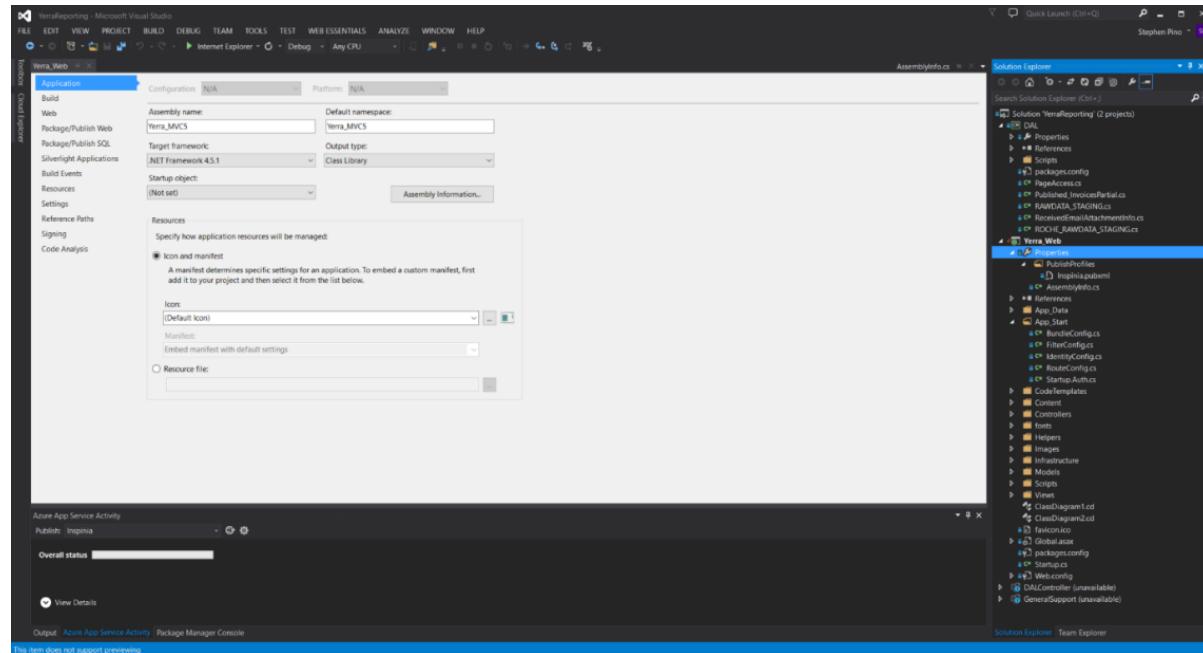
To begin with let us look at the properties folder & respective files:



You can access the project properties by either right clicking and selection properties from the menu list (at the bottom) or double clicking the properties file.

*Note: From below source: *"Some deployment options are configured in project properties that are stored in the project file (the .csproj or .vbproj file). In most cases, the default values of these settings are what you want, but you can use the **Project Properties** UI built into Visual Studio to work with these settings if you have to change them."*

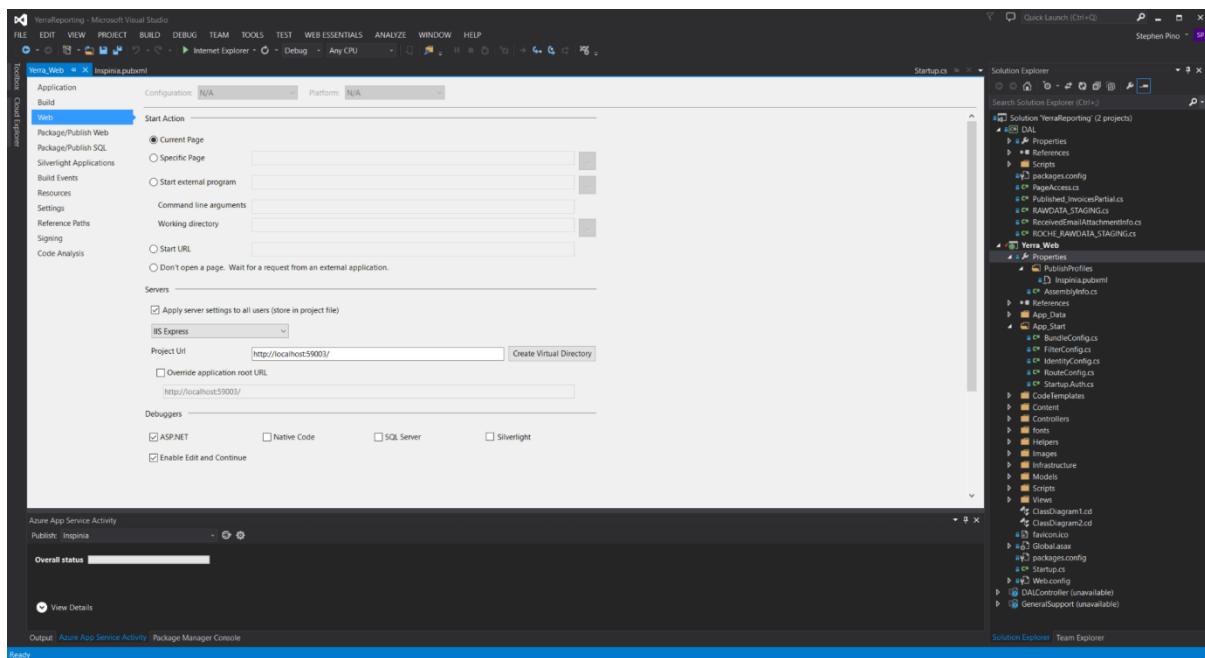
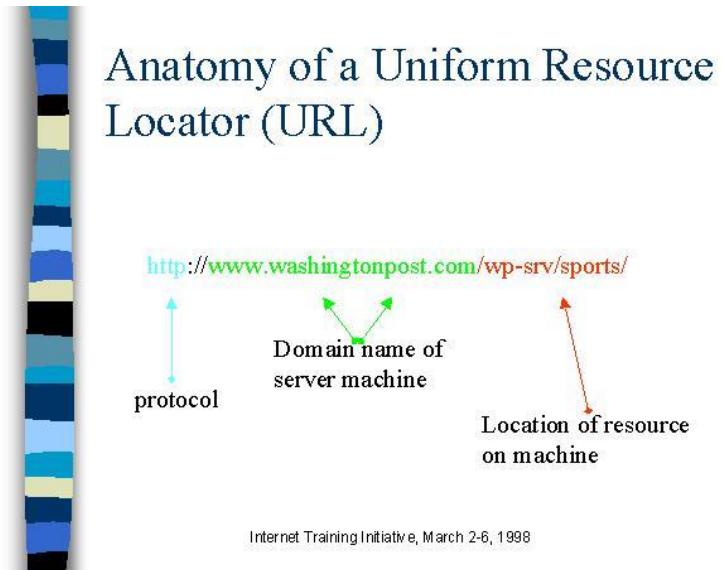
*Source: <http://www.asp.net/mvc/overview/deployment/visual-studio-web-deployment/project-properties>



Web properties tab, shown in below image, can be used to configure a Project Uniform Resource Locator (URL**) for example when creating a virtual directory, you might want to create a secure local host with a SSL certificate if deciding to developing in and publish to the cloud.**

"URL definition: URL is an acronym for Uniform Resource Locator and is a reference (an address) to a resource on the Internet. A URL has two main components: Protocol identifier: For the URL <http://example.com>, the protocol identifier is http . Resource name: For the URL <http://example.com>, the resource name is example.com ."

*Source: <https://docs.oracle.com/javase/tutorial/networking/urls/definition.html>

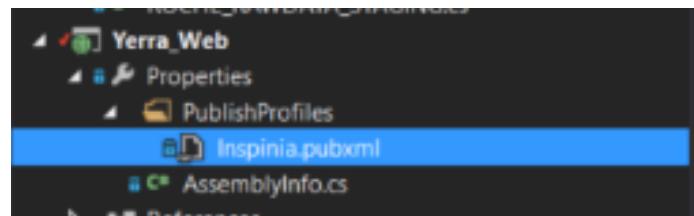


"URLs are instead mapped to specific methods which execute in response to the incoming request, generally returning either a View (MVC) or some sort of structured data (Web API) corresponding to the requested resource. In other words, instead of pointing to actual physical resources within a file system, MVC and Web API routes instead point to an abstraction which represents the resource requested, in both cases a method which will return the requested item."

Source: <http://johnatten.com/2013/07/21/routing-basics-in-asp-net-mvc/#What-makes-a-good-URL>

1.7 Properties - PublishProfiles_Inspinia.pubxml

Introduction:



First off please see the green comments in the file itself below. This is good developer/coding practice. This should be applied throughout any application built.

Enclosed source code - Inspinia.pubxml file content:

A screenshot of Microsoft Visual Studio showing the 'Inspinia.pubxml' file in the code editor. The window title is 'YerraReporting - Microsoft Visual Studio'. The menu bar includes FILE, EDIT, VIEW, PROJECT, BUILD, DEBUG, TEAM, XML, TOOLS, TEST, WEB ESSENTIALS, ANALYZE, WINDOW, and HELP. The toolbar has various icons for file operations. The code editor displays the XML configuration for the publish profile. It includes sections for project tools version, property groups, and item groups. A green comment block is present within the XML structure.

```
<?xml version="1.0" encoding="utf-8"?>
<!--
  This file is used by the publish/package process of your Web project. You can customize the behavior of this process
  by editing this MSBuild file. In order to learn more about this please visit http://go.microsoft.com/fwlink/?LinkId=208812.
-->
<Project ToolsVersion="4.0" xmlns="http://schemas.microsoft.com/developer/msbuild/2003">
  <PropertyGroup>
    <WebPublishMethod>FileSystem</WebPublishMethod>
    <LastUsedBuildConfiguration>Release</LastUsedBuildConfiguration>
    <LastUsedPlatform>Any CPU</LastUsedPlatform>
    <SiteUrlToLaunchAfterPublish></SiteUrlToLaunchAfterPublish>
    <LaunchSiteAfterPublish>True</LaunchSiteAfterPublish>
    <ExcludeApp_Data>False</ExcludeApp_Data>
    <publishUrl>C:\Users\mark\OneDrive\Code\Published\YerraDashboard\2016-01-21A</publishUrl>
    <DeleteExistingFiles>False</DeleteExistingFiles>
    <PrecompileBeforePublish>True</PrecompileBeforePublish>
    <EnableUpdateable>True</EnableUpdateable>
    <DebugSymbols>False</DebugSymbols>
    <WDPMergeOption>DoNotMerge</WDPMergeOption>
  </PropertyGroup>
  <ItemGroup>
    <DeployParameterValue Include="$(DeployParameterPrefix)ScaffoldingContext-Web.config Connection String" />
  </ItemGroup>
</Project>
```

Breakpointing:

In respect to the Yerra Clearly application version, breakpointing (F10) through this file takes you through the following route: **Bundle.Config.cs** then **Global.asax.cs** then **Startup.cs** then back to **Global.asax.cs** then back to **BundleConfig.cs**. accessing their respective methods along the way.

C# Methods:

In **Startup.cs** it is a generic method that remains unchanged:

```
[assembly: OwinStartupAttribute(typeof(Yerra_MVC5.Startup))]
namespace Yerra_MVC5
{
    public partial class Startup
    {
        public void Configuration(IAppBuilder app)
        {
            ConfigureAuth(app);
        }
    }
}
```

```

        }
    }
}
```

In **Global.asax.cs** file accessing the inherited **BundleConfig C# Class**. This is generic and remains unchanged:

```

public class MvcApplication : System.Web.HttpApplication
{
    protected void Application_Start()
    {

        AreaRegistration.RegisterAllAreas();
        FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);
        RouteConfig.RegisterRoutes(RouteTable.Routes);
        BundleConfig.RegisterBundles(BundleTable.Bundles);

    }
}
```

Developer comments:

“From the top of my head it will be the gui framework publishing file not necessarily used by the application for anything important.”

And

“VS 2013 is clever enough to work out multiple users sharing files so it will also create local copies (usually suffixed by “_user” or similar).

You would not necessarily change this file direct but rather through the project properties.”

Secondly for editing the file; here is an extract from the link provided in the **.pubxml** file itself and shown below:

[Editing Publish Profile \(.pubxml\) Files](#)

*When you want to configure settings that are specific to a particular publish profile, you edit the publish profile file. Publish profile files are named <filename>.pubxml and are located in the PublishProfiles folder. The PublishProfiles folder is under Properties in a C# web application project, under My Project in a VB web application project, or under App_Data in a web site project. Each .pubxml file contains settings that apply to one publish profile. The values you enter in the **Publish Web** wizard are stored in these files.”*

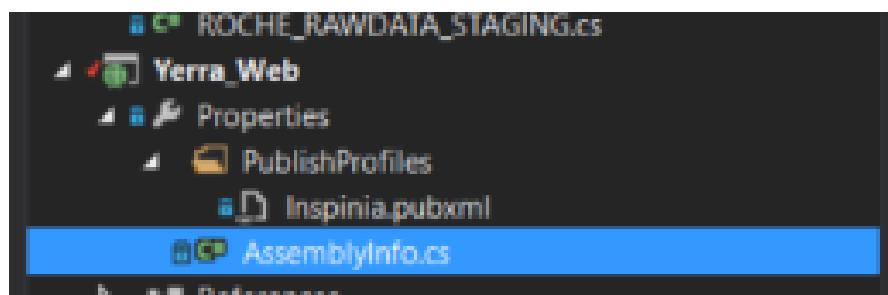
*Source: [https://msdn.microsoft.com/en-us/library/ff398069\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/ff398069(v=vs.110).aspx)

*Source: <http://stackoverflow.com/questions/23095446/what-is-a-pubxml-user-file>

Note: Some deployment options can only be specified by editing one of the XML files that control the Web Publishing Pipeline (WPP).

1.8 Properties – AssemblyInfo.cs

Introduction:



Enclosed file source code:

Firstly, here is the explanation for this C# Class file:

"AssemblyInfo.cs contains information about your assembly, like name, description, version, etc. You can find more details about its content reading the (green) comments that are included in it (see below source code).

If you delete it, your assembly will be compiled with no information, i.e., in the Details tab of the file properties you will see no name, no description, version 0.0.0.0, etc.

The value associated with assembly: Guid is the ID that will identify the assembly if it will be exposed as a COM object. So, if your assembly isn't COM-exposed, you don't need this. It is randomly generated. In any case, normally, you don't need to modify it."

Source: <http://stackoverflow.com/questions/13180543/what-is-assemblyinfo-cs-used-for>

```
using System.Reflection;
using System.Runtime.CompilerServices;
using System.Runtime.InteropServices;

// General Information about an assembly is controlled through the following
// set of attributes. Change these attribute values to modify the information
// associated with an assembly.
[assembly: AssemblyTitle("Yerra_MVC5")]
[assembly: AssemblyDescription("")]
[assembly: AssemblyConfiguration("")]
[assembly: AssemblyCompany("")]
[assembly: AssemblyProduct("Yerra_MVC5")]
[assembly: AssemblyCopyright("Copyright © 2014")]
[assembly: AssemblyTrademark("")]
[assembly: AssemblyCulture("")]

// Setting ComVisible to false makes the types in this assembly not visible
// to COM components. If you need to access a type in this assembly from
// COM, set the ComVisible attribute to true on that type.
[assembly: ComVisible(false)]

// The following GUID is for the ID of the typelib if this project is exposed to COM
[assembly: Guid("0869213c-15bd-4ee9-820e-b8ca69fdf067")]

// Version information for an assembly consists of the following four values:
//
```

```
// Major Version
// Minor Version
// Build Number
// Revision
//
// You can specify all the values or you can default the Revision and Build Numbers
// by using the '*' as shown below:
[assembly: AssemblyVersion("1.0.0.0")]
[assembly: AssemblyFileVersion("1.0.0.0")]
```

AssemblyInfo.cs GUI:

You can also access the information in this Class file through VS GUI as follows:

"Go to your Project Properties, the Application tab, and click the Assembly Information button.

That's what is stored in AssemblyInfo.cs.

In Windows Explorer, right click your project's .exe output, select Properties, and go to the Details tab. That is the information generated by AssemblyInfo.cs."

And

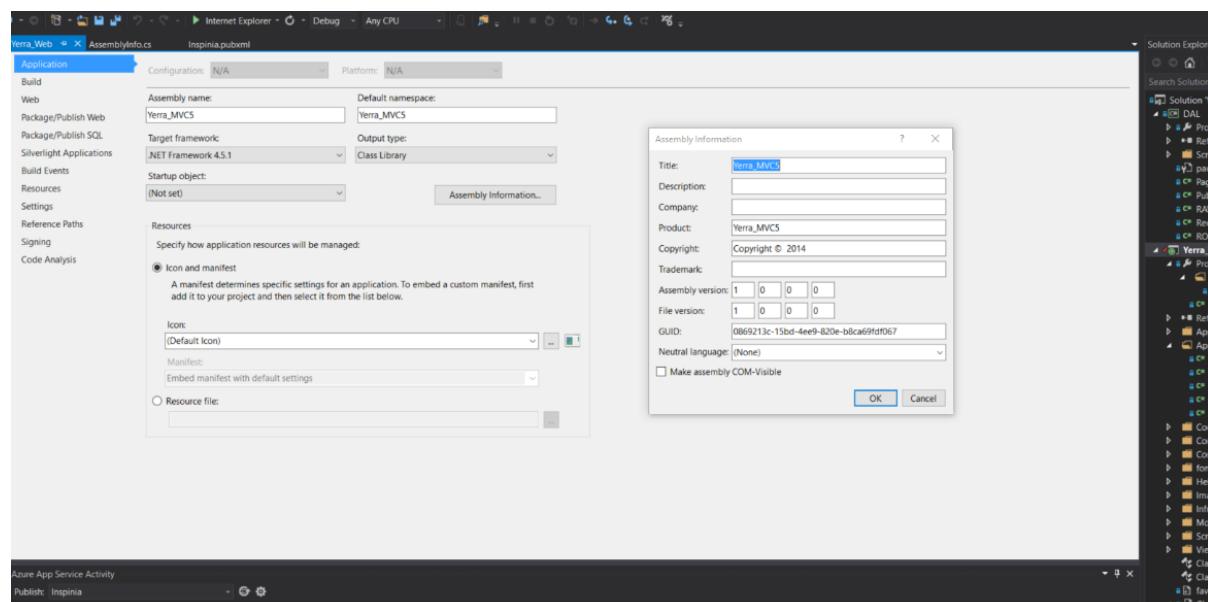
"In AssemblyInfo file you can store information which you can get from every place in the project, so you don't have to update all the places but just the assemblyInfo.

For example - in this file you update the version number, and it is updated automatically in your site. In the html page, to get the version number, write:

@Assembly.GetAssembly(typeof(ProjectName.WebSite.Controllers.MyController)).GetName().Version.ToString() *and it will be updated each time you upload a new version."*

Source: <http://stackoverflow.com/questions/13180543/what-is-assemblyinfo-cs-used-for>

See below image of VS GUI example:



Paying attention to this section of the code – the GUID “Represents a globally unique identifier (GUID)”.

Source: [https://msdn.microsoft.com/en-us/library/system.guid\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.guid(v=vs.110).aspx)

```
// The following GUID is for the ID of the typelib if this project is exposed to COM
[assembly: Guid("0869213c-15bd-4ee9-820e-b8ca69fdf067")]
```

Developer comments:

Lead developer: “*the AssemblyInfo file is a header that is placed into the program executable and can be controlled via the properties option for each project (not solution) basically the output from the project properties is rendered in this particular file.*”

Breakpointing:

“*The resulting GUID is the program's unique identifier which you would use to reference in WEB.CONFIG or APP.CONFIG if it were setup as a library. If you check application / library references within most WEB.CONFIGs you would see **public token keys** - this is the same Guid reference. Using a 32-bit GUID isn't 100% guaranteed to be unique of course across the world but the chances of two libraries being used in the same solution are extremely remote and not worth worrying about.*”

1.9 References

Introduction:

“*Before you write code against an external component or connected service, your project must first contain a reference to it. A reference is essentially an entry in a project file that contains the information that Visual Studio needs to locate the component or the service.*

*To add a reference, right click on the References node in Solution Explorer and choose **Add Reference.**”*

Also

“*All projects contain an implied reference to mscorelib. Visual Basic projects contain an implied reference to **Microsoft.VisualBasic**.*

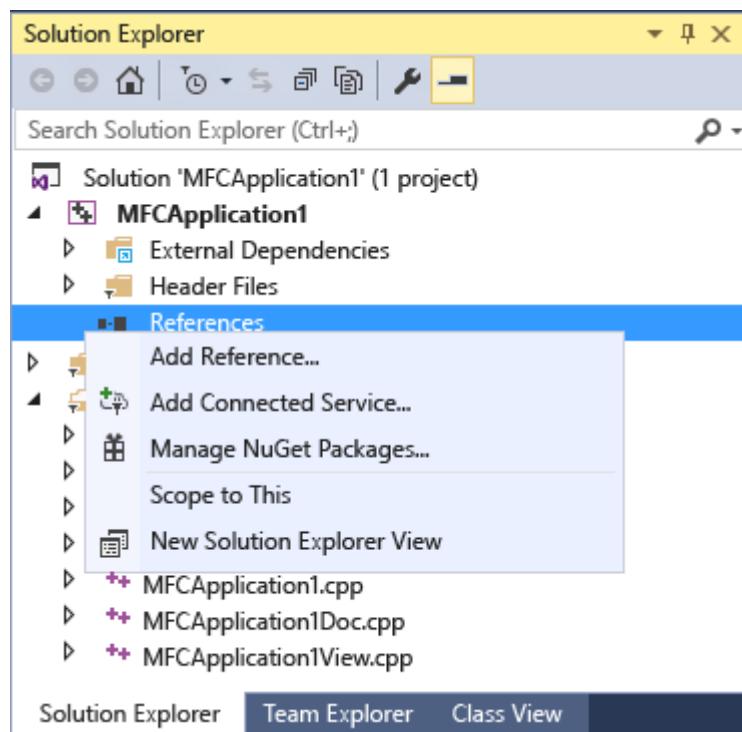
*All projects in Visual Studio contain an implied reference to **System.Core**, even if **System.Core** is removed from the list of references.”*

Source: <https://msdn.microsoft.com/en-us/library/ez524kew.aspx>

Here is an example of Yerra Clearly references:



Here is a generic Reference list and how to add a Reference – Right click – Add reference:



You can make a reference to the following types of components/services:

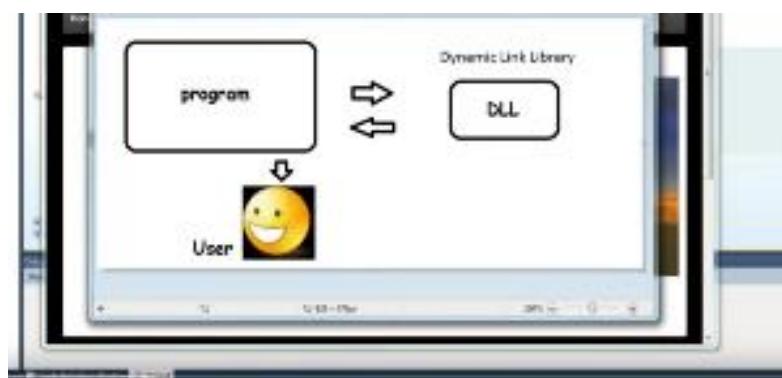
- Windows Store app references
- .NET Framework class libraries or assemblies
- COM components
- Other assemblies or class libraries of projects in the same solution
- XML Web services

Source: <https://msdn.microsoft.com/en-us/library/ez524kew.aspx>

.DLL Files as References

*Note: Only used for Windows applications.

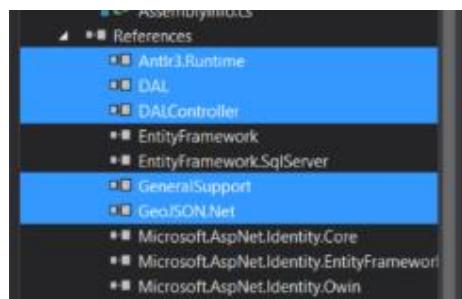
References file types are as follows; “*Dynamic Link Libraries (**DLL**)s are like EXEs but they are not directly executable. They are similar to .so files in Linux/Unix. That is to say, **DLLs** are MS’s implementation of shared libraries. **DLLs** are so much like an EXE that the **file** format itself is the same.”*



Source: <https://www.google.co.uk/webhp?sourceid=chrome-instant&ion=1&espv=2&ie=UTF-8#q=what+is+a+dll+file> 23 Sep 2008.

*Note: **Script files** maybe better to use, in the creation of games for example, as they do not need to be compiled like a .dll file, which does need compiling. Script files speed up the idea of the running without too many problems. For web applications .dll files are fine though.

The Yerra Clearly application has all the standard references (.dll files) that come with a generic ASP.net website template but also include the following highlighted references for additional services etc. See image below.



System.Core is missing from Yerra Clearly reference list, however, this is actually still being used by the app. Summary of the System can be seen in the image below.

<http://stackoverflow.com/questions/458098/business-logic-layer-and-data-access-layer-circular-dependency>

Developer comments:

Lead developer: "Antlr3 (added by Visual Studio for scripted builds), GeoJSON.NET (used for the mapping). You quite often get these additional references when you add packages via NuGet?"

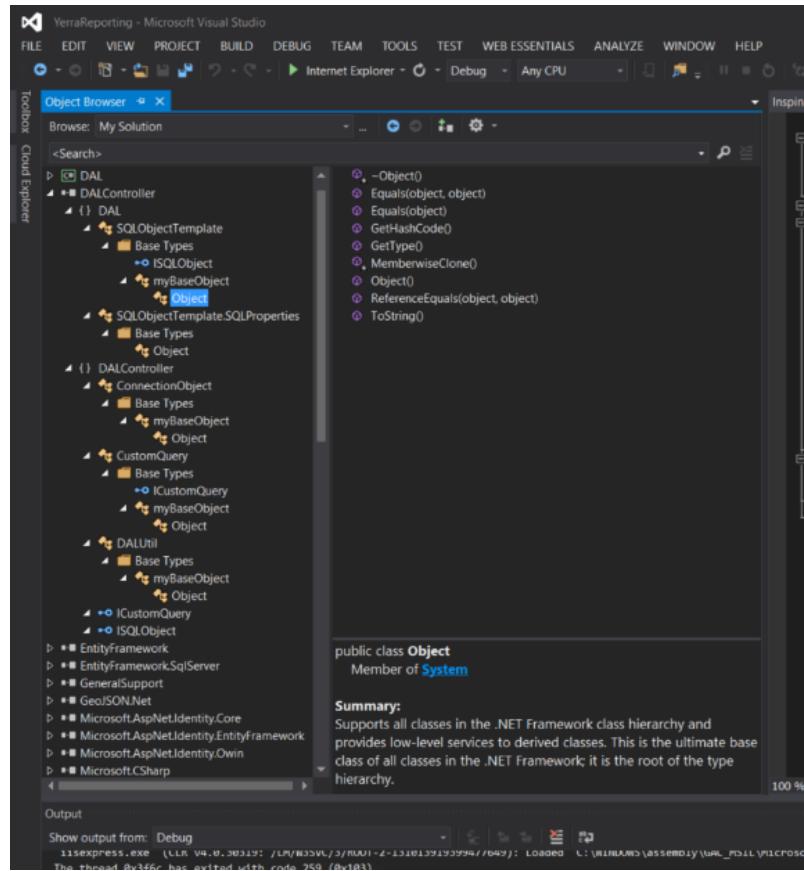
2.0 Visual Studio Object Browser

Introduction:

"The **Object Browser** lets you select and examine the namespaces, classes, methods, and other programming symbols available for use in your project. You can open the **Object Browser** from the **View** menu, by clicking the **Object Browser** button on the main toolbar, or by typing **CTRL+ALT+J**."

Source: [https://msdn.microsoft.com/en-us/library/exy1facf\(v=vs.100\).aspx](https://msdn.microsoft.com/en-us/library/exy1facf(v=vs.100).aspx)

Please see the summary list in below image showing browsing references:



Developer comments:

DAL Controller reference see above image

Lead developer: "the DAL is the Data Access Layer for the database and GeneralSupport contains a bunch of generic methods specifically created to act as helpers for any application that I create."

*DAL is actually a subset of DALController which is the true library - both DALController and GeneralSupport are a set of libraries I created around 4-5 years ago designed to assist any developer within my department to quickly (and to enforce standardisation) create applications without having to re-create commonly required routines. General Support also contains extensive error handling and enumeration code. General Support also underpins *all* other libraries created in this application (DALController also depends upon it).*

What is the most important one...(reference) well they all are! :)

2.1 App_Data folder

Introduction:

The Yerra Clearly applications uses this folder with no content inside as per a generic template.

Microsoft explains the purpose of this folder as:

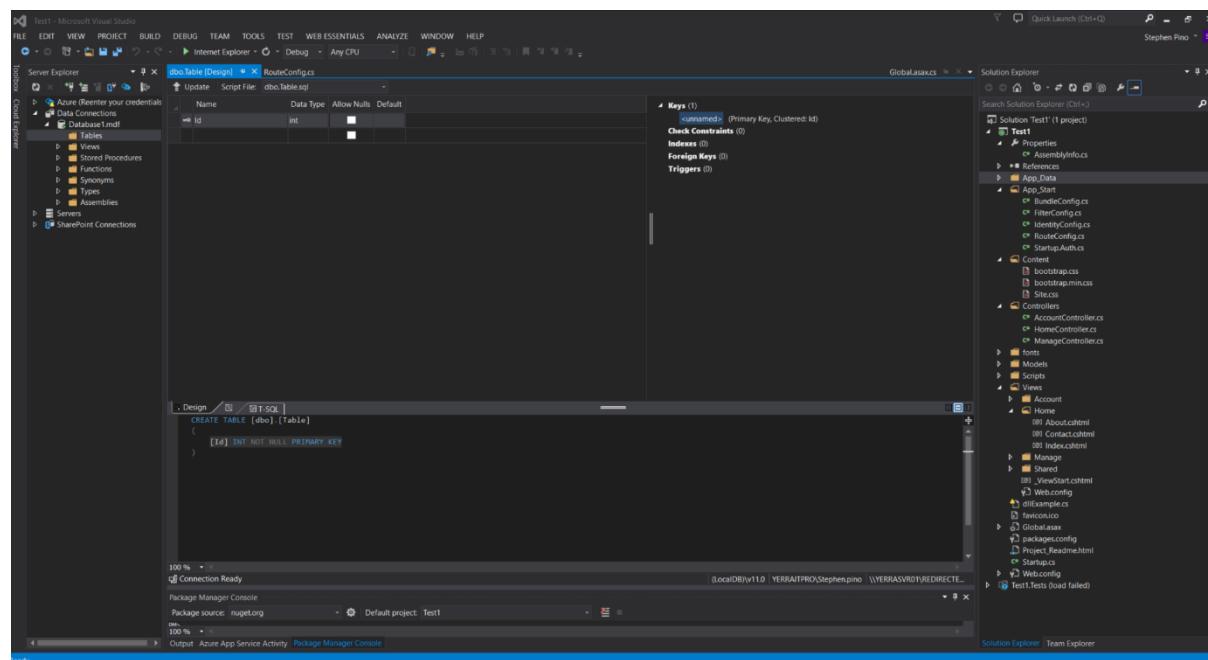
"Contains application data files including .mdf database files, XML files, and other data store files.

The App_Data folder is used by ASP.NET to store an application's local database, such as the

database for maintaining membership and role information. For more information, see [Introduction to Membership](#) and [Understanding Role Management](#).

Source: <https://msdn.microsoft.com/en-us/library/ex526337.aspx>

You can add things like your database in here see below example:



2.2 App_Start folder

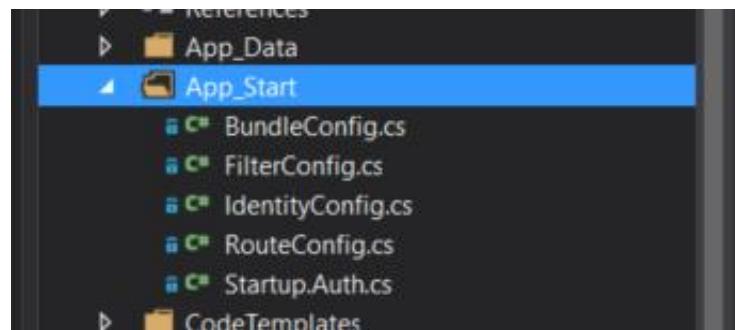
Introduction:

"App_Start folder has been introduced in MVC4. It contains various configurations files like as BundleConfig.cs, FilterConfig.cs, RouteConfig.cs, WebApiConfig.cs for your application. All these settings are registered within Application_Start method of Global.asax.cs file.

- **BundleConfig.cs** – This is used to create and register bundles for CSS and JS files. By default, various bundles are added in this files including jQuery, jQueryUI, jQuery validation, Modernizr, and Site CSS.
- **FilterConfig.cs** – This is used to register global MVC filters like error filters, actions filters etc. By default it contains HandleErrorAttribute filter.
- **RouteConfig.cs** – This is used to register various route patterns for your ASP.NET MVC application. By default, one route is registered here named as Default Route.
- **WebApiConfig.cs** – This is used to register various WEB API routes like as ASP.NET MVC, as well as set any additional WEB API configuration settings."

Source: http://www.planetofcoders.com/what-is-app_start-folder-in-asp-net-mvc/

See image of this folder below:



The Yerra Clearly application **App_Start** folder has the same identical folder as a generic App_Start folder.

2.2.1 BundleConfig.cs

The Yerra Clearly Class file differs from the generic class file as it contains much more bundle methods.

Developer comments:

Lead developer: *"BundleConfig is a recently new addition and is specific to MVC - you can basically tie a bunch of JS files (or CSS styling files) together under one bundle and then when you reference it using the "bundle path" that you configures it will load *all* of the files tied in with that bundle. This is really good as it means when you get updated files with different file names (e.g. jquery_<version number>.js) you update the bundleconfig only without having to update all of the references within the pages."*

Breakpointing:

Bundle.Config.cs then **Global.asax.cs** then **Startup.cs** then back to **Global.asax.cs** then back to **BundleConfig.cs**. accessing their respective methods along the way.

This images actually shows a generic Bundle.Config.cs file:

```

1  using System.Web;
2  using System.Web.Optimization;
3
4  namespace Test1
5  {
6      public class BundleConfig
7      {
8          // For more information on bundling, visit http://go.microsoft.com/fwlink/?LinkId=301862
9          public static void RegisterBundles(BundleCollection bundles)
10         {
11             bundles.Add(new ScriptBundle("~/bundles/jquery").Include(
12                         "~/Scripts/jquery-{version}.js");
13
14             bundles.Add(new ScriptBundle("~/bundles/jqueryval").Include(
15                         "~/Scripts/jquery.validate*"));
16
17             // Use the development version of Modernizr to develop with and learn from. Then, when you're
18             // ready for production, use the build tool at http://modernizr.com to pick only the tests you need.
19             bundles.Add(new ScriptBundle("~/bundles/modernizr").Include(
20                         "~/Scripts/modernizr-*");
21
22
23             bundles.Add(new ScriptBundle("~/bundles/bootstrap").Include(
24                         "~/Scripts/bootstrap.js",
25                         "~/Scripts/respond.js");
26
27             bundles.Add(new StyleBundle("~/Content/css").Include(
28                         "~/Content/bootstrap.css",
29                         "~/Content/site.css");
29         }
30     }
31 }

```

2.2.2 FilterConfig.cs

Introduction:

The Yerra Clearly application source code remains generic, see below:

```
using System.Web;
using System.Web.Mvc;

namespace Yerra_MVC5
{
    public class FilterConfig
    {
        public static void RegisterGlobalFilters(GlobalFilterCollection filters)
        {
            filters.Add(new HandleErrorAttribute());
        }
    }
}
```

These are linked or registered in the **Global.asax.cs** file – see following code.

Source: <http://dotnet-concept.com/Tutorials/2015/1/5798809/BundleConfig-FilterConfig-and-RouteConfig-in-MVC>

```
namespace Yerra_MVC5
{
    public class MvcApplication : System.Web.HttpApplication
    {
        protected void Application_Start()
        {
            AreaRegistration.RegisterAllAreas();
            FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);
            RouteConfig.RegisterRoutes(RouteTable.Routes);
            BundleConfig.RegisterBundles(BundleTable.Bundles);
        }
    }
}
```

Developer comments:

None available at this time.

Breakpointing - Source code flow:

FilterConfig.cs then **Global.asax.cs** then **Startup.cs** then back to **Global.asax.cs** accessing their respective generic methods along the way.

2.2.3 IdentityConfig.cs

Introduction:

This remains as a **generic** file in the Yerra Clearly Application.

Background:

"The ASP.NET Identity framework was originally introduced in 2013 as the follow-on to the ASP.NET Membership system, a staple of MVC applications for several years, but which was beginning to show its age. Originally, ASP.NET Identity presented a useful, if somewhat minimal API for managing"

security and authorization in the context of a public-facing web application built using ASP.NET. The Identity framework introduced modern features such as social network log-in integration, and easily extensible user model definitions.

The new RTM release introduces the following features, among others:

- *Extended User Account Definition, including Email and contact information*
- *Two-Factor Authentication via email or SMS messaging, functionally similar to that used by Google, Microsoft, and others*
- *Account Confirmation via email*
- *Administrative management of Users and Roles*
- *Account Lock-Out in response to invalid log-in attempts*
- *Security Token Provider to regenerate a user's security token in response to changes in security settings.*
- *Improved support for Social log-ins*
- *Easy Integration of Claims-Based Authorization”*

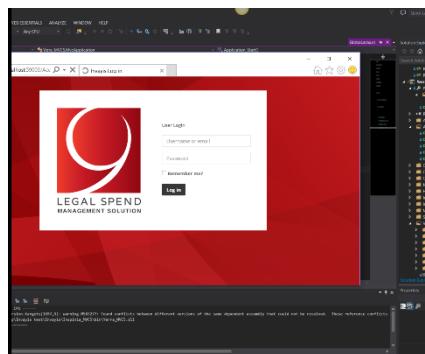
Source: <http://www.codeproject.com/Articles/762428/ASP-NET-MVC-and-Identity-Understanding-the-Basics>

Developer comments:

None provided at this time.

Breakpointing:

IdentityConfig.cs then **Global.asax.cs** then **Startup.cs** then **filteringConfig.cs** then **Global.asax.cs** back to **Startup.cs** then finally the Yerra Clearly log in screen (see image below) accessing their respective generic methods along the way.



This **IdentityConfig.cs Class** file contains methods that follow a similar path to the above passing through these files utilising the methods. As our lead developer describes these are “*just C# routines connecting to different parts of the environment.*”

2.2.4 RouteConfig.cs

Introduction:

This is a fundamental Class file for this or any web application. It has been enhanced and modified greatly from a generic file of this kind. We will breakpoint through an example of a none generic method in this file and find its function – this will be used as an example for this document.

"A route is a URL pattern that is mapped to a handler. The handler can be a physical file, such as an .aspx file in a Web Forms application. A handler can also be a class that processes the request, such as a controller in an MVC application. To define a route, you create an instance of the [Route](#) class by specifying the URL pattern, the handler, and optionally a name for the route.

You add the route to the application by adding the [Route](#) object to the static [Routes](#) property of the [RouteTable](#) class. The [Routes](#) property is a [RouteCollection](#) object that stores all the routes for the application.

You typically do not have to write code to add routes in an MVC application. Visual Studio project templates for MVC include preconfigured URL routes. These are defined in the [MvcApplication](#) class, which is defined in the Global.asax file." See screenshot below.

Source: <https://msdn.microsoft.com/en-us/library/cc668201.aspx#routes>

```

1  using System;
2  using System.Collections.Generic;
3  using System.Data.Entity;
4  using System.Linq;
5  using System.Web;
6  using System.Web.Mvc;
7  using System.Web.Optimization;
8  using System.Web.Routing;
9
10 using Yerra_MVC5;
11
12 public class MvcApplication : System.Web.HttpApplication
13 {
14     protected void Application_Start()
15     {
16
17         AreaRegistration.RegisterAllAreas();
18         FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);
19         RouteConfig.RegisterRoutes(RouteTable.Routes);
20         BundleConfig.RegisterBundles(BundleTable.Bundles);
21     }
22 }
23
24
25

```

"Introduction

ASP.NET MVC uses ASP.NET routing to map incoming browser requests to controller action methods. It is a pattern-matching system for mapping incoming requests to specified MVC Controllers and Actions. If it fails to map the route for an incoming request then MVC will show a 404 error. ASP.NET Routing uses a route table. A route table is created when your web application first starts. When an MVC application first starts, the Application_Start() method is called. This method, in turn, calls the RegisterRoutes() method. The RegisterRoutes() method creates the route table and registers one or more route patterns to the route table. If we create an application using MVC 4 or MVC 5, it has a default route register in Route.config as in the following."

Source: <http://www.c-sharpcorner.com/UploadFile/mscratnesh/routing-in-Asp-Net-mvc571/>

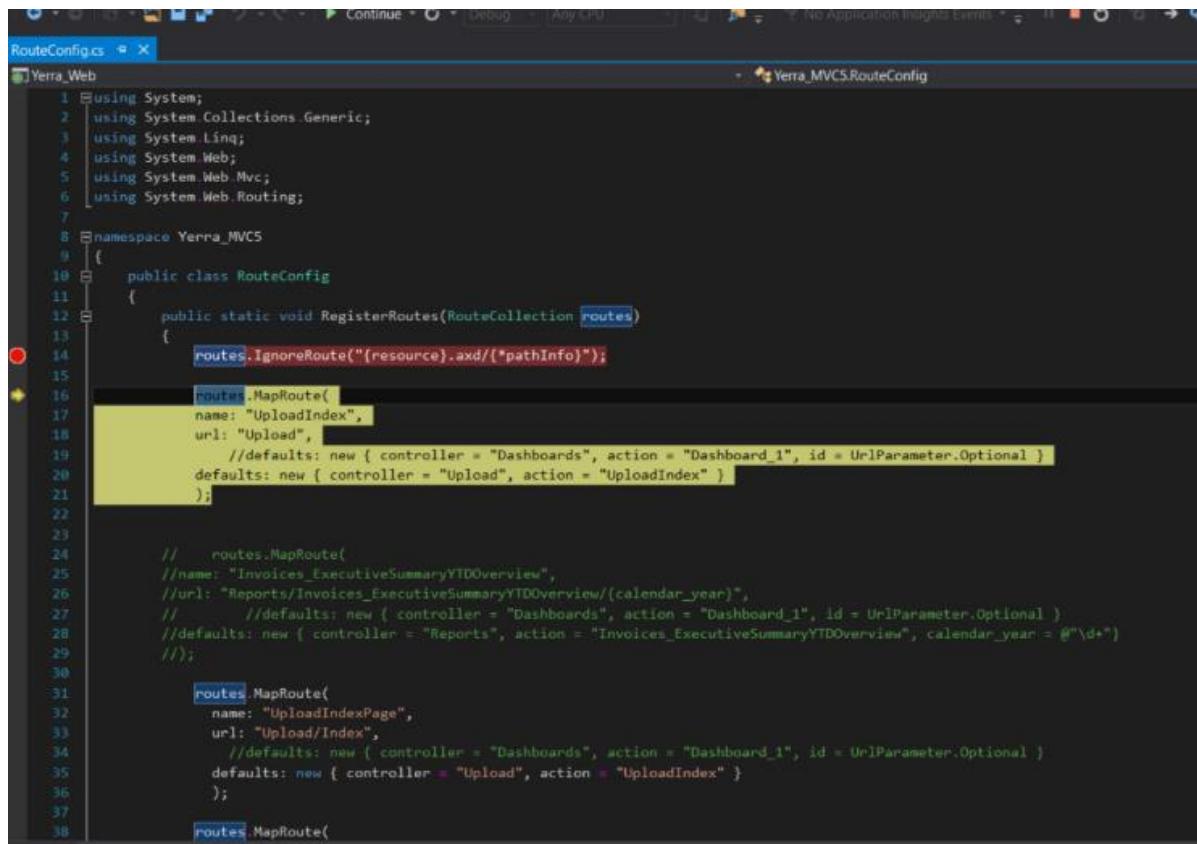
Developer comments:

Lead developer: *"One of the most important files I would advise you get your head around is Route.Config - this controls how an HTTP request is sent to a controller and then subsequently to the "view" file."*

Breakpointing:

Breakpointing screenshot examples below. The Red spot indicates the breakpoint and allows you to iterate through the code step by step.

1. RouteConfig.cs (runs through all methods in this class file)

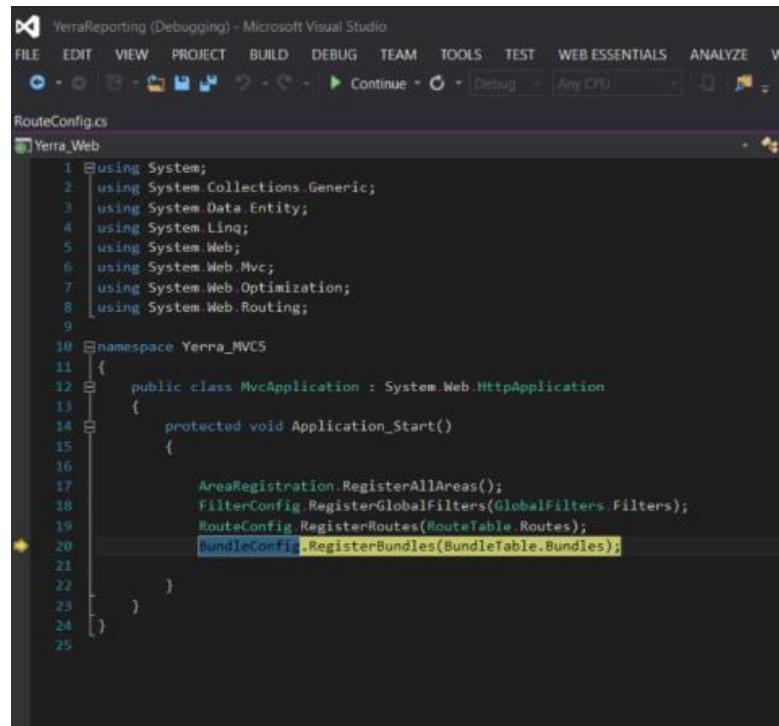


```

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Web;
5  using System.Web.Mvc;
6  using System.Web.Routing;
7
8  namespace Yerra_MVC5
9  {
10     public class RouteConfig
11     {
12         public static void RegisterRoutes(RouteCollection routes)
13         {
14             routes.IgnoreRoute("{resource}.axd/{*pathInfo}");
15
16             routes.MapRoute(
17                 name: "UploadIndex",
18                 url: "Upload",
19                 //defaults: new { controller = "Dashboards", action = "Dashboard_1", id = UrlParameter.Optional },
20                 defaults: new { controller = "Upload", action = "UploadIndex" }
21             );
22
23
24             // routes.MapRoute(
25             // name: "Invoices_ExecutiveSummaryYTDOverview",
26             // url: "Reports/Invoices_ExecutiveSummaryYTDOverview/{calendar_year}",
27             // defaults: new { controller = "Dashboards", action = "Dashboard_1", id = UrlParameter.Optional },
28             // defaults: new { controller = "Reports", action = "Invoices_ExecutiveSummaryYTDOverview", calendar_year = @"\d+" }
29             //);
30
31             routes.MapRoute(
32                 name: "UploadIndexPage",
33                 url: "Upload/Index",
34                 //defaults: new { controller = "Dashboards", action = "Dashboard_1", id = UrlParameter.Optional },
35                 defaults: new { controller = "Upload", action = "UploadIndex" }
36             );
37
38             routes.MapRoute(

```

2. Global.asax.cs with RouteConfig and BundleConfig accessed in the below method.



```

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS TEST WEB ESSENTIALS ANALYZE WI
Continue ▶ Debug Any CPU
RouteConfig.cs
Yerra.Web
1  using System;
2  using System.Collections.Generic;
3  using System.Data.Entity;
4  using System.Linq;
5  using System.Web;
6  using System.Web.Mvc;
7  using System.Web.Optimization;
8  using System.Web.Routing;
9
10 namespace Yerra_MVC5
11 {
12     public class MvcApplication : System.Web.HttpApplication
13     {
14         protected void Application_Start()
15         {
16
17             AreaRegistration.RegisterAllAreas();
18             FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);
19             RouteConfig.RegisterRoutes(RouteTable.Routes);
20             BundleConfig.RegisterBundles(BundleTable.Bundles);
21
22         }
23     }
24 }
25

```

3. Startup.cs (accessing Configuration method)

```

1  public static void Main(string[] args)
2  {
3      CreateWebHostBuilder(args).Build().Run();
4  }
5
6  public static IWebHostBuilder CreateWebHostBuilder(string[] args) =>
7      WebHost.CreateDefaultBuilder(args)
8          .ConfigureKestrel((context, options) =>
9              options.Limits.KeepAliveTimeout = TimeSpan.FromMinutes(1));
10
11         .Configure(app =>
12             app.UseStaticFiles();
13             app.UseRouting();
14             app.UseEndpoints(endpoints =>
15                 endpoints.MapControllerRoute(
16                     name: "default",
17                     pattern: "{controller=Home}/{action=Index}/{id?}"));
18         );
19     }
20
21     public class Program
22     {
23         public static void Main(string[] args)
24         {
25             CreateWebHostBuilder(args).Build().Run();
26         }
27     }
28
29     public class Startup
30     {
31         public void ConfigureServices(IServiceCollection services)
32         {
33             services.AddControllersWithViews();
34         }
35
36         public void Configure(IApplicationBuilder app)
37         {
38             app.UseStaticFiles();
39             app.UseRouting();
40             app.UseEndpoints(endpoints =>
41                 endpoints.MapControllerRoute(
42                     name: "default",
43                     pattern: "{controller=Home}/{action=Index}/{id?}"));
44         }
45     }
46 }

```

4. Clearly Log in screen appears at this point.

5. RouteConfig.cs

Source: <http://www.c-sharpcorner.com/UploadFile/mscratnesh/routing-in-Asp-Net-mvc571/>

Source: <http://stackoverflow.com/questions/16051061/mvc-route-config-order-routes>

Source: <http://stackoverflow.com/questions/10723628/custom-maproute>

Example of RouteConfig.cs and MVC:

As we are concentrating on the RouteConfig.cs Class file we already realise its significance in the application.

Seeing & understanding the link between the class/ controller / RouteConfig / View files – see below code snippets and how each file links with the one another:

ReportsController.cs method:

```

    public ActionResult IPM_ViewDetailsByResponsibleAttorneyAndCaseRecordNumber(string Responsible_Attorney,
string Case_Record_Number, int? calendar_year, int? calendar_month)
{
    Yerra_MVC5.Models.YerraReporting_BL y_bl = null;
    if (Responsible_Attorney == null || Case_Record_Number == null ||
calendar_month == null || calendar_year == null)
    {
        return View("Index", new Yerra_MVC5.Models.YerraReporting_BL());
    }
    y_bl = new Models.YerraReporting_BL(iCalendar_Year: (int)calendar_year,
iCalendar_Month: (int)calendar_month);
    y_bl.ViewDetailsInformation.IPM_RESPONSIBLE_ATTORNEY =
Responsible_Attorney;
}

```

```

        y_b1.ViewDetailsInformation.IPM_CASE_RECORD_NUMBER = Case_Record_Number;
    return View(y_b1);
}

```

RouteConfig.cs route map:

```

routes.MapRoute(
    name: "IPM_ViewDetailsByResponsibleAttorneyAndCaseRecordNumber",
    url:[REDACTED]
"Reports/IPM_ViewDetailsByResponsibleAttorneyAndCaseRecordNumber/{Responsible_Attorney
}/{Case_Record_Number}/{calendar_year}/{calendar_month}",
    //defaults: new { controller = "Dashboards", action = "Dashboard_1",
id = UrlParameter.Optional }
    defaults: new { controller = "Reports", action =
"IPM_ViewDetailsByResponsibleAttorneyAndCaseRecordNumber", Responsible_Attorney = "",
Case_Record_Number = "", calendar_year = @"\d+", calendar_month = @"\d+" }
);

```

IPM_ResponsibleAttorneyBySpend.cshtml:

```

@model Yerra_MVC5.Models.YerraReporting_BL
{
    ViewBag.Title = "Responsible Attorney and Client Matter by Spend";
}



<div class="col-lg-10">
        <h2>Responsible Attorney Spend - Quarter
@Model.ViewDetailsInformation.dtFirst_Calendar_Month.ToShortDateString()</h2>
        </div>
</div>



<div class="ibox float-e-margins">
        <div class="ibox-title ">
            <h5>Top 25 Responsible Attorneys</h5>
        </div>


```

Developer notes referring to above source code examples:

Lead developer: “One thing to remember is that both the case and spelling must match the respective method in the (respective) controller class exactly otherwise the desired method will never be hit. The case is the most common one that is often ignored.

Also if you are determining something as being an integer value you use @"\d+" (format code for numbers) and for string it is simply "". You can nominate other formats by looking at **regex expressions** though you only need to look at the basics. This is very important as it will dictate what it expects on the desired method side. **So I guess this predefines the property for when you pass the property into a controller method for example the reports controller?**

You can also make more “generic” routes (that can be found in the route.config) to cover a number of methods and situations though you need to be careful on ordering otherwise it may block a specific route you want to utilise.

A good article on route config ordering can be found here
<http://stackoverflow.com/questions/16051061/mvc-route-config-order-routes> "

Also,

In answer to the question in Red above - yes - this absolutely determines the expected variables that the method must support in the targeted controller class. However, there is nothing stopping you from having methods with the same names carrying different variable types as long as you have them all defined in the route.config."

2.2.5 Regex Expressions

Developer comments:

Junior developer: "I had just noticed this and could not remember what it meant - for integer value you use @"\\d+" - I do now, thanks! So these capture number sets as opposed to characters/strings.

Found this lovely example on Stackoverflow: <http://stackoverflow.com/questions/14017134/what-is-d-d-in-regex>

- (start a capture group
- \d a [shorthand character class](#), which matches all numbers; it is the same as [0-9]
- + one or more of the expression
-) end a capture group
- / a literal forward slash
- (start a capture group
- \d a [shorthand character class](#), which matches all numbers; it is the same as [0-9]
- + one or more of the expression
-) end a capture group

Here is an example:

```
>>> import re
>>> exp = re.compile('(\d+)/(\d+)')
>>> foo = re.match(exp, '1234/5678')
>>> foo.groups()
('1234', '5678')
```

Lead developer comment in reply to above:

"To be fair all you need to know about is "" and @\"\\d+" - if trying to pass entire classes over you can ignore the route.config and ensure that you embed the class details within the MVC view file to ensure it is picked up in the controller (usually embed by using a HIDDENFOR command and the respective ID).

MVC is a world way away from ASP.NET whereas it was a lot simpler in ASP.NET but you are "closer to the metal" with MVC."

Generic routing showing standard MapRoute() method;

```
routes.MapRoute(
    name: "ViewMatterNameByCountry",
    url: "Reports/ViewMatterNameByCountry/{country_name}/{calendar_year}/{calendar_month}",
```

```
//defaults: new { controller = "Dashboards", action = "Dashboard_1",
id = UrlParameter.Optional }
    defaults: new { controller = "Reports", action =
"ViewMatterNameByCountry", country_name = "", calendar_year =
 @"\d+", calendar_month =
 @"\d+" }
);
```

“One of the fundamental concepts to understand when using ASP.NET MVC and/or the ASP.NET Web API is routing, which essentially defines how your application will process and respond to incoming HTTP requests.”

Source: <http://johnatten.com/2013/07/21/routing-basics-in-asp-net-mvc/>

Another example:

“Example Route Mapping for Blog Application with Added Constraint:

```
routes.MapRoute(
    name: "BlogPost",
    url: "blog/posts/{postId}",
    defaults: new
    {
        controller = "Posts",
        action = "GetPost",
    },
    new {postId = @"\d+" }
);
```

The tiny Regular Expression `@\d+` in the code above basically limits the matches for this route to URLs in which the **postId** parameter contains one or more digits. In other words, nothing but integers, please.”

Source: <http://johnatten.com/2013/08/21/customizing-routes-in-asp-net-mvc/#Route-Matching>

Creating a new route example:

```

using System.Web.Mvc;
using System.Web.Routing;
namespace MvcApplication1
{
    public class MvcApplication : System.Web.HttpApplication
    {
        public static void RegisterRoutes(RouteCollection routes)
        {
            routes.IgnoreRoute("{resource}.axd/{*pathInfo}");
            routes.MapRoute(
                "Blog",
                "Archive/{entryDate}",
                new { controller = "Archive", action = "Entry" } // Parameter defaults
            );
            routes.MapRoute(
                "Default",
                "{controller}/{action}/{id}",
                new { controller = "Home", action = "Index", id = "" } // Parameter defaults
            );
        }
        protected void Application_Start()
        {
            RegisterRoutes(RouteTable.Routes);
        }
    }
}

```

The order of the routes that you add to the route table is important. Our new custom Blog route is added before the existing Default route. If you reversed the order, then the Default route always will get called instead of the custom route.

The custom Blog route matches any request that starts with /Archive/. So, it matches all of the following URLs:

```

/Archive/12-25-2009
/Archive/10-6-2004
/Archive/apple

```

The custom route maps to an incoming request to a controller named Archive and invokes the Entry() action. When the Entry() method is called, the entry date is passed as a parameter named entryDate.

You can use the Blog custom route with the controller in Listing 2.

Listing 2 - ArchiveController.cs

```

using System;
using System.Web.Mvc;
namespace MvcApplication1.Controllers
{
    public class ArchiveController : Controller
    {
        public string Entry(DateTime entryDate)
        {
            return "You requested the entry from " + entryDate.ToString();
        }
    }
}

```

"Routing is one of the primary aspects of the MVC framework which makes MVC what it is. While that is possibly over-simplifying, the routing framework is where the MVC philosophy of "convention over configuration" is readily apparent."

Source: <http://www.codeproject.com/Articles/641783/Customizing-Routes-in-ASP-NET-MVC>

Also,

"In the authoritative book [Professional ASP.NET MVC 4](#) , the authors refer to some common guidelines for high-quality URLs:

Usability expert Jacob Nielsen (www.useit.com) urges developers to pay attention to URLs and provides the following guidelines for high-quality URLs. You should provide:

- A domain name that is easy to remember and easy to spell
- Short URLs
- Easy-to-type URLs
- URLs that reflect the site structure
- URLs that are **hackable** to allow users to move to higher levels of the information architecture by hacking off the end of the URL
- Persistent URLs which don't change"

Source: <http://johnatten.com/2013/07/21/routing-basics-in-asp-net-mvc/#What-makes-a-good-URL>

John Atten also explains; "take note that we have added the new, more specific route **before the default in the file** (which determines the order it is added to the route dictionary). This is because **routes are evaluated for a match to an incoming URL in order**. The default MVC route a pretty general, and if route order is not carefully considered when adding additional routes to your project, the default route may inadvertently match a URL intended for a different handler."

Source: <http://johnatten.com/2013/08/21/customizing-routes-in-asp-net-mvc/#Route-Matching>

Source: <http://stackoverflow.com/questions/16051061/mvc-route-config-order-routes>

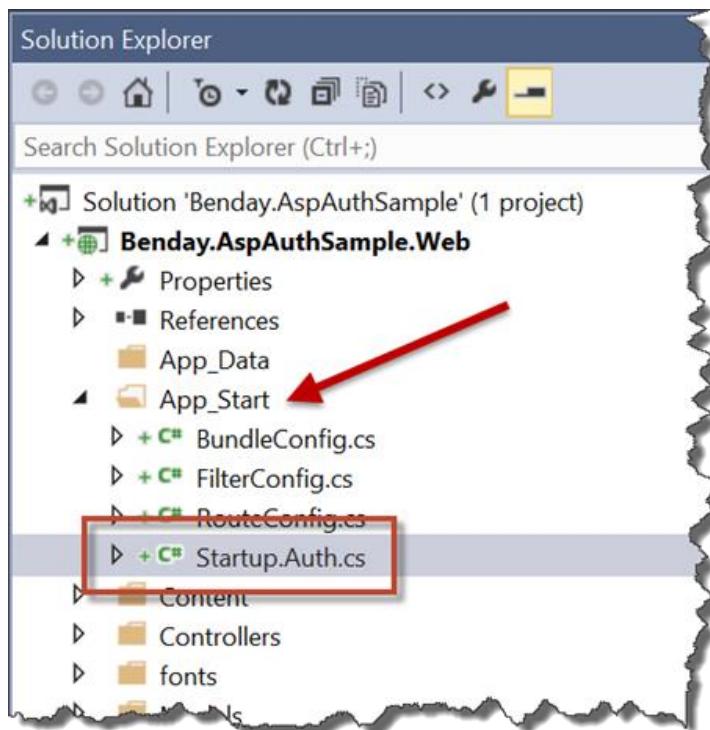
Also, John Atten asks us to also bear in mind that "*It is a sad fact that despite all the wonderfully simple examples we can find while learning, the real world often fails to [data] model in the way we want.*"

Source: <http://johnatten.com/2013/08/21/customizing-routes-in-asp-net-mvc/#Route-Matching>

2.2.6 Startup.Auth.cs

Introduction:

MSDN explains the reason for this folder and files as follows: "Open Startup.Auth.cs. The settings for the user authentication providers for your ASP.NET MVC project are stored and specified in the Startup.Auth.cs file. In Solution Explorer, go to the ASP MVC project and expand the App_Start folder. Double-click the Startup.Auth.cs file."



Source: <https://code.msdn.microsoft.com/ASPNET-MVC-5-Entity-12399352/sourcecode?fileId=144479&pathId=1036809692>

Developer comments:

Benjii.me, on their website, describes the purpose of this file quite well & as follows:

"First, let's take a look at what runs on startup. You'll find a new file ~/App_Start/Startup.Auth.cs. This file contains basic configuration for your authentication system, and allows you to specify which type you want to use. In our case we're looking for Cookie Based Authentication.

Here's a peek at what the file looks like by default. {Actually slightly different now as per below – also see green comments for further understanding}.

```
using System;
using Microsoft.AspNet.Identity;
using Microsoft.AspNet.Identity.Owin;
using Microsoft.Owin;
using Microsoft.Owin.Security.Cookies;
using Microsoft.Owin.Security.Google;
using Owin;
using Yerra_MVC5.Models;
```

```

namespace Yerra_MVC5

{public partial class Startup
{
    // For more information on configuring authentication, please visit
    http://go.microsoft.com/fwlink/?LinkId=301864
    public void ConfigureAuth(IAppBuilder app)
    {
        // Configure the db context, user manager and signin manager to use a
        single instance per request
        app.CreatePerOwinContext(ApplicationDbContext.Create);

        app.CreatePerOwinContext<ApplicationUserManager>(ApplicationUserManager.Create);

        app.CreatePerOwinContext<ApplicationSignInManager>(ApplicationSignInManager.Create);

        // Enable the application to use a cookie to store information for the
        signed in user
        // and to use a cookie to temporarily store information about a user
        logging in with a third party login provider
        // Configure the sign in cookie
        app.UseCookieAuthentication(new CookieAuthenticationOptions
        {
            AuthenticationType = DefaultAuthenticationTypes.ApplicationCookie,
            LoginPath = new PathString("/Account/Login"),
            Provider = new CookieAuthenticationProvider
            {
                // Enables the application to validate the security stamp when the
                user logs in.
                // This is a security feature which is used when you change a
                password or add an external login to your account.
                OnValidateIdentity =
                    SecurityStampValidator.OnValidateIdentity<ApplicationUserManager, ApplicationUser>(
                        validateInterval: TimeSpan.FromMinutes(30),
                        regenerateIdentity: (manager, user) =>
                            user.GenerateUserIdentityAsync(manager))
            }
        });
        app.UseExternalSignInCookie(DefaultAuthenticationTypes.ExternalCookie);

        // Enables the application to temporarily store user information when they
        are verifying the second factor in the two-factor authentication process.
        app.UseTwoFactorSignInCookie(DefaultAuthenticationTypes.TwoFactorCookie,
        TimeSpan.FromMinutes(5));

        // Enables the application to remember the second login verification
        factor such as phone or email.
        // Once you check this option, your second step of verification during the
        login process will be remembered on the device where you logged in from.
        // This is similar to the RememberMe option when you log in.

        app.UseTwoFactorRememberBrowserCookie(DefaultAuthenticationTypes.TwoFactorRememberBrowserCookie);

        // Uncomment the following lines to enable logging in with third party
        login providers
        //app.UseMicrosoftAccountAuthentication(
        //    clientId: "", "████████")
        //    clientSecret: "");;

        //app.UseTwitterAuthentication()
    }
}

```

```

    // consumerKey: "",  

    // consumerSecret: "");  
  

    //app.UseFacebookAuthentication(  

    //    appId: "",  

    //    appSecret: "");  
  

    //app.UseGoogleAuthentication(new GoogleOAuth2AuthenticationOptions())  

    //{
    //    ClientId = "",  

    //    ClientSecret = ""  

    //});  

}  

}

```

Alright, looks like cookie based authentication is there by default, excellent."

Source:<http://benjii.me/2014/02/mvc-5-membership-and-user-authentication-whats-new-and-getting-started/>

Also, Lin a Stackoverflow user/developer describes this in more detail and offers links to key parts as follows:

"Every OWIN application has a startup class where you specify components for the application pipeline.

If you start a new Visual Studio project, you'll see pieces of OWIN in it. OWIN is a specification that defines an API for framework and servers to cooperation. The point of OWIN is to decouple server and application. For example, ASP.NET Identity uses OWIN security, SignalR self hosting uses OWIN hosting, and etc., the examples all use OWIN, therefore they all need to have a startup class, that is defined in "Startup.cs" file.

The Global.asax, the ASP.NET application file, is an optional file that contains code for responding to application-level events raised by ASP.NET or by HttpModules.

For more details:

OWIN

<http://www.asp.net/aspnet/overview/owin-and-katana>

Global.asax

[http://msdn.microsoft.com/en-us/library/1xaas8a2\(v=vs.71\).aspx](http://msdn.microsoft.com/en-us/library/1xaas8a2(v=vs.71).aspx)

You can find more ideas about why OWIN in the following article:

[http://www.asp.net/aspnet/overview/owin-and-katana/an-overview-of-project-katana"](http://www.asp.net/aspnet/overview/owin-and-katana/an-overview-of-project-katana)

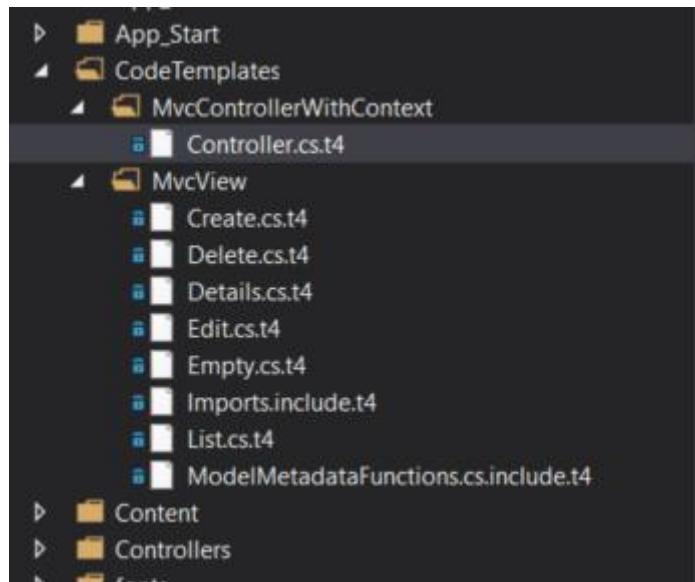
Source: <http://stackoverflow.com/questions/20034105/what-is-the-new-startup-cs-file-for-in-visual-studio-2013-projects>

You will see the Owin namespaces at the beginning of the previous piece of code above – please read the complete Stackoverflow hyperlink directly above.

Breakpointing:

If we begin the breakpointing at `public void ConfigureAuth(IAppBuilder app)` **Startup.Auth.cs** the flow passes through all the source code methods (in that file) & then moves onto **Startup.cs** and then loads the **Yerra Clearly Log In screen**.

2.3 Code Templates Folder - cs.t4 Class files



Introduction:

This article provides a solid description of these file types as follows: "*T4 is a general-purpose templating engine you can use to generate C# code, Visual Basic code, XML, HTML, or text of any kind. The first step in using T4 is to create a template file, which is a mix of literal text, code, and processing directives. The T4 template processor will transform this file into a text file of the desired type by executing the code and processing directives inside. As an example, you can right-click on any project in Visual Studio 2010 and add a new item of type Text Template.*"

Source: <https://msdn.microsoft.com/en-us/data/gg558520.aspx>

Developer comments:

"I have no idea what those files are, possibly created as temp files by the VS IDE but I'm speculating."

Breakpointing:

This does not provide any feedback suggesting these files are not being used by the YerraClearly system.

2.4 Content Folder/ CSS files & Images

Introduction:

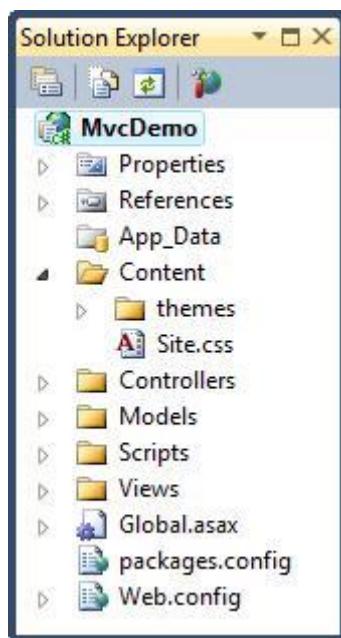
"The **Content** folder is used for static files like style sheets (css files), icons and images.

Visual Web Developer automatically adds a **themes** folder to the Content folder. The themes folder is filled with jQuery styles and pictures."

Also,

"Visual Web Developer also adds a standard style sheet file to the project: the file **Site.css** in the content folder. The style sheet file is the file to edit when you want to change the style of the application."

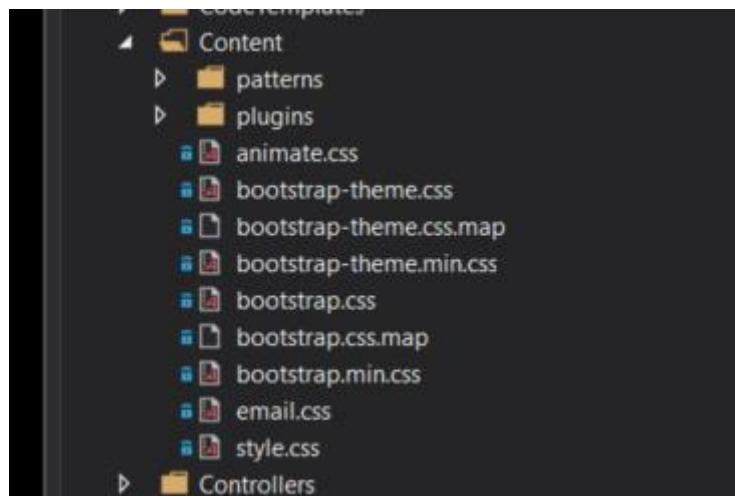
Here is an image of the generic Contents folder:



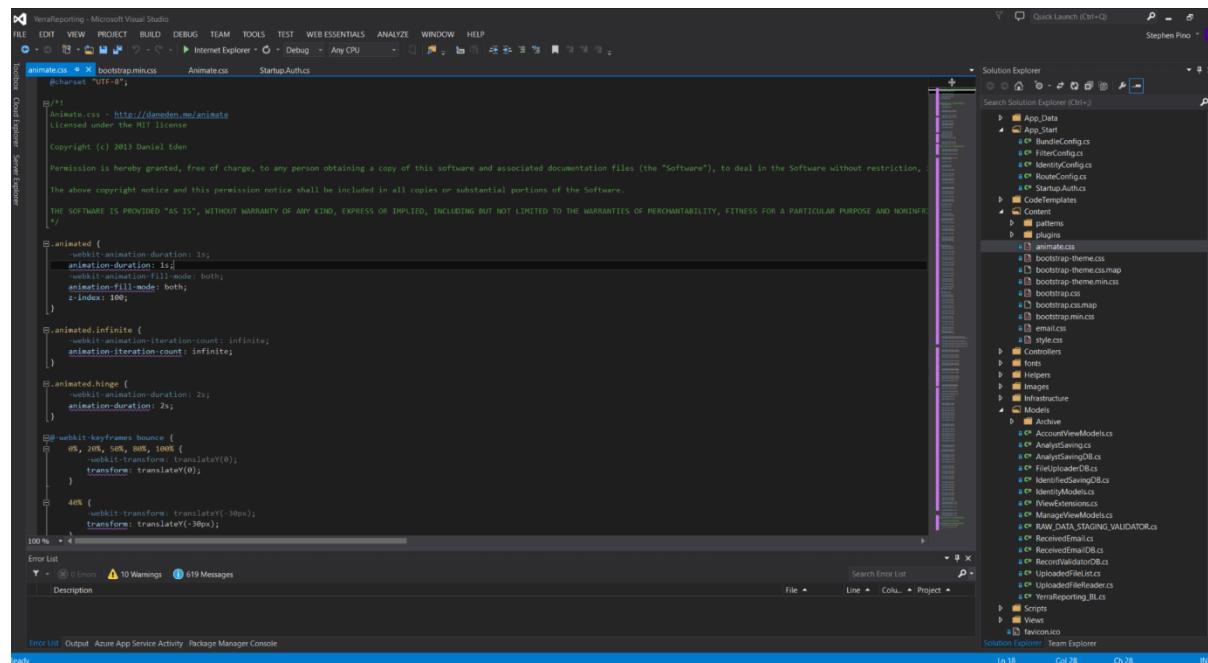
Source: http://www.w3schools.com/aspnet/mvc_folders.asp

*Note: The style sheet file (Site.css) is the generic file but is not used in the Yerra Clearly application.

The Yerra Clearly Contents folder looks like this:

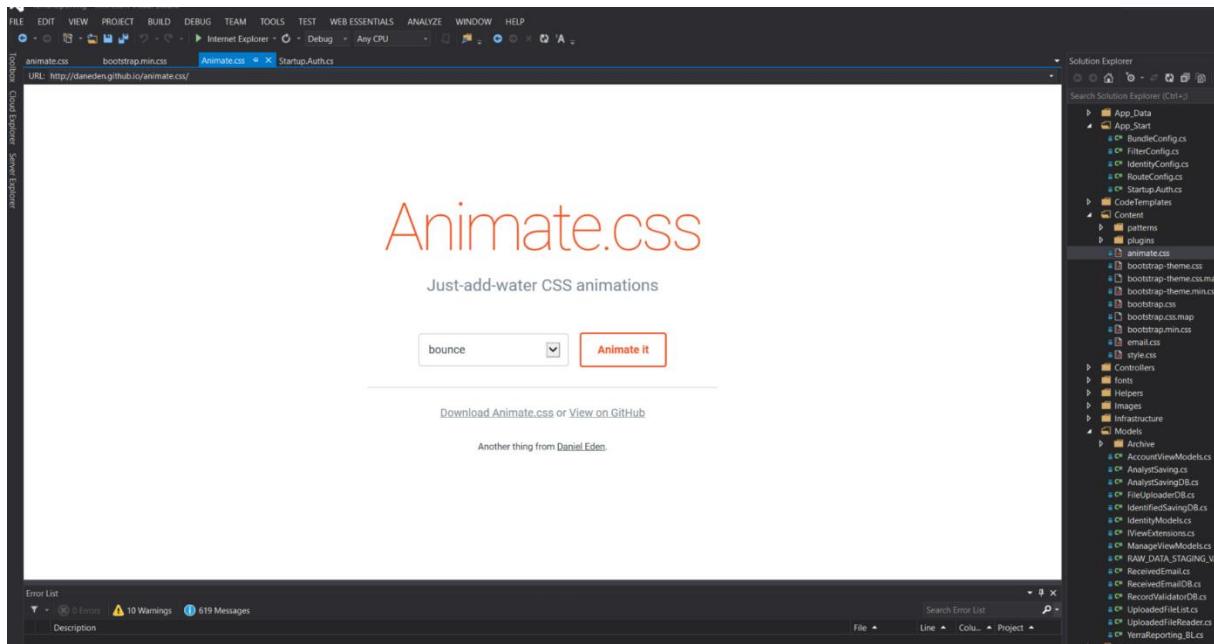


The `animate.css` is self-explanatory whereas it provides the Yerra Clearly app with front end UI animations applied to any text.



The file is downloaded from the following website and implemented into any website for predefined cross browser CSS animation styling.

See the `Animate.css` website by Daniel Eden below (notice the CSS files can also be accessed on Github):



The Github CSS files can be searched for through the file URL hyperlink and via Visual Studio:

Source: <http://daneden.github.io/animate.css/>

The email.css file:

```

/*
 * What follows is the result of much research on cross-browser styling.
 * Credit left inline and big thanks to Nicolas Gallagher, Jonathan Neal,
 * Kroc Camen, and the HSBP dev community and team.
 */

/* =====
 * Base styles: opinionated defaults
 * ===== */
html {
  color: #222;
  font-size: 1em;
}

```

The screenshot shows a GitHub repository page for 'h5bp/html5-boilerplate'. The 'main.css' file is displayed, which contains the CSS for 'email.css'. The code is well-commented, explaining the styling approach and credits. The GitHub interface includes a navigation bar, a search bar, and a sidebar showing the project's structure.

Source: <https://github.com/h5bp/html5-boilerplate/blob/master/src/css/main.css>

Developer comments:

Jakub Konecki, a .Net Contractor at ServiceWare Ltd comments on Stackoverflow as follows:

*"It's a good approach to add all your files/resources to the solution (at least you can see them in VS ;-)
Other developers won't be surprised that your solution really needs some other files.*

If those files are the 'content' of your website (like css files) then you can store them in the Content folder. It is up to you what folder structure you come up with. For example, you might want to store your images in folder related to the modules of your application, or just all in the images folder. MVC doesn't enforce any particular way of keeping your resource files.

Just leave the Build Action set to Content and Copy to Output Directory to 'Do not copy'.

You might want to take a look at [T4MVC](#) project (also available as NuGet package) which will help you in keeping urls to your images / files compile-time safe."

Matt Woodward a developer on Stackoverflow speaks of routing and how it may interact with the Contents folder simply by saying: *"No magic, the System.Web.Routing.RouteCollection class has a property RouteExistingFiles which controls the behavior.*

The default is false, which means ASP Routing should not route the URL, but just return the default content. In this case the "/Content/all.min.css?v=251" skips the MVC routing rules entirely.

if you want to add a routing rule for the content folder, you need to add the rule, and set RouteExistingFiles to true."

Source: <http://stackoverflow.com/questions/575640/is-the-content-folder-sacred-in-asp-net-mvc>

The majority of the CSS files in the Contents folder contain links to their respective URL a great deal of which will point you to Github or their respective website – simply by holding CTRL + Click.

Typically, images are also stored in this folder in an images or img folder as per Yerra Clearly.

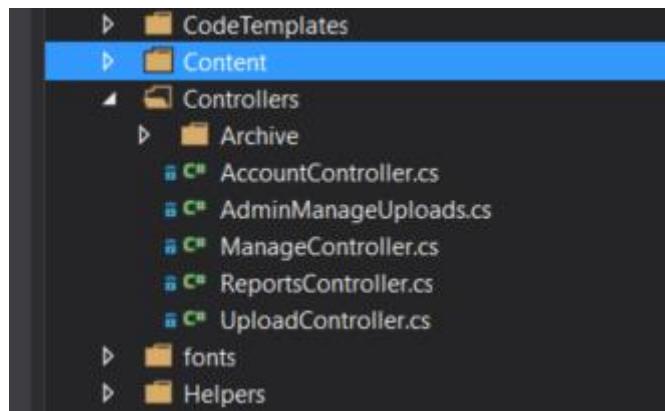
Breakpointing:

Not required in this file type.

2.5 Controllers Folder

Introduction:

To begin with here is an image of the Yerra Clearly Controllers folder:



The Generic Controllers folder only contains the Account, Home and Manage controllers respectively. It does not contain an Archive folder either, that was set up purely for the purposes of the Yerra Clearly App.

Developer comments:

Yerra Junior developer asks: *"It appears that the AccountController & ManageController both remain generic is that so?"*

Yerra Lead developer responds: *"Largely but with a few tweaks specific to what the application does."*

Also for the front login system but is a good example for the purpose of these files:

*"In the code you can review the **AccountDB** within the models folder, **Account Views** and the **AccountController** for the login screen. It also requires a connection within web.config that tells the application where to store the user details.*

Note - creating a login system is fairly advanced MVC but there are many tutorials on it including within the online books available on safari."

This could do with a much deeper explanation but all we have at this point in time.

Candice Zhou on the ASP.Net forum describes the purpose for these generic class files:

*"**AccountController** and **ManageController** controller contains the sample code for Membership and Authorization which provided by default on project creation because its common stuff for web application and its help full to all for beginners, intermediate and experts also. From those controller you can manage/ create Role Based application more easily and quickly."*

Source: <http://forums.asp.net/t/2080557.aspx?Account+vs+Manage+Controller+in+ASP+NET+5>

Web developer Simone Chiaretta suggests in his Website blog listing his 12 best practices for ASP.Net MVC:

“1 – Delete the AccountController

You will never use it and it’s a super-bad practice to keep demo code in your applications.”

Source: <http://codeclimber.net.nz/archive/2009/10/27/12-asp.net-mvc-best-practices.aspx>

However, Liam comments on Simone’s blog by posting this reply: *“I’m not sure that Best Practice #1 is enough justification in itself. Why would I never use AccountController? If I do use the existing AccountController am I breaking best practices? It seems to function just fine for a simple login and registration page.”*

These comments in itself show the flexibility of MVC and the generic Class files for developers decision making. Basically, you can do what you want as long as you know what you are doing. The generic HomeController.cs Class file contains methods that can receive response from the server: *“The controller file in our application **HomeController.cs**, defines the two controls **Index** and **About**.”*

Also,

*“The files **Index.cshtml** and **About.cshtml** in the Views folder defines the ActionResult views Index() and About() in the controller.”*

Source: http://www.w3schools.com/aspnet/mvc_controllers.asp

However, the Yerra Clearly application does not use the generic **HomeController**.

2.5.1 Archive folder

Developer comments:

Yerra Junior developer question: *“Do the Controllers in the Archive folder still have any purpose in the application?”*

Lead developer response: *“No, just legacy files that I may refer to.”*

2.5.2 AdminManageUploads.cs

Developer comments:

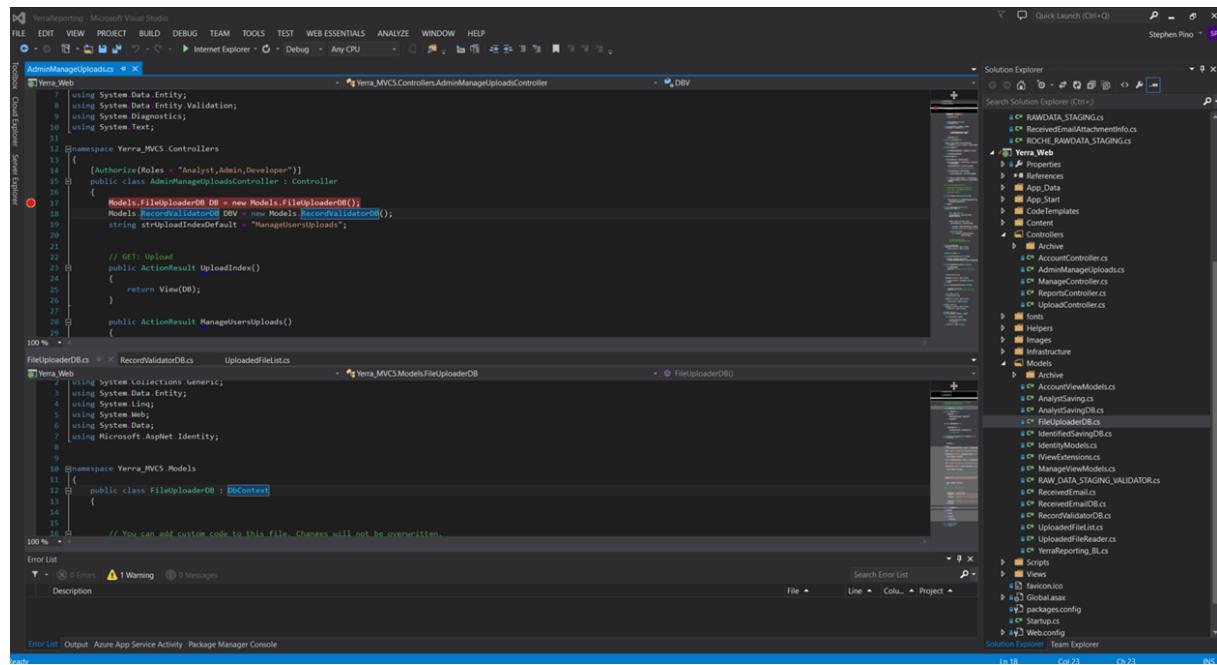
Here we will use the very first objects and method as an example: *“AdminManageUploads.cs – break-pointing through this Class appears to cycle through the following Javascript files:*

JQuery-2.1.1.min.js – Bootstrap.min.js – MetisMenu.min.js – Pace.min.js – blank[dynamic] – Pace.min.js – Yerra Clearly Login screen – Pace.min.js.

If we breakpoint through these Class Objects they cycle through the above files:

```
namespace Yerra_MVC5.Controllers
{
    [Authorize(Roles = "Analyst,Admin,Developer")]
}
```

```
public class AdminManageUploadsController : Controller
{
    Models.FileUploaderDB DB = new Models.FileUploaderDB();
    Models.RecordValidatorDB DBV = new Models.RecordValidatorDB();
    string strUploadIndexDefault = "ManageUsersUploads";
```



{Yerra junior developer} Please could you please expand on what the relationships are here? I have used **FileUploaderDB** Class as an example and in image above.

{Lead developer} **Apologies not sure what you asking here? What is the purpose of the javascript files? Mainly for bootstrap but I think you mean something else..?**

Junior developer continues to ask question as follows: “What controller class handles the request from the server enabling it to handle the extraction of these files to database tables RAWDATA_STAGING and then to update dbo.ReceivedEmailsAttachmentsInfo table in SQL Server?”

Lead developer responds: “In terms of uploads it uses two controllers - **UploadController** and **AdminManageUploads** depending on who is using it.”

Breakpointing:

Needs doing.

2.5.3 ReportsController.cs

Developer comments:

Junior developer asks: "I am concentrating on routes right now and as you suggested. I now understand the importance of this the more I learn.

I see the link between the routes in **RouteConfig.cs** class file and the **ReportsController.cs** class file. The method parameters are also passed into the URL route path."

RouteConfig:

```
routes.MapRoute(
    name: "ViewWeeklyDetailsForPhase",
    url: "Reports/ViewWeeklyDetailsForPhase/{USER_1}/Year/{calendar_year}/Week/{calendar_week}"
    , [REDACTED]
        //defaults: new { controller = "Dashboards", action = "Dashboard_1",
        id = UrlParameter.Optional } [REDACTED]
        defaults: new { controller = "Reports", action =
    "ViewWeeklyDetailsForPhase", USER_1 = "", calendar_year = @"\d+", calendar_week =
    @"\d+" } [REDACTED]
    );

```

ReportsController:

```
public ActionResult ViewWeeklyDetailsForPhase(string USER_1, int? calendar_year,
int? calendar_week)
{
    [REDACTED]
    Yerra_MVC5.Models.YerraReporting_BL y_bl = new Models.YerraReporting_BL();
    if (USER_1 == null || calendar_week == null || calendar_year == null)
    {
        [REDACTED]
        return View("Index", new Yerra_MVC5.Models.YerraReporting_BL());
    }
    [REDACTED]
    y_bl.ViewDetailsInformation.dtFirst_Calendar_Month=GeneralSupport.SF.First
DateOfWeekISO8601((int)calendar_year, (int)calendar_week);
    y_bl.ViewDetailsInformation.User_1=USER_1;
    return View(y_bl);
}
```

Lead developer response: "both calendar_year and calendar_week are allowed "nullables" - this means that it is possible for the method to accept no parameters for calendar_year, calendar_week.

In terms of "FirstDateOfWeekISO8601", this is a handy function that returns the date based on the first day of the given week number and year. This is then passed to the main business logic class for reports for consumption by whatever tables/properties the ViewWeeklyDetailsForPhase *view* requires for it.

Hope that makes sense!"

Breakpointing:

Here is an example of the types of methods held within this file:

```
[Authorize(Roles = "Analyst,Admin,Developer")]
public ActionResult Validation_ViewAllLateInvoicesForVendor(string
LAW_FIRM_NAME)
{
    Yerra_MVC5.Models.YerraReporting_BL y_bl;
    if (LAW_FIRM_NAME==null)
    {
        return View("Index", new Yerra_MVC5.Models.YerraReporting_BL());
    }
    y_bl = new Yerra_MVC5.Models.YerraReporting_BL();
    y_bl.ViewDetailsInformation.LAW_FIRM_NAME = LAW_FIRM_NAME;
    return View(y_bl);
}
```

Associated route in RouteConfig.cs:

```
routes.MapRoute(
    name: "Validation_ViewAllLateInvoicesForVendor",
    url: "Reports/Validation_ViewAllLateInvoicesForVendor/{LAW_FIRM_NAME}",
    //defaults: new { controller = "Dashboards", action = "Dashboard_1",
id = UrlParameter.Optional }
    defaults: new { controller = "Reports", action =
"Validation_ViewAllLateInvoicesForVendor", LAW_FIRM_NAME = ""})
);
```

2.5.4 UploadController.cs

Introduction:

No comments at this time.

Code example from file:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using Microsoft.AspNet.Identity;
using System.Data.Entity;
using System.Data.Entity.Validation;
using System.Diagnostics;
using System.Text;

namespace Yerra_MVC5.Controllers
{
    [Authorize(Roles = "Analyst,Admin,Developer")]
    public class UploadController : Controller
    {
        Models.FileUploaderDB DB = new Models.FileUploaderDB();
        Models.RecordValidatorDB DBV = new Models.RecordValidatorDB();
        string strUploadIndexDefault = "UploadIndex";

        // GET: Upload
        public ActionResult UploadIndex()
        {
```

```

        return View(DB);
    }

    // GET: Upload
    public ActionResult ArchiveIndex()
    {
        return View(DB);
    }

    [HttpPost]
    public ActionResult PublishFile(int id)
    {
        Models.UploadedFileList ufl_instance = new Models.UploadedFileList();
        ReturnType rt = ReturnType.True;
        rt = ufl_instance.PublishFile(id);
        return PartialView("_PartialValidationSummary", DB);
    }
}

```

Developer comments - ReturnType rt enum:

Junior developer asks: *"The rt variable is an enum what is this for? does it return an error number?"*

Lead developer: *"ReturnType is a special type I created a long time ago as the normal bool of true/false was not suitable enough for error checking."*

So then... it returns several responses - True, False, Error, ValidationError.

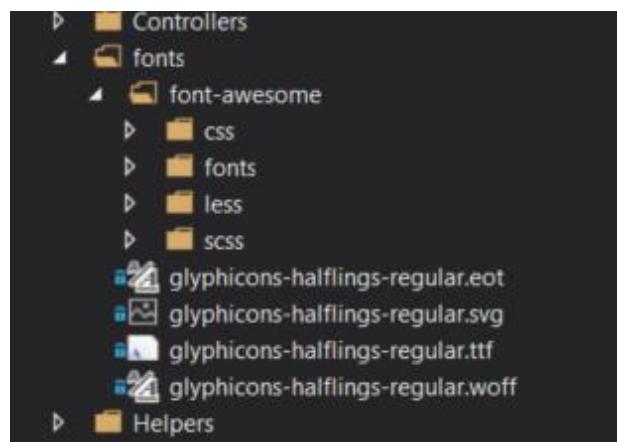
The reason for this is that False may just mean it did not return a value but doesn't necessarily mean an error. If it does return an error, I normally then call a routine I created (within the same GeneralSupport module) that generates an exception error message that can be displayed on client or stored somewhere. I've made it open ended allowing you to link it with any storage type. So for example, if any exceptions are generated it is automatically stored within the SQL database (in this case, a table called dbo.Exceptions). The routine automatically stores the method name where the error was generated automatically without the need to manually put a string name of the method in the exception.

Every class I create is a descendant of "mybase" (something else created a long time ago) which has a few built in properties such as exception, validation error etc making it easier to debug code if you do not have the luxury of a debugger but know where the errors have been stored."

Break-pointing:

Not available at this time.

2.6 Fonts folder



Introduction:

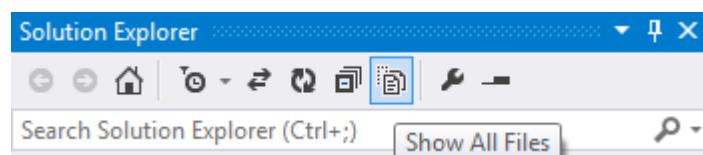
Fonts folder contains custom font files for your application.

Jumpingcode a developer on stackoverflow describes how to add files to the font folder for use in the program:

"You can either add an existing item:

1. Right click the folder you want to add the files
2. Hover over Add
3. Select Existing item
4. Select the file(s) you would like to add.

Alternatively, if you've already done it the way you have, you need to add them to the solution file. You can do this manually (bothersome, could end up screwing it up), or, in solution explorer you will see there are icons at the top, there is one that shows all files:



Now if you navigate to your folder in the solution explorer, you can right click this items and select Include in project"

2.6.1 Fonts

Introduction:

Font frameworks such as "Font Awesome gives you scalable vector icons that can instantly be customized — size, color, drop shadow, and anything that can be done with the power of CSS."

Source: <http://fontawesome.io/>

2.6.2 CSS

Introduction:

To start with "The style sheet for the application is called Site.css. It is located in the Content folder." This can be modified to suit your needs.

Source: http://www.w3schools.com/aspnet/mvc_layout.asp

If you do not know what CSS is: “**CSS** stands for **Cascading Style Sheets**. **CSS** describes how HTML elements are to be displayed on screen, paper, or in other media. **CSS** saves a lot of work. It can control the layout of multiple web pages all at once.”

Source: http://www.w3schools.com/css/css_intro.asp

2.6.3 Sass

Introduction:

“CSS on its own can be fun, but stylesheets are getting larger, more complex, and harder to maintain. This is where a preprocessor can help. Sass lets you use features that don't exist in CSS yet like variables, nesting, mixins, inheritance and other nifty goodies that make writing CSS fun again.”

Source: <http://sass-lang.com/guide>

2.6.4 Less

Introduction:

Some “Less” files are also used in the Yerra Clearly application. To describe these file types: “Less is a CSS pre-processor, meaning that it extends the CSS language, adding features that allow variables, mixins, functions and many other techniques that allow you to make CSS that is more maintainable, themable and extendable.”

Source: <http://lesscss.org/>

2.6.5 Glyphicons

Introduction:

These file types are also used in the Yerra Clearly application as part of the Bootstrap framework: “**Glyphicons** are icon fonts which you can use in your web projects. **Glyphicons Halflings** are not free and require licensing, however their creator has made them available for Bootstrap projects free of cost.”

Clicking on the source hyperlink below will take you directly to the Bootstrap Glyph page.

Source: <http://getbootstrap.com/components/>

2.7 Helpers folder

Introduction:

No comments.

2.7.1 HMTLHelperExtensions.cs

“After you create an extension method, and build your application successfully, the extension method appears in Visual Studio Intellisense like all of the other methods of a class (see Figure 2). The only difference is that extension methods appear with a special symbol next to them (an icon of a downward arrow).” Please follow the below link and complete the tutorial:

Source: <http://www.asp.net/mvc/overview/older-versions-1/views/creating-custom-html-helpers-cs>

Developer comments:

Junior developer: “Please can you provide me with your understanding of the **HMTLHelperExtensions.cs** file and its purpose in Yerra Clearly app?”

A generic ASP.net file doesn't have this so I am not entirely sure about it. I realise it is linked to the HtmlHelper Class & it is used to help with the Views files to take out the need for excessive HTML tags? I notice Two static methods present.

File source code:

```
namespace Yerra_MVC5
{
    public static class HTMLHelperExtensions
    {
        public static string IsSelected(this HtmlHelper html, string controller =
null, string action = null)
        {
            string cssClass = "active";
            string currentAction =
(string)html.ViewContext.RouteData.Values["action"];
            string currentController =
(string)html.ViewContext.RouteData.Values["controller"];

            if (String.IsNullOrEmpty(controller))
                controller = currentController;

            if (String.IsNullOrEmpty(action))
                action = currentAction;

            return controller == currentController && action == currentAction ?
cssClass : String.Empty;
        }

        public static string PageClass(this HtmlHelper html)
        {
            string currentAction =
(string)html.ViewContext.RouteData.Values["action"];
            return currentAction;
        }
    }
}
```

Thanks,"

Lead developer: "*can't say I'm using it for anything other than the general framework which I'm using.*

Basically it's a utility class that can be re-used as opposed to typing in the same routine in every controller you need it for. It's equivalent to the GeneralSupport library I created except this is more for MVC"

More information can be found here: [https://msdn.microsoft.com/en-us/library/microsoft.web.mvc.html.htmlhelperextensions\(v=vs.118\).aspx](https://msdn.microsoft.com/en-us/library/microsoft.web.mvc.html.htmlhelperextensions(v=vs.118).aspx)"

2.8 Images folder

Introduction:

This file holds images that are used in the views of any web application. It also holds Widgets, logos, map images etc.

2.9 Infrastructure folder

Introduction:

These files are set up by the developer and so are not generic.

Developer comments:

Junior developer: "The infrastructure folder holds 2 classes LocalEnum.cs and UploadHelpers.cs – see attached image.

Could you provide a background on these 2 files please – how are they used in the system?"

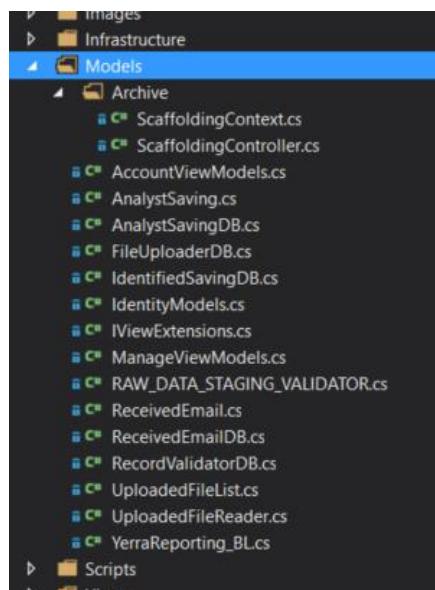
Lead developer: "they are used by the validation module to make it easier to work out the status of an uploaded file.

In the database it is represented by numbers (1 to 4 etc) which is the standard way of doing things, however when working out the status in the code front end it can be a little difficult to remember what each number means so you often create enums to map a "readable" world against a numeric value and then use that to create logic.

The upload file class is exactly that, it is the controller class that reads the file for uploading.

Also, did you know that if you right click a value or a definition or a method you can then select "Find all references" and it will show you every line that uses that item you have selected on. Great for learning how things work as you can see it in action where it is being used."

3.0 Models folder



Introduction:

Above is an example of the Yerra Clearly Models folder which is different to the generic Models folder. Also: "The Models folder contains the classes that represent the application models. Models hold and manipulate application data."

Source: http://www.w3schools.com/aspnet/mvc_folders.asp

3.1 Yerra_Reporting_BL.cs:

Introduction:

These files are set up by the developer and so are not generic.

This file has over 5531 lines of code which is not acceptable. This file needs modulated, cleaned up and comments added. To go through this entire file is unrealistic so we speak to the creator and obtain as much information about the source code as possible.

Developer comments:

Lead developer: *"The most important file to evaluate is **Yerra_Reporting_BL** - this contains a vast amount of logic for the reports and is the core for most of the system. ideally when in house you perhaps consider breaking it down into more modular units based on report context?"*

Junior developer: *"You have already given me a short but detailed explanation on this important class file, however, is there anymore pointers you can give me about it? Just trying to learn as much as possible about it."*

For example using shift + F12 on `[] ColorSequence` links to `GetDonutData()` method = Is this adding 1 and changing colour at same time to timekeeper list and appending to view page?

```
namespace Yerra_MVC5.Models
{
    public class YerraReporting_BL : GeneralSupport.myBaseObject
    {
        private string[] ColorSequence = new string[] {"#4D4D4D",
"#"5DA5DA",
"#"FAA43A",
"#"60BD68",
"#"F17CB0",
"#"B2912F",
"#"B276B2",
"#"DECF3F",
"#"F15854",
"#"b5b8cf"};
    }

    public string GetDonutData()
    {
        System.Text.StringBuilder sb = new System.Text.StringBuilder();
        if (tkList != null)
        {
            for (int iCount = 0; iCount < tkList.Count; iCount++)
            {
                sb.Append("\r\n");
                sb.Append("value: " + tkList[iCount].TimeKeeper_Amount.ToString());
                sb.Append(",\r\n");
                sb.Append("color: " + ColorSequence[iCount % 10] + "\r\n");
                sb.Append("highlight: "#1ab394\r\n");
                sb.Append("label: " + tkList[iCount].TimeKeeper_FirstName + " "
+ tkList[iCount].TimeKeeper_Surname + "\r\n");
                if (iCount == tkList.Count - 1)
                {
                    sb.Append("}\r\n");
                }
                else
                {

```

```

        sb.Append("},\r\n");
    }
    // value: 300,
    // color: "#a3e1d4",
    // highlight: "#1ab394",
    // label: "App"
}
return sb.ToString();
}

```

Thanks"

Lead developer responds: *"not sure that colour sequence is used any more but it powers the JavaScript/JQuery charting stuff (such as pie charts etc) so the colours are always consistent. The main one is duotone and whatever class is right next to that one.*

GetDonutData is used by a few charts within the system where it builds a JSON data string based on the data it pulls down from the database. To be fair this is something I originally created for the application but have since then did a lot more "inline" coding within the views which is the standard way of doing things. At some point I would like to replace the function (where it is being used) with inline view razor code."

Junior developer: *"I am going spend some time in this class file, as you said it is an important one.*

I am looking at these Class parameters for patterns and repetition in the source code file you wrote. When I shift + F12 some properties remain in this class file and some are inherited into other model class files? I think inherited is the correct term here.

I see some are used to pass in from a Method's parameters.

Source code:

```

public class ViewDetailsParameters
{
    public int ID = -1;
    public string Code_LU = "";
    public string CodeType = "";
    public DateTime dtFirst_Calendar_Month; // Legacy but still in use by most
functions
    public DateTime dtStart_Date = new DateTime(1900,1,1); // New for Date
Range Functions
    public DateTime dtEnd_Date = DateTime.MaxValue; // New for Date Range
Functions
    public string User_1 = "";
    public string User_2 = "";
    public string TimeKeeper_Name = "";
    public string LAW_FIRM_NAME = "";
    public string Country_Name = "";
    public string CLIENT_MATTER_ID = "";
    public string CLIENT_CITY = "";
    public string IPM_CASE_RECORD_NUMBER = "";
    public string IPM_RESPONSIBLE_ATTORNEY = "";
    public string IPM_CASE_STATUS = "";
    public string IPM_CASE_SUBSTATUS = "";
    public string Spare1 = "";
    public string Spare2 = "";
    public string Spare3 = "";
    public string Spare4 = "";
    public ReportingPeriod ReportingPeriod = ReportingPeriod.Quarterly;
}

```

```
[ ]}
```

Code flow for each property above shows where being used in the system:

ID =	<i>YerraReporting_BL.cs</i>
Code_LU =	<i>Reports Controller.cs & YerraReporting_BL.cs</i>
CodeType =	<i>Reports Controller.cs & YerraReporting_BL.cs</i>
dtFirst_Calendar_Month =	<i>Reports Controller.cs & YerraReporting_BL.cs</i>
dtStart_Date =	<i>Reports Controller.cs & YerraReporting_BL.cs &</i>
<i>AdminManageUploads.cs</i>	
dtEnd_Date =	<i>Reports Controller.cs & YerraReporting_BL.cs &</i>
<i>AdminManageUploads.cs</i>	
User_1 =	<i>Reports Controller.cs & YerraReporting_BL.cs</i>
User_2 =	<i>YerraReporting_BL.cs</i>
TimeKeeper_Name =	<i>Reports Controller.cs & YerraReporting_BL.cs</i>
LAW_FIRM_NAME =	<i>Reports Controller.cs & YerraReporting_BL.cs</i>
Country_Name =	<i>Controller.cs & YerraReporting_BL.cs</i>
CLIENT_MATTER_ID =	<i>Reports Controller.cs & YerraReporting_BL.cs</i>
CLIENT_CITY =	<i>Reports Controller.cs & YerraReporting_BL.cs</i>
IPM_CASE_RECORD_NUMBER =	<i>Reports Controller.cs & YerraReporting_BL.cs</i>
IPM_RESPONSIBLE_ATTORNEY =	<i>Reports Controller.cs & YerraReporting_BL.cs</i>
IPM_CASE_STATUS =	<i>Reports Controller.cs & YerraReporting_BL.cs</i>
IPM_CASE_SUBSTATUS =	<i>Reports Controller.cs & YerraReporting_BL.cs</i>
Spare1 =	<i>Reports Controller.cs & YerraReporting_BL.cs</i>
Spare2 =	<i>Reports Controller.cs & YerraReporting_BL.cs</i>
Spare3 =	<i>YerraReporting_BL.cs</i>
Spare4 =	<i>YerraReporting_BL.cs</i>
ReportingPeriod =	<i>Reports Controller.cs & YerraReporting_BL.cs</i>

Also, where does the (-21) days factor here? Just trying to understand nuggets of logic.

```
[HttpGet]
public ActionResult ReceivedInvoiceLog()
{
    //if (User.IsInRole("Admin") || User.IsInRole("Developer"))
    //{
        Yerra_MVC5.Models.YerraReporting_BL y_b1 = new Models.YerraReporting_BL();
        y_b1.ViewDetailsInformation.dtStart_Date = DateTime.Now.AddDays(-21);
        y_b1.ViewDetailsInformation.dtEnd_Date = DateTime.Now;
        return View(y_b1);
    //}
    //else
    //{
        //    return RedirectToAction("UserIndex");
    //}
}
```

Also, is it safe to say there is repetition in the code as per below seem to be 2 common methods 1. ReturnType & 2. Datatable:

```
private ReturnType LoadViewPatentSpendForVendorAndPatentCode(ref
System.Data.DataTable _tblPatentSpendForVendorAndPatentCode, DateTime
dtFirstCalendarMonth)
{
    ReturnType rt = ReturnType.False;
```

```

        DALController.CustomQuery cq = new DALController.CustomQuery();

        cq.vDirectSPName =
"Invoices_ViewDetailsPatentSpendForVendorAndPatentCode";
        cq.vParameterList.Add("@LAW_FIRM_NAME", System.Data.SqlDbType.VarChar).Value = ViewDetailsInformation.LAW_FIRM_NAME;
        cq.vParameterList.Add("@LINE_ITEM_TASK_CODE", System.Data.SqlDbType.VarChar).Value = ViewDetailsInformation.Code_LU;
        cq.vParameterList.Add("@First_Calendar_Month", System.Data.SqlDbType.Date).Value = dtFirstCalendarMonth;
        rt = cq.LoadDataTable(ref _tblPatentSpendForVendorAndPatentCode);
        if (rt == ReturnType.Error)
        {
            _Exception = GeneralSupport.SF.GEX(cq.Exception, "Unspecified error occurred attempting to use " + cq.vDirectSPName);
        }
        return rt;
    }

    private System.Data.DataTable _tblDetailsByCountryCode = null;

    public System.Data.DataTable tblDetailsByCountryCode
    {
        get
        {
            ReturnType rt = ReturnType.False;
            if (_tblDetailsByCountryCode == null)
            {
                rt = LoadDetailsByCountryCode(ref _tblDetailsByCountryCode, ViewDetailsInformation.dtFirst_Calendar_Month);
            }
            return _tblDetailsByCountryCode;
        }
    }
}

```

Any further advice welcome.

Lead developer: *"If I recall correctly the -21 days was due to the sheer of amount of data for an entire month so -21 is the default. This populates the start / end calendar dates (End = current, start = -21)*

I'm engaged in other things today so my time is brief but will review and respond when I can thanks."

Junior developer: *"If for example we take IPM_CASE_RECORD_NUMBER found in both Reports Controller.cs & YerraReporting_BL.cs. We see*

Reports Controller.cs:

This is outputting data to the view?

```

    public ActionResult IPM_ViewDetailsByResponsibleAttorneyAndCaseRecordNumber(string Responsible_Attorney,
string Case_Record_Number, int? calendar_year, int? calendar_month)
{
    Yerra_MVC5.Models.YerraReporting_BL y_bl = null;
    if (Responsible_Attorney == null || Case_Record_Number == null ||
calendar_month == null || calendar_year == null)
    {
        return View("Index", new Yerra_MVC5.Models.YerraReporting_BL());
    }
}

```

```

        }
        y_bl = new Models.YerraReporting_BL(iCalendar_Year: (int)calendar_year,
iCalendar_Month: (int)calendar_month);
        y_bl.ViewDetailsInformation.IPM_RESPONSIBLE_ATTORNEY =
Responsible_Attorney;
        y_bl.ViewDetailsInformation.IPM_CASE_RECORD_NUMBER = Case_Record_Number;
        return View(y_bl);
    }

    public ActionResult
IPM_ViewDetailsByResponsibleAttorneyCaseRecordNumber(string Responsible_Attorney,
string Case_Record_Number, int? calendar_year, int? calendar_month)
{
    Yerra_MVC5.Models.YerraReporting_BL y_bl = null;
    if (Responsible_Attorney == null || Case_Record_Number == null ||
calendar_month == null || calendar_year == null)
    {
        return View("Index", new Yerra_MVC5.Models.YerraReporting_BL());
    }
    y_bl = new Models.YerraReporting_BL(iCalendar_Year: (int)calendar_year,
iCalendar_Month: (int)calendar_month);
    y_bl.ViewDetailsInformation.IPM_RESPONSIBLE_ATTORNEY =
Responsible_Attorney;
    y_bl.ViewDetailsInformation.IPM_CASE_RECORD_NUMBER = Case_Record_Number;
    return View(y_bl);
}

```

YerraReporting_BL.cs:

```
public string IPM_CASE_RECORD_NUMBER = "";
```

and

This is Loading the data into the data table that gets returned to the view via the above methods?

```

    public ReturnType
Load_IPM_ViewLineItemsDetailsResponsibleAttorneySpendByCaseRecordNumberForQuarter(ref
System.Data.DataTable tblTemp, DateTime dtFirst_Calendar_Month)
{
    ReturnType rt = ReturnType.False;
    DALController.CustomQuery cq = new DALController.CustomQuery();
    cq.vDirectSPName =
"dbo.IPM_ViewLineItemsDetailsResponsibleAttorneySpendByCaseRecordNumberForQuarter";
    cq.vParameterList.Add("First_Calendar_Month",
System.Data.SqlDbType.Date).Value = dtFirst_Calendar_Month;
    cq.vParameterList.Add("ResponsibleAttorney",
System.Data.SqlDbType.VarChar).Value =
ViewDetailsInformation.IPM_RESPONSIBLE_ATTORNEY;
    cq.vParameterList.Add("CaseRecordNumber",
System.Data.SqlDbType.VarChar).Value = ViewDetailsInformation.IPM_CASE_RECORD_NUMBER;
    rt = cq.LoadDataTable(ref tblTemp);
    if (rt != ReturnType.True)
    {
        rt = ReturnType.Error;
        _Exception = GeneralSupport.SF.GEX(cq.Exception, "Error obtaining
data: " + cq.vDirectSPName);
    }
    return rt;
}

```

```

    public ReturnType
Load_IPM_ViewDetailsResponsibleAttorneyFeeSpendByCaseRecordNumberForQuarter(ref
System.Data.DataTable tblTemp, DateTime dtFirst_Calendar_Month)
{
    ReturnType rt = ReturnType.False;
    DALController.CustomQuery cq = new DALController.CustomQuery();
    cq.vDirectSPName = [REDACTED]
"dbo.IPM_ViewDetailsResponsibleAttorneyFeeSpendByCaseRecordNumberForQuarter";
    cq.vParameterList.Add("First_Calendar_Month", [REDACTED]
System.Data.SqlDbType.Date).Value = dtFirst_Calendar_Month;
    cq.vParameterList.Add("ResponsibleAttorney",
System.Data.SqlDbType.VarChar).Value =
ViewDetailsInformation.IPM_RESPONSIBLE_ATTORNEY;
    cq.vParameterList.Add("CaseRecordNumber",
System.Data.SqlDbType.VarChar).Value = ViewDetailsInformation.IPM_CASE_RECORD_NUMBER;
    rt = cq.LoadDataTable(ref tblTemp);
    if (rt != ReturnType.True)
    {
        [REDACTED]
        rt = ReturnType.Error;
        _Exception = GeneralSupport.SF.GEX(cq.Exception, "Error obtaining
data: " + cq.vDirectSPName);
    }
    return rt;
}

    public ReturnType
Load_IPM_ViewDetailsResponsibleAttorneyExpenseSpendByCaseRecordNumberForQuarter(ref
System.Data.DataTable tblTemp, DateTime dtFirst_Calendar_Month)
{
    ReturnType rt = ReturnType.False;
    DALController.CustomQuery cq = new DALController.CustomQuery();
    cq.vDirectSPName = [REDACTED]
"dbo.IPM_ViewDetailsResponsibleAttorneyExpenseSpendByCaseRecordNumberForQuarter";
    cq.vParameterList.Add("First_Calendar_Month", [REDACTED]
System.Data.SqlDbType.Date).Value = dtFirst_Calendar_Month;
    cq.vParameterList.Add("ResponsibleAttorney",
System.Data.SqlDbType.VarChar).Value =
ViewDetailsInformation.IPM_RESPONSIBLE_ATTORNEY;
    cq.vParameterList.Add("CaseRecordNumber",
System.Data.SqlDbType.VarChar).Value = ViewDetailsInformation.IPM_CASE_RECORD_NUMBER;
    rt = cq.LoadDataTable(ref tblTemp);
    if (rt != ReturnType.True)
    {
        [REDACTED]
        rt = ReturnType.Error;
        _Exception = GeneralSupport.SF.GEX(cq.Exception, "Error obtaining
data: " + cq.vDirectSPName);
    }
    return rt;
}

```

Apologies if my reading of the code is out.

Thanks.”

Lead developer: “sort of - the report controller first sets up the parameters via ViewDetailsInformation (e.g. dates and selected attorney brought about from the original page). At this point no actual data retrieval mechanisms are being triggered. This object with the populated ViewDetailsInformation class is sent to the view via the reports controller.

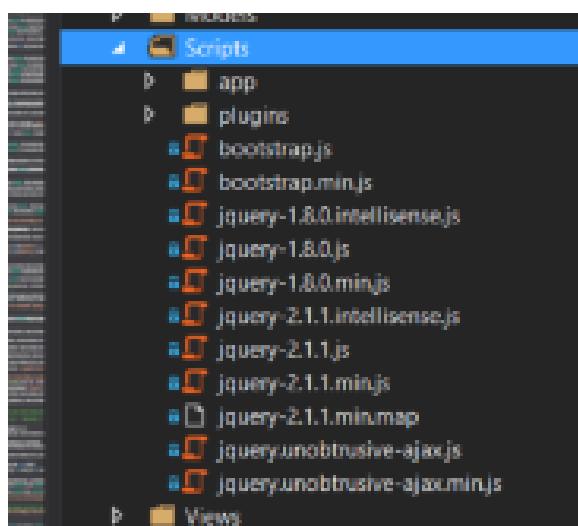
It then passes to the view that attempts to access the object, this object is normally a property of an object within the business logic file that checks if it is null and if null it then loads the data based on the parameters set within the ViewDetailsInformation property.

At this point the data retrieval mechanisms are triggered."

We have established that there is repetition in how the code has been wrote. This file would need to be Refactored further in future developments of the system.

It also provides us with an example of how they are scaffolded with the controller classes.

3.2 Scripts folder



Introduction:

"The Scripts folder stores the JavaScript files of the application.

By default Visual Web Developer fills this folder with standard MVC, Ajax, and jQuery files:"

Also,

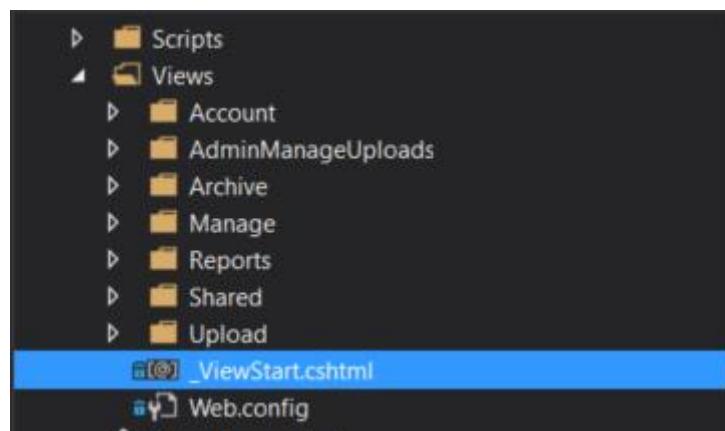
Note: The files named "modernizr" are JavaScript files used for supporting HTML5 and CSS3 features in the application."

Source: http://www.w3schools.com/aspnet/mvc_folders.asp

Developer comments:

No comments at this time.

3.3 Views folder



Introduction:

"The Views folder stores the HTML files related to the display of the application (the user interfaces). The Views folder contains one folder for each controller.

Visual Web Developer has created an Account folder, a Home folder, and a Shared folder (inside the Views folder).

The Account folder contains pages for registering and logging in to user accounts.

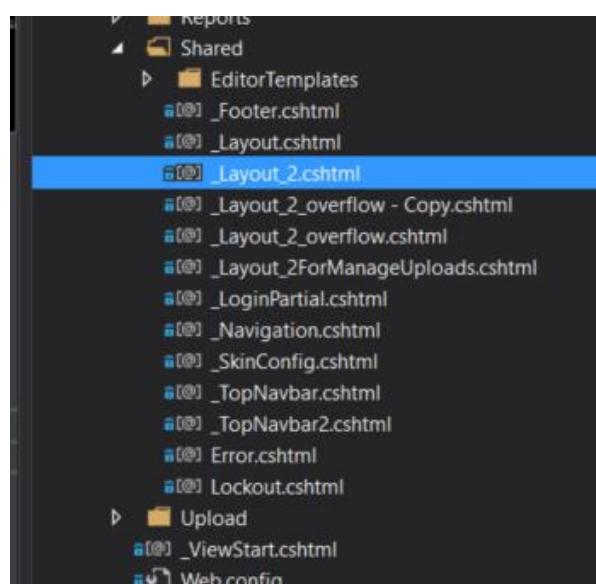
The Home folder is used for storing application pages like the home page and the about page.

The Shared folder is used to store views shared between controllers (master pages and layout pages)."

Developer comments:

No developer comments at present.

3.3.1 _ViewStart.cshtml & _Layout_2.cshtml



Introduction:

*"First and foremost, it is the **start** place of **executing pipelines** for views by RazorViewEngine as its name suggested. The _ViewStart.cshtml applied to all views in current folder and all subfolders. That means, whenever **RazorViewEngine** renders the view (inside of Views folder), it execute _ViewStart.cshtml first and then renders the view."*

Source: <https://sandippatilprogrammer.wordpress.com/2014/04/30/using-viewstart-in-asp-net-mvc-5/>

Also see above views folder image.

Developer comments:

Junior comments: *"The _ViewStart.cshtml file is the default layout for all Views folders and sub folders unless configured otherwise please see this excellent article on the various methods off how you can manipulate the Views and apply you're on view layouts."*

Source: <https://sandippatilprogrammer.wordpress.com/2014/04/30/in-details-viewstart-in-asp-net-mvc-5/>

Also;

Junior developer: *"With regards to _ViewStart.cshtml & _Layout_2.cshtml files I see you have changed the layout from generic, can you please elaborate on this?"*

I notice some Javascript classes at the bottom of _Layout_2.cshtml, are these to make the pages more dynamic with fading in etc. ?"

```
<script type="text/javascript">
$(document).ready(function () {
    jQuery('.post').addClass("hidemeX").viewportChecker({
        classToAdd: 'showmeX animated fadeInRight', // Class to add to the
elements when they are visible
        offset: 100
    });

    jQuery('.postL').addClass("hidemeX").viewportChecker({
        classToAdd: 'showmeX animated slideInLeft', // Class to add to the
elements when they are visible
        offset: 100
    });

    jQuery('.postR').addClass("hidemeX").viewportChecker({
        classToAdd: 'showmeX animated slideInRight', // Class to add to the
elements when they are visible
        offset: 100
    });

});

function goBackOne() {
    window.history.go(-1);
}
</script>
```

Lead developer: *"that bit of code {shown above} controls a few things - it helps "DIV" elements "fade in" as you scroll downwards if anyone of them have the hidemeX tag within their CSS class."*

The goBackOne() function is exactly that - some pages (most noticeably the ViewDetailsX pages) have a Go Back button, this then runs the javascript function to force the browser to go back one page.

These types of files eliminate (or reduces) the need to have the same code in every page."

3.4 Global.asax

Introduction:

This article describes the purpose of this file very well: "Have you ever felt the need of writing logic at the application level; precisely a location or a file where you could handle events or errors at the application level? Well if yes, then enter the Global.asax. Using this file, you can define event handlers with application-wide or session-wide scope."

Source: <http://www.dotnetcurry.com/aspnet/126/aspnet-global-asax-events-methods>

Also, Global.asax allows you to write code that runs in response to application level events, such as Application_BeginRequest, application_start, application_error, session_start, session_end etc.

"The Global.asax, also known as the ASP.NET application file, is located in the root directory of an ASP.NET application. This file contains code that is executed in response to application-level and session-level events raised by ASP.NET or by HTTP modules. You can also define 'objects' with application-wide or session-wide scope in the Global.asax file. These events and objects declared in the Global.asax are applied to all resources in that web application."

Note 1: The Global.asax is an optional file. Use it only when there is a need for it.

Note 2: If a user requests the Global.asax file, the request is rejected. External users cannot view the file.

The Global.asax file is parsed and dynamically compiled by ASP.NET. You can deploy this file as an assembly in the \bin directory of an ASP.NET application."

Source: <http://www.dotnetcurry.com/aspnet/126/aspnet-global-asax-events-methods>

Source Code file example:

```
using System;
using System.Collections.Generic;
using System.Data.Entity;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using System.Web.Optimization;
using System.Web.Routing;

namespace Yerra_MVC5
{
    public class MvcApplication : System.Web.HttpApplication
    {
        protected void Application_Start()
        {

            AreaRegistration.RegisterAllAreas();
            FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);
            RouteConfig.RegisterRoutes(RouteTable.Routes);
            BundleConfig.RegisterBundles(BundleTable.Bundles);
        }
    }
}
```

```
    }  
}
```

Developer comments:

No comments at this time.

3.5 Packages.config

Introduction:

Packages.config file is managed by NuGet to keep track of what packages and versions you have installed in the application.

Source: <http://stackoverflow.com/questions/27311813/mvc-packages-config-can-be-removed>

Developer comments:

Darin Dimitrov a developer on Stackoverflow explains the purpose of this file well: *"This file is managed by the NuGet infrastructure. It's used to track installed packages with their respective versions. If you installed the ASP.NET MVC 3 Tools Update it uses NuGet by default to track packages such as jQuery, EntityFramework, Modernizr. That's why you might be seeing this file when you create a new bare bone project."*

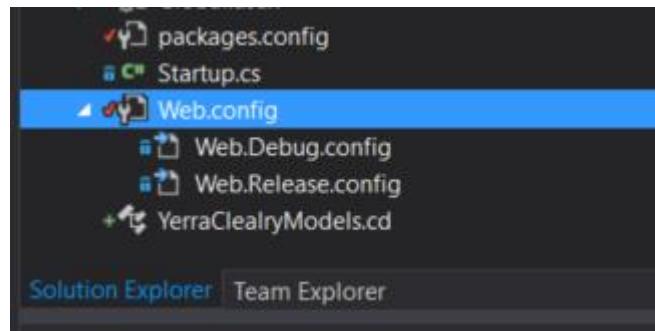
Source: <http://stackoverflow.com/questions/6304293/asp-net-mvc3-what-is-the-packages-config-for>

Some **example code** within this file:

```
<?xml version="1.0" encoding="utf-8"?>  
<packages>  
  <package id="Antlr" version="3.5.0.2" targetFramework="net451" />  
  <package id="Dependency" version="2.0.0" targetFramework="net451" />  
  <package id="EntityFramework" version="6.1.0" targetFramework="net451" />  
  <package id="GeoJSON.Net" version="0.1.38" targetFramework="net451" />  
  <package id="Glimpse" version="1.8.6" targetFramework="net451" />  
  <package id="Glimpse.Ado" version="1.7.3" targetFramework="net451" />  
  <package id="Glimpse.AspNet" version="1.8.0" targetFramework="net451" />  
  <package id="Glimpse.EF6" version="1.6.5" targetFramework="net451" />  
  <package id="Glimpse.Mvc5" version="1.5.3" targetFramework="net451" />  
  <package id="jQuery" version="1.8.0" targetFramework="net451" />  
  <package id="Microsoft.AspNet.Identity.Core" version="2.2.1"  
targetFramework="net451" />  
  <package id="Microsoft.AspNet.Identity.EntityFramework" version="2.2.1"  
targetFramework="net451" />  
  <package id="Microsoft.AspNet.Identity.Owin" version="2.2.1"  
targetFramework="net451" />  
  <package id="Microsoft.AspNet.Mvc" version="5.2.2" targetFramework="net451" />
```

3.6 Web.config

Introduction:



Web.config file contains application level configurations.

Just by Googling this file name we get a fine explanation: "**Web.config** is the main settings and configuration file for an ASP.NET web application. It is an XML document that resides in the root directory of the site or application and contains data about how the web application will act."

Source: https://www.google.co.uk/webhp?sourceid=chrome-instant&rlz=1C1MSNA_enGB681GB681&ion=1&espv=2&ie=UTF-8#q=Web.config

GoDaddy explain the need for this file very well on their website: "**WHAT IS A WEB.CONFIG FILE?** A web.config file lets you customize the way your site or a specific directory on your site behaves. For example, if you place a web.config file in your root directory, it will affect your entire site (www.coolexample.com). If you place it in a /content directory, it will only affect that directory (www.coolexample.com/content). However, in order for a web.config file to register in a specific directory, there must be a web.config file in the root directory.

web.config files work on our Windows servers.

Using a web.config file, you can control:

- Database connection strings.
- Error behavior.
- Security.

web.config files are XML documents. ".config" is not an extension like .html or .txt. For more information on how to set up web.config files, [visit Microsoft's Web site](#).

Source: <https://uk.godaddy.com/help/what-is-a-webconfig-file-5445>

3.6.1 What are the Web.Debug.config and Web.Release.Config files for?

Developer comments:

Developer named bluish comments on Stackoverflow:

"Are these files used to specify debug and release specific settings, so you don't clutter up the main web.config?"

It isn't limited to three files, you could (in theory) have as many files as you have environments. The "top level" Web.config provides a template of your web config. The files under it provide replacement values specific to that environment (like if you have different connection strings for local/stage/test/whatever).

Does it even make sense to place a connection string in the root web.config file if I have have a local and remote one in the debug and release web.configs respectively.

It would only make sense if it wasn't going to change between environments. Sounds like in your case it does so, in your case no, it would not make sense to leave it in the Web.config."

Source: <http://stackoverflow.com/questions/2791236/what-are-the-web-debug-config-and-web-release-config-files-for>

Also, please see this article for a further explanation:

<https://blogs.msdn.microsoft.com/webdev/2010/10/26/asp-net-web-projects-web-debug-config-web-release-config/>

3.7 DbContext Class

Introduction:

Please see the Yerra Clearly class diagram generated by Visual Studio and shown in the appendix of this document. This file is at the root of the ASP.net application.

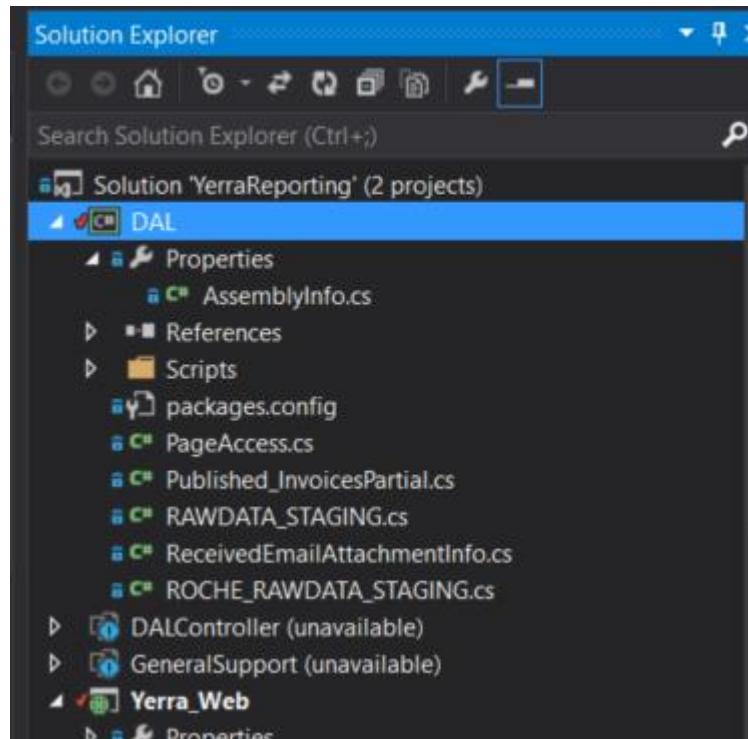
Developer comments:

Junior: "Can I ask is **DbContext** the base class for YerraClearly?"

Lead: "DbContext is used by some of the models, it is nothing I have created but rather forms an integral part of MVC. It's is part of the entity framework View -> Model -> Database. Handy for simple pieces of automation but struggles when you attempt to do anything complex."

Source: <https://msdn.microsoft.com/en-gb/data/jj729737.aspx>

3.8 Section 2 DAL (Data Access Layer)



Introduction:

DAL stands for data access layer...

Developer comments:

None at this point in time.

3.9 DAL Properties – AssemblyInfo.cs

Introduction:

Please refer to Section ??? under heading **Properties – AssemblyInfo.cs** and these details also apply here. The code example below shows the DAL version.

Code example:

```
using System.Reflection;
using System.Runtime.CompilerServices;
using System.Runtime.InteropServices;

// General Information about an assembly is controlled through the following
// set of attributes. Change these attribute values to modify the information
// associated with an assembly.
[assembly: AssemblyTitle("DAL")]
[assembly: AssemblyDescription("")]
[assembly: AssemblyConfiguration("")]
[assembly: AssemblyCompany("")]
[assembly: AssemblyProduct("DAL")]
[assembly: AssemblyCopyright("Copyright © 2015")]
[assembly: AssemblyTrademark("")]
[assembly: AssemblyCulture("")]
```

```
// Setting ComVisible to false makes the types in this assembly not visible
// to COM components. If you need to access a type in this assembly from
// COM, set the ComVisible attribute to true on that type.
[assembly: ComVisible(false)]

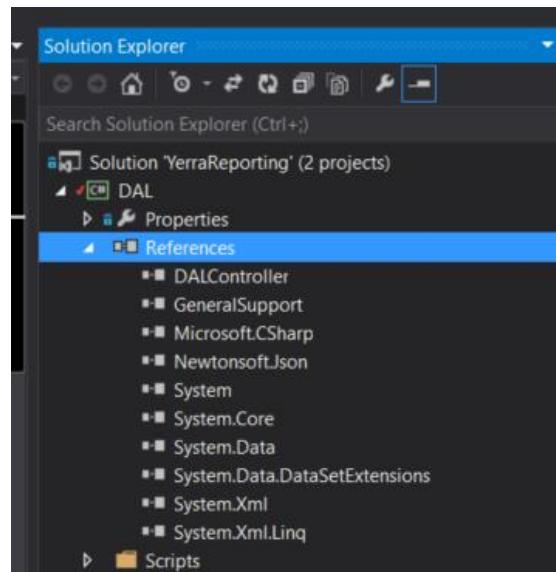
// The following GUID is for the ID of the typelib if this project is exposed to COM
[assembly: Guid("f0bf3a32-d3fb-404b-9a1b-19f53b932f50")]

// Version information for an assembly consists of the following four values:
// Major Version
// Minor Version
// Build Number
// Revision
//
// You can specify all the values or you can default the Build and Revision Numbers
// by using the '*' as shown below:
// [assembly: AssemblyVersion("1.0.*")]
[assembly: AssemblyVersion("1.0.0.0")]
[assembly: AssemblyFileVersion("1.0.0.0")]
```

4.0 DAL References

Introduction:

Please refer to Section 1??? under heading References and these details also apply here. The image below shows the DAL version and this is generic.

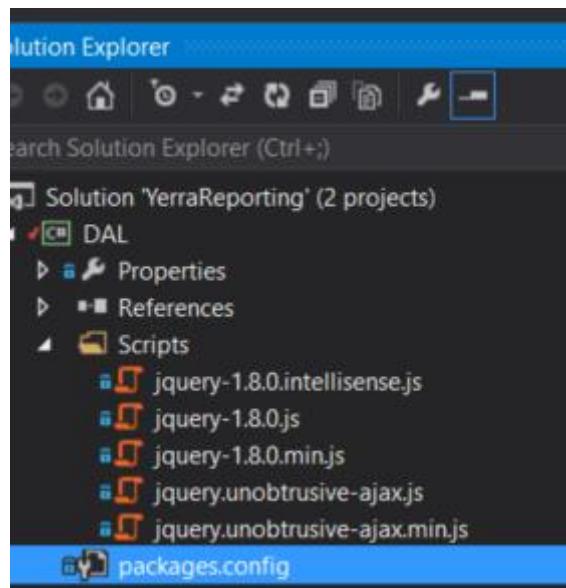


4.1 DAL Scripts folder

Introduction:

These are generic script files that make the Yerra Clearly app pages more dynamic.

Please refer to Section 1??? under heading Scripts folder and these details also apply here. The image below shows the DAL version and this is generic.



4.2 DAL Packages.config

Introduction:

Please refer to Section 1??? under heading **Packages.config** and these details also apply here. The code example below shows the DAL version. Also, see image directly above.

The code below is evident that the code has been modified/stripped back for the DAL version.

Source: <http://stackoverflow.com/questions/27311813/mvc-packages-config-can-be-removed>

Developer comments:

No comments at this point in time.

Code example:

```
<?xml version="1.0" encoding="utf-8"?>
<packages>
  <package id="jQuery" version="1.8.0" targetFramework="net451" />
  <package id="Microsoft.jQuery.Unobtrusive.Ajax" version="3.2.3"
targetFramework="net451" />
</packages>
```

4.3 DAL - PageAccess.cs

Introduction:

Two of the namespaces in this file link to the DALController and GeneralSupport references as in the Yerra_Web project.

```
using GeneralSupport;
using DALController;
```

As with most of the other Classes built in this application most of them have over 500 lines of code – Refactoring may be considered, especially for the larger class files, as a future development.

This file contains

Developer comments:

We had problems building and running the Yerra Clearly app on the junior developer laptop this is the conversation that unfolded:

Lead developer: "*I've added both GlobalSupport and DALController to the repository, please try syncing and re-referencing your project to these projects. If you need a hand just give me a shout and I'll call tomorrow.*"

Junior developer: "*I have tried to fetch the new Github repos and merge them into my local Invayis repos folder, however, I am sorry to say this hasn't worked.*

After cloning both DAL and General Support from Github into VS I have tried to simply copy and paste the files from both my local Repos into my local Invayis repo (in Windows File Explorer), however this also hasn't worked.

I am sure this is high level but please can we go over this together tomorrow and try and get it up & running. Please can we arrange a time for a call?

On the positive side it has given me a deeper understanding on how the sync, pull/fetch and push features work on Github – I am still not expert but it has helped "

Lead developer: "*no problem - seems you are making some progress though!*

If you expand the Yerra_web solution you will see something called references (try this on your local copy first). If you click add reference, you can then navigate to the bin folder of each of the two projects (DALController and General Support - however you may need to open each project and compile first) - find the DLL in the debug or release directory and click ok. This will then setup the reference. Normally I would add a project but to simply I believe I will remove the DALCOntrller and GeneralSupport projects from the yerra web solution and simply reference the dll's - this will make it less complex to share code and get it working first time."

Junior developer: "*I have cloned the DAL and General Support folders from Github but there doesn't seem to be any .dll files in the folders. I have browsed for the .dll file and not there both in VS and outside it through file explorer.*

When I try to run the DALController project file through the VS debugger it throws up error message see attached.

Let me know your thoughts we could maybe speak over phone tomorrow if suitable time?"

Lead developer: "*you would need to build the project (you dont actually run it as its a class library).*

If you right click project and select "Build" it will then create a DLL for you.

This is the DLL you would then link to when adding the reference to the web application.

*However, you *may* need to reference the General Support project first as that is the base for all other projects including the DAL Controller. Again you would load up project, build it and that will create a DLL. You then link DALController to General Support via the references option. This will then build. You would then reference both DALController and GeneralSupport in the Yerra Web project. I suspect you may also need to reference these two projects within the DAL project that forms part of the yerra web solution.*

DAL = Data Access Layer.

I can call no problem for other things but I'm sure you will crack it before that conversation!"

This resolved the problem.

Awaiting further feedback from Lead developer.

Code example:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Data.SqlClient;
using System.Data;
using GeneralSupport;
using DALController;

/* Do not change namespace so it can be used between projects - ideally when creating
a Data Access Layer for a project name it DAL */
namespace DAL
{
    public class PageAccess : GeneralSupport.myBaseObject, DALController.ISQLObject
    {
        #region "protected Variables"
        protected SQLProperties _Properties = new SQLProperties();
        protected string _strConnectionKey = "default";
        protected DateTime _dtBaseDate = GeneralSupport.SF.GetBaseDate();
        protected string _strTableName = "dbo.PageAccess";
        protected SqlConnection _sqlConn = null;
        protected bool _blTransactionEnabled;
        protected System.Data.SqlClient.SqlTransaction _sqlTransaction;
        #endregion

        #region Constructors
        public PageAccess(bool blTransactionProtected = false)
        {
            _blTransactionEnabled = blTransactionProtected;
        }
        #endregion

        #region "Properties"
```

4.4 DAL Published_InvoicesPartial.cs

Introduction:

Two of the namespaces in this file link to the DALController and GeneralSupport references as in the Yerra_Web project.

```
using GeneralSupport;
using DALController;
```

As with most of the other Classes built in this application most of them have over 500 lines of code – Refactoring may be considered, especially for the larger class files, as a future development.

This file contains

Developer comments:

Awaiting developer comments.

4.5 DAL - RAWDATA_STAGING.cs

Introduction:

Developer comments:

Awaiting developer comments.

4.6 DAL – ReceivedEmailAttachmentInfo.cs

Introduction:

Developer comments:

Awaiting developer comments.

4.7 DAL - ROCHE_RAWDATA_STAGING.cs

Introduction:

Developer comments:

Awaiting developer comments.

4.8 eMailProcessor

Need some info on this.

4.9 Currency Convertor

Developer comments:

Lead developer: "received invoice log now fixed, issue was down to the development site not having the latest currency conversions so it was unable to compute an invoice to CHF as it did not have any currency information later than 22-11-2015. The currency scraper only populates live."

Junior: "Just going through your emails (so this is an older one) and have some more questions to help my understanding, see in Red below."

Lead developer: "sure will do - the Global Currency table is a more recent addition but to give you a general logic/process flow background before we delve into the code here it is:

A scheduled task (created in C#) connects to a Web Service hosted by Open Exchange Rates.COM and requests the latest set of currencies by looking at each invoice date in the system and then comparing against all known currency conversions in ExchangeRate_Global_LU. **Does it base the currency rate on an average for that day or what it is at 'close of play' that day?**

It then sends a JSON request to the web service asking for all currencies based on the missing invoice dates - in other words it asks for a *historic* currency based on what invoice dates it is missing. However, for the most part the vast majority of new invoices will be the current date. **How are the missing invoice dates determined?**

However, if the date is the current date it continuously polls the web service until midnight as the currency can rapidly change throughout the day.

In terms of currency conversion during report generation it checks for three things:

- 1) Matches the line item's "INVOICE_DATE" against the Currency_Date within the ExchangeRate_Global_LU
* AND *
- 2) Checks that the BASE CURRENCY (to convert to) is set to 'CHF' (Swiss Franc) *NOTE* to support multiple customers in the yerra solution we will need to make this more dynamic if they don't use CHF because at the moment CHF is hardcoded into all of the queries
* AND *
- 3) Matches the line item's CURRENCY_CODE against a corresponding currency code within ExchangeRate_Global_LU

If all three match it then has a number either greater or less than 1 to work out the multiplier (in reality the divisor) for that line item's currency.

if it doesn't find a match the entire line item is omitted and is not part of the report - this can happen for a few reasons

- 1) *The Invoice imported into the system is a new invoice that does not yet have a currency for that particular invoice date; this will be resolved within 1 to 15 minutes automatically*
- 2) *The CURRENCY_CODE within the line item is invalid or some different standard to ISO CURRENCY CODES (meaning it is still invalid).*

Does that explanation give an idea of what you were looking for?"

Lead developer response to above in Red: "**A1** - it takes a cut every 15 minutes uses the latest grabbed currency for that day as the official currency for the current day (realtime). At midnight the latest currency before exchange close down (can't remember when) is the official currency for that date. For previous historic dates we can't do any more granular than daily as the INVOICE DATE (ledes) is limited to date only and 2) you can't get time based currencies (only date) for historic dates.

A2 - It performs a comparison between invoice dates in Ledes and what it has in the global currency table and then polls the web service for the missing invoice dates. Technically we could grab every historic date possible but you would need a bigger server and 2) it would slow down report querying if it needed to join ledes records within a billion plus currency conversions.

Hope that helps,"

5.0 Github

Account holds Yerra Clearly code base, Currency Convertor, YerraEmailProcessor, General Support and DALController.

Source: <https://github.com/>

Developer comments:

Lead developer: *"By the way, the Currency Conversion application is not part of the yerra web application - this is a separate console based application that connects to a web service to extract currencies from a paid API. This can be found on our github repo, in terms of projects it is much simpler than the yerra web site as it has a specific purpose (that is, connect to a web site) so this could be an easy thing to get your teeth into if you come to a point you are bewildered by the Yerra Web application and need a change of scenery."*

Also,

"The {Github} monthly charge isn't too expensive (around 38£ I think), but it will be critical to support this project as the repositories must be private otherwise some files with sensitive information (web.config) will be exposed to the internet.

I can't stress this enough, some people make a repo public, publish changes and then make private once they realise it is public. There have been hundreds of well publicised incidents in the media where a repo was public for less than 10 seconds and the entire repo was scraped. Apparently there are a lot of criminal gangs who constantly scan github repos using automated software in order to find insecure sensitive files."

Junior developer: *"Another question my sync'd version of Yerra Clearly seems to Pending edit Red ticks next to Yerra_Web and DAL. I haven't altered anything so any idea why these might appear?*

{Lead developer} I suspect when you were changing the references to the DLLs this would have modified the project definition files hence the red tick. "

Junior developer: *"Also, if Yerra Clearly has been modified or changed by you will these changes automatically sync through github?*

{Lead developer}: No I need to manually push the files, though I've held off recently as it may pull changes down by yourself. I've included a screenshot from github where it seems to be indicating some files have been updated.

This is not a problem as conflict resolution is fully supported but I've held off due to at the time the short availability I previously had on the project."

- After you switch contexts you'll see an organization-focused dashboard that lists organization repositories and activities.

8 hours ago	RoadHog31 pushed to master at Incoria/Invayis	 master is now f6bae01
on Feb 3	RoadHog31 pushed to master at Incoria/Invayis	  76a0127 accountdb
on Feb 3	RoadHog31 pushed to master at Incoria/Invayis	  03a48a6 accountdb
on Feb 3	RoadHog31 pushed to master at Incoria/Invayis	  024d44a Addnewtonsoft.json reference
on Feb 3	RoadHog31 pushed to master at Incoria/Invayis	  master is now 3091aa9
on Feb 2	RoadHog31 pushed to master at Incoria/Invayis	  3091aa9 Added both Generalsupportand Dalcontroller refs
on Feb 2	RoadHog31 pushed to master at Incoria/Invayis	  56b2f98 DALand general support ref added
on Feb 1	RoadHog31 pushed to master at Incoria/Invayis	  master is now 0a092be
on Feb 1	RoadHog31 pushed to master at Incoria/Invayis	  master is now 0a092be

Note: These changes were not anything significant.

5.1 Data Quality {DQ} issues

Introduction:

There have been some significant issues with the quality of data received from clients and how the reporting tool handles this. Below is an outline of this and the plan to resolve these issues.

Developer comments:

Junior developer questions with Lead developer responses in Red as follows:

Junior developer: *"I am bringing myself up to speed with the IPM & MatterId Data Quality Issues we have been having over the past few months – please can you point me to examples of the code and databases etc. that handle this, specifically the logic?"*

Is IPM/MatterID data dealt within the code of the Executive summary YTD, ViewAllTimekeepers/2016 etc. ?"

Lead developer: The issues are *"Due to the low quality of the data in terms of Client Matter ID you are getting around a 50% hit. The decision to change logic to match on IPM was made a while ago and I believe the consensus was that client matter ids would rapidly improve 2016 onwards."*

Yerra Clearly Data flow:

1. Latest IPM Case Record Number is uploaded to SQL Server database and 3nf Table created – can I have name of this please?

IPM_STAGING -> Published_IPM

2. Then IPM data is cross referenced with the **Client Matter ID** on the raw invoice data that is uploaded and validated into Yerra Clearly.

No it isn't, only the logic that generates the reports maps the client matter id.

3. If no data match then the invoice is ignored – **and goes where? Held on physical file on server?**

No it is ignored by the report, as it can't match an IPM with a Client Matter ID so it is not calculated, hence the "drop" in CHF totals once we moved from raw published invoices to only using invoices mapped with IPM.

4. If match ok displayed to front end view. Reports default to "Zero" Why?

DQ Resolution:

The plan is to:

1. **Make a Vendor Management page** - "A number of filters allow users to narrow by vendor or date to make the list more manageable."
2. Introduce **"dual logic"** – 1. for IPM and 1. for Raw invoices. **Where is this in the source code?**

Most logic is in the SQL side, very little processing logic is performed within the application itself.

3. Uploading latest IPM data. Different column positions fine as long as headings the same.

4. ***Vendor and data cleanse page*** to resolve back end issues – for example some due to Genentech US firms data who use different fields for timekeeper names. So you are creating a ***Vendor 'look up' table***.

Yes. Though with more fields to support advanced matching/analysis.

5. Our business logic trims country code off matterID.

Only for some calculations - for example you would need it when you are trying to work out totals per country.

6. ***OrphanedClientMatterId*** UI page – implemented as temporary fix but can be left in permanently if required. [Senior Analyst confirmed this is a permanent feature].

Does this outline look correct, please feel free to comment/correct?"

To follow on from the above remarks, we have further dialogue around DQ as follows:

Lead developer – “it was stated a while back that Client Matter IDs would improve hence the decision to use IPM as the base table and match with the client matter id. However, within the first quarter (FY Swiss) there seems to be little evidence to support that consensus.

In terms of the code itself - all logic relating to IPM can be found within the Stored Procedures prefixed with IPM_ (xxxx).

*Also, not all code makes use of IPM / Invoices - in fact there has been requests to drop using IPM in some instances which most likely has led to re-conciliation issues between some similar reports where the totals do not match. Once the Vendor/TimeKeeper software is in place and all Client Matters ID match I would advise *all* reports are moved over to match with IPM. Flipside of this is that the matching is only as good as the IPM file so there will be totals under calculated until the latest IPM file is imported.*

I've tried to answer best I can in red for your other points."

Also, Lead developer comments from previous emails: “There are two sets of logic - one for IPM and one for raw invoices. IPM will show less due to the need to match CLIENT MATTER ID with Case Record Number whereas non IPM Site Spend will be based on raw invoices.

In terms of the logic changing for IPM nothing has changed? The only thing that did change was the latest IPM submission that is sent by XXXX - therefore if there is an issue with that file it would impact the IPM (only) results?”

Also,

“01/01/1754 means that the date cannot be found in the database so if you are looking for an IPM and finding a date that does not sound right - would you be able to give me an example one thanks.”

And,

“It is because the Client Matter IDs are of a such a bad quality they do not match their respective IPM counterparts. Due to linkage with the IPM, Invoice rows are only calculated if there is an IPM match.”

As well as, “*TotalSpend, TotalSpendYTD, Fee/Expense Breakdown, Hourly/Rate Fixed Fee, Total Invoices and TimeKeeper list do not rely on an IPM match so the Open Matters/Active Timekeepers may be a little low in comparison.*”

5.2 Section 3 Database

Introduction:

The client LEDES files and IPM linkage data is stored in Microsoft SQL Server 2012 database tables.

Lead developer: "*I tend to create everything directly in SQL Enterprise manager but to be honest it should really be done via Visual Studio as you can then manage it as a version controlled project. I create directly within SQL Enterprise manager for expediency as opposed to having to deploy the entire project every time like we do with the code.*"

Developer comments:

Hopefully, the following dialogue will provide a flavour of the types of issues and technicalities behind the Yerra Clearly app database.

Junior developer: "*Obviously I have access to Development_Yerra_InvoiceReporting – just to be clear is this the development data you gather before putting to live?*

Also, once brought ‘in house’ what other tables will I have access to? Are there live tables that are identical to Development_Yerra_InvoiceReporting?”

Lead developer: "*The other main database is Yerra_Live_InvoiceReporting, the Dev database is where the initial changes are made before copying to live.*

Hmm what else, there is a third database we use for demo purposes but currently not being used mainly because the demo site was called invayisdemo.com.”

5.2.1 Database Utilisation

Lead developer: "*In terms SQL tracing I quite often use SQL Profiler so you can work out execution plans/see where things are going wrong (for example slow query performance etc) though that deserves a book on its own.”*

Lead developer: "*Apologies for the delay in getting back to you – I’m using my work account as its easier to embed pictures compared to Hotmail. I have thrown in two diagrams to give a high level view of the datasets being used and the hot topic at the moment which is the email processor.*

In terms of the “web site”, you don’t need your own version as you can run it from your own desktop once the code compiles. In terms of the errors, are you able to provide a screenshot of the errors only or copy and paste? If you hit the “warnings” filter in the error list it should remove all of the warning so we can concentrate on the errors only. I’m using it on one desktop and 3 laptops without any issues nor did I need to make any changes to the builds to get the code working. I suspect it is probably something as simple as having to install a particular .NET framework on your PC.

5.2.2 Database design

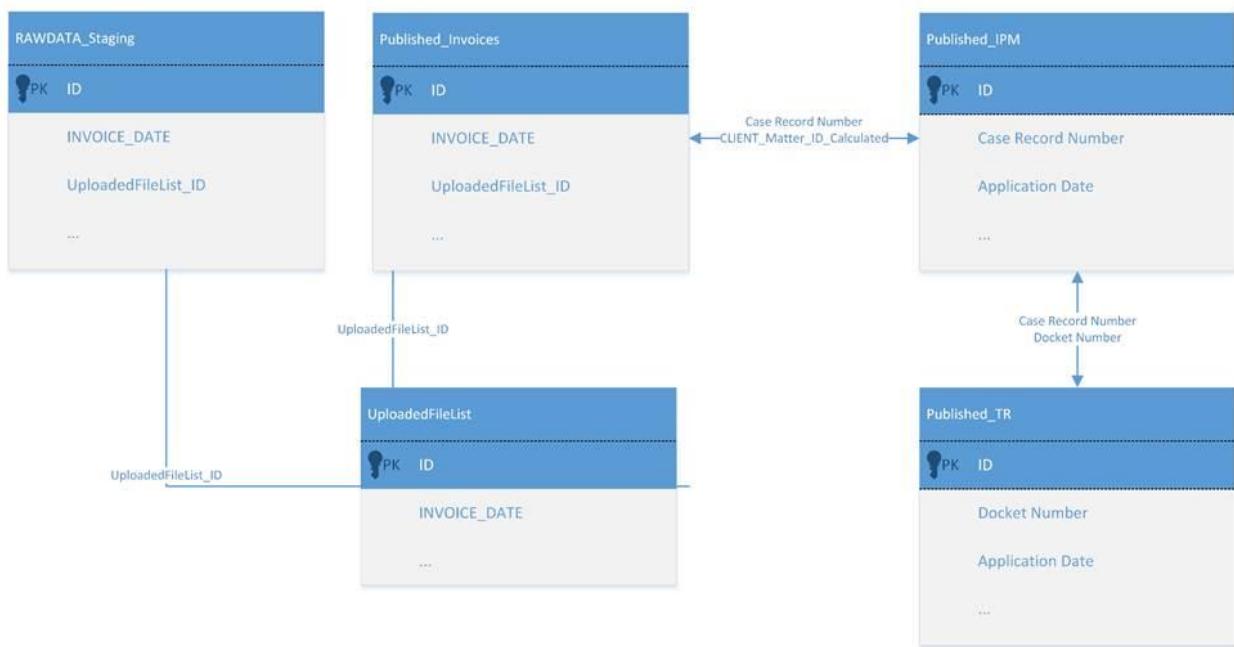
In terms of syncing with YSDASH & Yerra Clearly – you don’t have to do this. Your copy of the code is the “master” copy, github ensures version control so we can revert back if we need to. YSDASH and INVAYIS only contains a copy of the “compiled code” (think of it as an application executable not the

original source code) and once we are happy that the code is fit for purpose, we “publish” the compiled code. We then take a copy of this compiled code and copy it over the YSDASH or INVAYIS website directories overwriting the original compiled code and that is how the website is updated. You don’t have access to do this yet but we can add that as we go along.

It is possible to setup another test copy for the repos on github but you run the risk of using the wrong one for changes that could either lose time or worse sync changes from your test copy to the master copy – I think you would be better off just making another copy on your local copy of the code (if you right click on the solution there is an option to “open in windows explorer” which shows you exactly where the local copy is kept) – removing the github syncing part and then just play about with the local copy copy of the code(!)

In terms of the dataset structure – I have included a very high level view, for more low level/technical just ask and we’ll go through it but this should give you a general understanding of how the major datasets link together:

Key Fact Tables

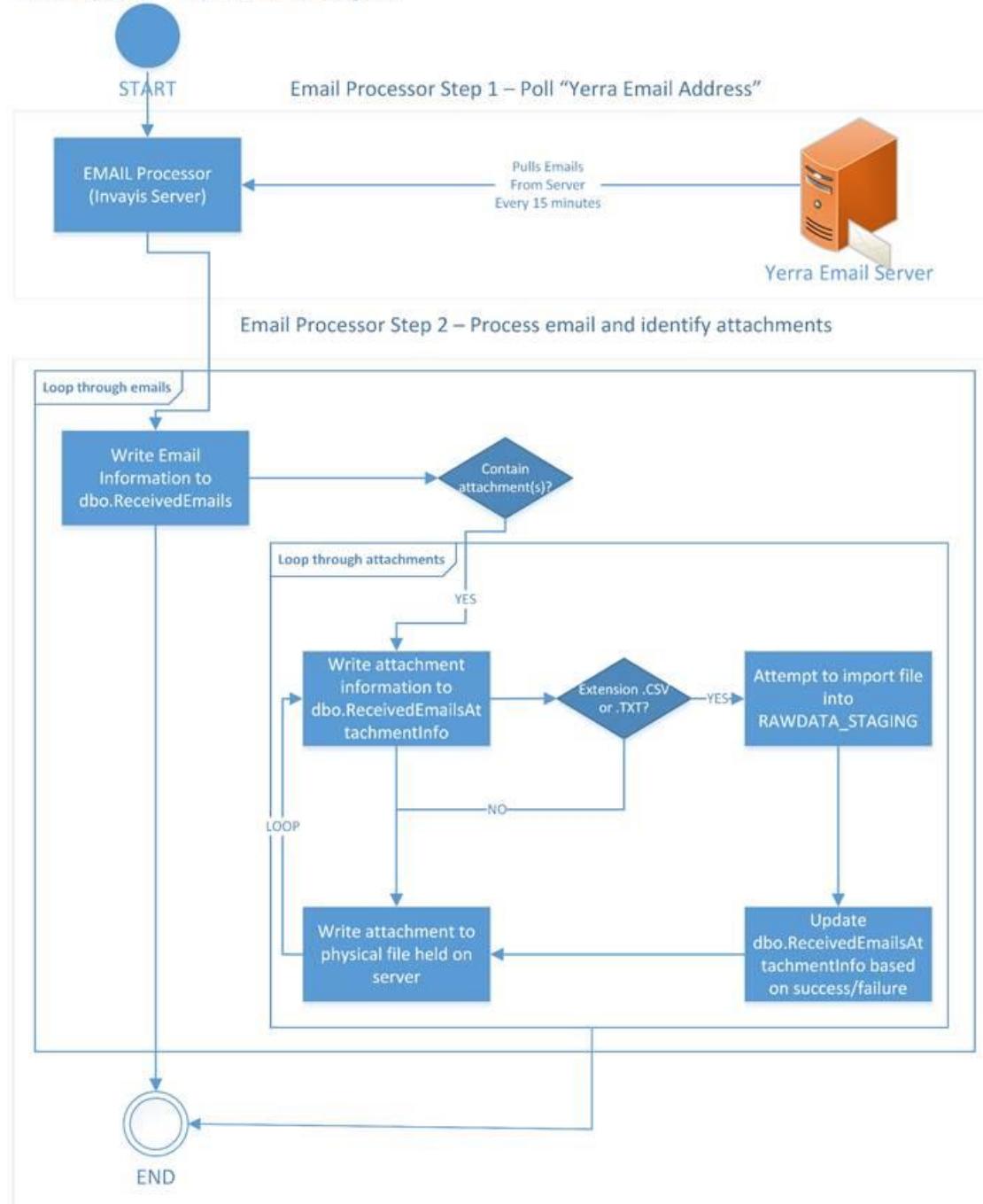


RAWDATA_STAGING is populated by both the analysts when they upload files and by the email processor. Each line item is grouped together with a particular ID which originates from the UploadedFileList table, this also controls and maintains the status of the validation of such line items and the publishing. As Published_Invoices also contains a “grouping id” from UploadedFileList it allows you to revoke a published set of records.

CLIENT_MATTER_ID, Case Record Number and Docket Number are the main links to join together the tables though the CLIENT MATTER ID tends to be free text (CLIENT MATTER ID belongs to Published_Invoices) so there are a lot of DQ issues with this which the analysts are trying to resolve. There is some heavy fuzzy calculations done on this field to get it right but some are that bad the analysts need to do a manual intervention.

Next up – we have a basic application flow on how the Email Processor works – this is completely exclusive to the web site and runs as a scheduled task as opposed to a service to easier fine tune when running. Any questions please give me a shout.”

Yerra “System” – Top Level Flow Diagram



Senior analyst: “Another small query is whether the site can be changed to a different web address? We have rebranded to Yerra Clearly therefore we will need the site to be transferred to this domain (which I think we now have) – can this be done and how difficult is it?”

Lead developer: "We can order a new web address within around 10 minutes unless you already have control, the longest part will be the Enterprise Level Security Certificate (SSL/HTTPS) where Yerra need to prove that they are indeed the legitimate company. However, this type of certificate is normally only used by banks and other large financial organisations so you could purchase a standard SSL and have it active in around 1-3 hours."

In total to migrate the site including purchasing and verification of the SSL certificate including purchasing and provision of a new web site name, you are looking at around 1-3 hours."

Also,

Senior analyst: "We have been given an urgent deadline by our client to get all figures working on the live site by February 10th (however the sooner the better). If you think the "two-tiered" logic will work here (i.e. if the system cannot find a match for an invoice in IPM then it should refer to the information in the LEDES) can this be put into place on live ASAP? We need all figures on the Executive Summary YTD page to add up correctly by this deadline (or sooner)."

Lead developer: "I think the easiest option is to migrate the dev site to live but disable the email options (well actually there isn't anything you can do with it) which we could do this weekend as it takes around 2-4 hours for database migration? We can plan for Sunday?"

Senior analyst: "Have you looked into the reason why invoices do not seem to appear on the Received Invoice Log page once they have been published?"

Lead developer: "Yes, it was due to a link between the uploaded files and the published records not always being the same between the two, this is now fixed but not currently visible as it also needs a small change to the report in terms of the date ordering. I'll have it in place for you by tonight."

Senior analyst: "An analyst and I have reviewed the current email module today and so far it looks good however we would like some changes made/some things to be added if possible. I have attached an Excel file which as a layout of how we would like the pages to look. There is a separate tab for each current page with notes at the top followed by a mock-up of the tables. All bold italic text are columns we would like to be added to that page and all plain italic text below that is an explanation of what we would like. Please let me know any questions you have around this. A few of the things we have asked for may not currently be possible however we are looking to change our process so all emails coming into the inbox moving forward should have only .txt files so there should be no invoices unable to be imported and we are also requesting that each invoice is a separate file rather than batched into one file – hopefully this should make some of our requests easier."

Lead developer: "Apologies I have already started on the work, I didn't realise you were only asking for timescales. It will be ongoing work but lets aim for the weekend or end of this working week. One issue with showing line items / published invoices on the EmailHistory page is that there is no real link between attachments and what is published when you mix both file types (e.g. a lot of stuff are PDF files or embedded images that form part of the email). I'm currently scratching my head trying to

work out how this can display both. However, as always everything is possible it just needs a little thinking.

*I will also be sending a diagram of the email processor to Stephen later today that details how the process works from a very high level, not sure if it is of any interest to you but will copy you in *****.”*

Lead developer also states the following: “There are two sets of logic - one for IPM and one for raw invoices. IPM will show less due to the need to match CLIENT MATTER ID with Case Record Number whereas non IPM Site Spend will be based on raw invoices.

No problem, as I mentioned earlier it is because the spreadsheet I gave you was based on raw invoices not linked to IPM but the YTD Spend to Date is based upon linking the client matter id (albeit with the country code eliminated to maximise matching) to the IPM master care record id. Due to the low quality of the data in terms of Client Matter ID you are getting around a 50% hit. The decision to change logic to match on IPM was made a while ago and I believe the consensus was that client matter ids would rapidly prove 2016 onwards.

I believe we can definitely hit by the end of the week and the received invoices log also includes manual uploads. The log is live to the point where on a new day they may not show until the first set of currencies are pulled in for that day. This is every 15 minutes and FX trading usually begins around 7am GMT.”

5.3 Cross Charging Report

Lead developer: “please find attached the Cross Charging Report - the steps taken were:

- Imported Master Patent Attorney List Excel File (named Patent Attorneys) into the database
- Performed a comparison and selected items where

Invoices CLIENT_CITY <> IPM Responsible Site

OR

IPM Responsible Site <> [Attorney List For Cross Charging].[Patent Attorney]

Couple of caveats

For this to work there must be a matching Client Matter ID between the invoices and the IPM file. I noticed there were around 4-500 missing records once I started joining the IPM file to the Master Patent Attorney List meaning there was some invalid Responsible Attorneys within the IPM file. With this in mind I have attached a second file where there are no matching Responsible Attorneys in IPM - there are over 72000 but I assume these will be mainly dead/abandoned cases.

For junior developer: this was a fairly simple query - the code is below (the cross charge match) so please take a look see what you think:

/** CROSS CHARGE **/

SELECT

 Published_IPM.[RESPONSIBLE ATTORNEY]
 ,Published_Invoices.CLIENT_CITY
 ,Published_IPM.[RESPONSIBLE SITE]
 ,[Attorney List For Cross Charging].[Site] as [Site From Patent Attorney Master List]
 ,Published_Invoices.*

FROM

Published_Invoices

 INNER JOIN Published_IPM ON

 Published_Invoices.CLIENT_MATTER_ID = Published_IPM.[CASE RECORD NUMBER]

 INNER JOIN [Attorney List For Cross Charging] ON

 Published_IPM.[RESPONSIBLE ATTORNEY] = [Attorney List For Cross Charging].[Patent Attorney]

WHERE

 Published_Invoices.CLIENT_CITY <> Published_IPM.[RESPONSIBLE SITE]

 OR

 Published_IPM.[RESPONSIBLE SITE] <> [Attorney List For Cross Charging].[Site]

/** CHECK FOR ATTORNEYS WITH NO MATCH **/

SELECT

 [Attorney List For Cross Charging].[Patent Attorney] as [Patent Attorney From Patent Attorney Master List]

 ,[RESPONSIBLE ATTORNEY]

 ,Published_IPM.*

FROM

 Published_IPM

LEFT JOIN

 [dbo].[Attorney List For Cross Charging] ON

 Published_IPM.[RESPONSIBLE ATTORNEY] = [Attorney List For Cross Charging].[Patent Attorney]

WHERE

 [Attorney List For Cross Charging].[Patent Attorney] is null

5.4 Report Spend for each Law firm

Lead developer: *"This is based on all current invoices and not linked to IPM to maximise all invoices.*

If you have any questions please do not hesitate to get in touch.

here is the code to do that:

```

SELECT
    LAW_FIRM_NAME as Vendor_Name
    ,SUM(LINE_ITEM_TOTAL / ExchangeRate_Global_LU.Conversion_Multiplier) as
Absolute_Spend_CHF
    , SUM(LINE_ITEM_TOTAL) as Absolute_Spend_Local_Currency
    , Published_Invoices.INVOICE_CURRENCY

FROM
    Published_Invoices

    INNER JOIN ExchangeRate_Global_LU ON
        Published_Invoices.INVOICE_CURRENCY =
ExchangeRate_Global_LU.ISO_Currency_Code
        AND Published_Invoices.INVOICE_DATE =
ExchangeRate_Global_LU.Currency_Date
        AND ExchangeRate_Global_LU.Base_ISO_Currency_Code = 'CHF'

WHERE
    [Status] = 'C'
    GROUP BY Published_Invoices.LAW_FIRM_ID,
LAW_FIRM_NAME,Published_Invoices.INVOICE_CURRENCY
    ORDER BY Absolute_Spend_CHF DESC

```

5.5 Report list of all law firms with their total spend

Senior analyst: *"A request has just come through from the client which needs done ASAP, can you please send me a list of all law firms with their total spend (everything that is currently in the system – all 2015 and 2016 spend) converted to CHF please? So basically the top 10 vendor table but for all vendors and I need all spend rather than a single quarter. Can you do this urgently as we need to build a report on this today."*

Lead developer: *"no problem, here you go - there are around 135 null entries meaning that there is no record of these invoice numbers within the system. This could be due to rejected files or perhaps invalid invoice numbers that were subsequently changed.*

There may be a few duplicates due to the system trying perform currency conversions on NULL (empty) values or perhaps invoice_number and invoice_date belonging to more than one vendor.

Stephen - code is this:

(before you check = both Master Case Record Number Calculated and Master_Client_Matter_ID_Calculated are automatically created by the system for each line item once they are successfully published, this is created by something called a SQL trigger upon insert or update that attempts to strip the country code from the client matter id to work out global spanning client matter ids / case records for better matching between Invoices and IPM).

```

SELECT
    DISTINCT
        INVOICE_NUMBERS_TEMP.INVOICE_NUMBER as INVOICE_NUMBER_CHECK
        ,INVOICE_NUMBERS_TEMP.INVOICE_DATE as INVOICE_DATE_CHECK
        ,LAW_FIRM_NAME
        ,CLIENT_MATTER_ID
        ,ISNULL(Published_Invoices.INVOICE_TOTAL /
ExchangeRate_Global_LU.Conversion_Multiplier,0) as INVOICE_TOTAL_CHF
        ,INVOICE_CURRENCY
        ,Published_IPM.DIVISION as Client
FROM INVOICE_NUMBERS_TEMP
LEFT JOIN Published_Invoices ON
    Published_Invoices.INVOICE_NUMBER = INVOICE_NUMBERS_TEMP.INVOICE_NUMBER AND
    Published_Invoices.INVOICE_DATE = INVOICE_NUMBERS_TEMP.INVOICE_DATE
LEFT JOIN Published_IPM ON
    Published_IPM.[MASTER CASE RECORD NUMBER CALCULATED] =
    Published_Invoices.MASTER_CLIENT_MATTER_ID_CALCULATED
LEFT JOIN ExchangeRate_Global_LU ON
    Published_Invoices.INVOICE_CURRENCY =
    ExchangeRate_Global_LU.ISO_Currency_Code
        AND Published_Invoices.INVOICE_DATE =
    ExchangeRate_Global_LU.Currency_Date
        AND ExchangeRate_Global_LU.Base_ISO_Currency_Code = 'CHF'
WHERE
    Published_Invoices.[Status] = 'C' OR Published_Invoices.[Status] is null
    ORDER BY
        LAW_FIRM_NAME,
        INVOICE_NUMBERS_TEMP.INVOICE_DATE

```

5.6 Site YTD spend for 2015

Lead developer: "it is a field that is automatically generated when records are published into the system."

The reason for it came about due to data quality concerns over the LEDES data and that some client matter ids will have additional information (e.g. W12345-GB-1, W12345-GB-2) that has no bearing with the IPM data that it is often linked to. Also some Client Matter IDs span multiple countries whereas for some reports we need to maximise matching between roche data and IPM data but the countries are irrelevant.

So to mitigate the DQ issues for some reports we "trim" the Client Matter ID and the IPM Case Record Number limiting it to W12345 so we know for certain that a particular LEDES line item matches a respective IPM item from a generic point of view.

There is some further validation done with the IPM <> LEDES line item to ensure it is pointing to the right IPM record. The MASTER CLIENT MATTER ID (& MASTER CASE RECORD NUMBER) exists to point the logic in the "general direction" of the correct set of IPM records.

Analyst - still have contract but I've been offsite all week and will not be back in the office to sign/scan until Monday now (change of plan for today, I'm now out offsite today too).

Finally, I may have missed an email potentially as I'm awaiting feedback for the current status of the email module thoug happy to copy to live at any time if you are ok and we can take it from there.

Live site will be down for around 30 mins as its a minor update.

In the code you can review the AccountDB within the models folder, Account Views and the AccountController for the login screen. It also requires a connection within web.config that tells the application where to store the user details.

Note - creating a login system is fairly advanced MVC but there are many tutorials on it including within the online books available on safari.

I tend to create everything directly in SQL Enterprise manager but to be honest it should really be done via Visual Studio as you can then manage it as a version controlled project. I create directly within SQL Enterprise manager for expediency as opposed to having to deploy the entire project every time like we do with the code."

5.7 Yerra Clearly Editable Report (GUI) – new development

Lead developer: "It is because the Client Matter IDs are of a such a bad quality they do not match their respective IPM counterparts. Due to linkage with the IPM, Invoice rows are only calculated if there is an IPM match."

Senior analyst: "How can we resolve this in the quickest possible way i.e. cleanse and correct the matter IDs – is it something we need to do on our side and if so how? Alternatively can you direct Stephen on how to do this or is it something you have to be involved in? We need to get as much spend showing through the tool as possible."

Lead developer: *I would perhaps tentatively suggest that there is a editable report that lists all client matter ids that do not have a matching IPM (Case Record Number).*

Staff could then perhaps edit these direct? Raw record edits - I'm usually against this sort of thing as it means we become responsible for the inbound data flows but as the Client Matter ID are of such bad quality with no obvious process in place to enable vendors to get it right first time this is perhaps the only option we have?

This would take around 15 hours to do I suspect."

5.8 Title required

Lead developer: *"As currently the IPM site data does not work 100% of the time (for the spend by site page logic) we wondered if it would be possible to have a 2 step logic in this - if the system is unable to find a site match in IPM for a matter ID rather than not assigning this spend could it then revert back to the site given in the LEDES data so there is no spend not being assigned to a site? Let me know if this is possible or if it needs further clarification.*

This is possible, I can do something called a "union" between the two datasets so it looks like one dataset in the report where items will only appear if not in the other. Makes it slightly more complex but nothing that cannot be done.

Senior analyst: *"In your below email you say that the Executive Summary YTD figures are now fixed, if so can this be transferred to the live site ASAP please? I understand that certain parts of the email module may transfer over too but as long as none of those is visible to the client that is fine. Additionally in the spreadsheet you mention that the phase coding update will be completed when the email module is done - was this also simply a matter of transferring to live or does it need put in place? If it simply needs transferred to live can this be done when this Executive Summary YTD update is also transferred?"*

Lead developer: *I tried this last night but the manual upload module for analysts failed so had to revert, the reason for this is that email module has changed the structures of the uploading part slightly so the live site backend database structure needs to change which is a major update and not something that I would advise until everything is signed off. I spent a bit of time looking at the logic specifically for the executive summary part and it *may* be possible to migrate these over. Something I will wait until the weekend though to ensure that the live site isn't down during peak periods."*

5.9 Legal Spend Application – new development

Senior analyst: "We may also be looking into customising the site for **legal spend/another client** soon so could you give me a quick rundown of the best way of doing this? E.g. duplicating the site and changing it completely from there or simply changing the client facing side of the site for different logins?"

Lead developer: *It is possible to extend the site (my preference) to support multiple clients but this may be more confusing for the analysts as I guess they would need to ensure they "uploaded" data to the right client and if they got it wrong Roche may see data not intended for them (and vice versa). So it could be simpler to simply copy the site and upload data to that one?"*

6.0 False Flagging – Data does match IPM?

Note: System should only flag when data does not match IPM.

Senior analyst: "That's strange, please find a screenshot below of one of the matter IDs in the LEDES data, Analyst has checked them and they were all in the Matter ID column of the LEDES in the exact format I sent in my email. The file name for this ID is herzog 22.02.2016 and 23.02.2016 (22 invoices).txt. This file has not yet been published to the site however has been uploaded by Analyst."

Line Item Num Hrs	Line Item Avg Amount	Line Item Total	Task Code	Expense Code	Activity Code	Line Item Unit Cost	Time Keeper Name	Client Matter ID	PO Number	Matter Name	Invoice Tax Total	Invoice Currency	Client City	Line Total
0.25	0	223.13	PA630		T115	250	Ottmanns, Dr.	P32048-EP		IGF EPZ For stu	36.96	EUR	Mannheim	35
1	0	5.95		E124		5	Eick, Dr. Alex.	P32048-EP		IGF EPZ For stu	36.96	EUR	Mannheim	0.3
1	0	2.38		E124		2	Eick, Dr. Alex.	P32048-EP		IGF EPZ For stu	36.96	EUR	Mannheim	0.3
0.5	0	148.75	PAS20		T115	250	Ottmanns, Dr.	P31109-JP		Identifications	54.85	EUR	Mannheim	23
0.25	0	24.38	PA630		T152	250	Ottmanns, Dr.	P31109-JP		Identifications	54.85	EUR	Mannheim	1.5
1	0	2.38		E124		2	Sitner Nadja	P300971-CN		Identifications	13.79	EUR	Mannheim	0.3
1	0	5.95		E124		5	Eick, Dr. Alex.	P31109-JP		Identifications	54.85	EUR	Mannheim	0.3
2	0	5.19		E101		0.5	Eick, Dr. Alex.	P31109-JP		Identifications	54.85	EUR	Mannheim	0.3
5	0	21.98		E101		0.5	Eick, Dr. Alex.	P31109-JP		Identifications	54.85	EUR	Mannheim	0.4
1	0	2.38		E108		2	Eick, Dr. Alex.	P31109-JP		Identifications	54.85	EUR	Mannheim	0.3
7	0	2.38		E108		2	Eick, Dr. Alex.	P31109-JP		Identifications	54.85	EUR	Mannheim	0.3
9	0	5.95		E124		5	Eick, Dr. Alex.	P31109-JP		Identifications	54.85	EUR	Mannheim	0.5
1	0	209.75	PA630		T152	250	Feuerherdt, Dr.	P32048-EP		Comparable si	117.04	EUR	Mannheim	49
1.5	0	70.11		EY01		0.5	Eick, Dr. Alex.	P31109-JP		Identifications	54.85	EUR	Mannheim	1.1
0.5	0	148.75	PA630		T115	250	Feuerherdt, Dr.	P32048-EP		Comparable si	117.04	EUR	Mannheim	23
1	0	5.95		E124		5	Eick, Dr. Alex.	P31109-JP		Identifications	54.85	EUR	Mannheim	0.5
1	0	2.38		EY08		2	Sitner Nadja	P32457-JP		Comparable si	117.04	EUR	Mannheim	0.3
		14.38		EY08		3	Dick, Dr. Alex.	P31109-JP		Identifications	54.85	EUR	Mannheim	0.3

Lead Analyst: "Further to the below we have found some matter IDs that are entered in the validation screen in the correct format and are in the IPM data however they are still highlighting amber in validation which suggests there is still no connection. Some of the matter IDs are below however it has happened for quite a few. Could you please let me know why this could be?"

P31779-WO

P30834-US

P31889-WO

P26438-US-1

P26622-US

P27378-US

P27457-JP

P31109-JP

P23186-CA

Lead Developer: "It is because the Client Matter IDs are of a such a bad quality they do not match their respective IPM counterparts. Due to linkage with the IPM, Invoice rows are only calculated if there is an IPM match."

6.1 User Manager Application – new development

Lead developer: "thanks for your message, no I mean we are lacking a user manager. The ability to create/delete/edit users / reset passwords etc.

This was never originally scoped into the requirements so all user management is performed in the back end by directly editing data tables requiring developer intervention. It would be more preferential for admins to manage users via some form of interface. I think this would be an ideal project for you to work on as you could technically create a new application (MVC) that once ready could be brought into the wider yerra clearly website or keep standalone.

The code below exists but it is not hooked up to any major view and it is also not hooked up to an email server to send emails (e.g. reset links etc) so it would be an end to end piece of work."

6.2 Truncating data into SQL table

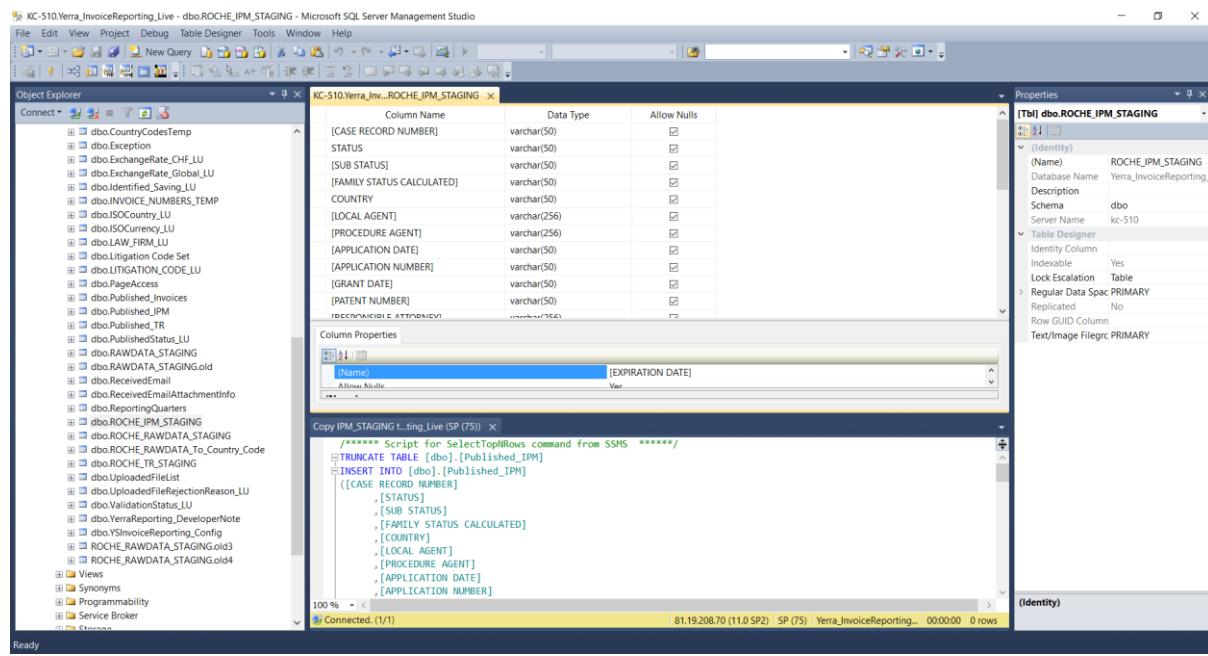
Junior developer: "Can you give me insight of how you perform updating the IPM data into a new SQL table{s}?

Is the excel imported directly into SQL?"

Lead developer: "It's much more simple, I use "Import Data" to import IPM into the staging tables (I first truncate this) and then I run a SQL script to map out into a normalised table with the correct data types (please see attached)."

PTO....

See below image showing the SQL query and the table.



6.3 Appendix

Yerra Clearly Visual Studio Class Diagram1.jpeg:

