Legacy Delivery Readme

# Stewart Hyde

### ***Introduction***

This project demonstrates some of technology that I learn over the last decade to make older legacy Microsoft Visual Studio 2008 MFC source code modernized to newer technologies and at same time being able to maintain existing product line until such day it can be updated. I believe this is extremely important in today’s world with many companies having out of date software.

In doing research for .Net Core 3.1, I found that there is simpler method of handling a lot of what was done previous provided logic to handle the latest .Net Core library.

### ***Goals***

* Simple legacy application use to demonstrate the techniques in Microsoft Visual Studio 2008 MFC
* Visual Studio 2019 Community Edition bridge to C++ with .Net Core 3.1
* Log4Not Logging component used by C++ application via .Net 3.5
* Improved call stack logging on exceptions in C++ and in .Net 3.5 client side
* Instead of Google Protocol and WCF made the application used .Net Core web interface and talking to database directly for legacy clients. Modern clients have no client logic all and completely portable.
* Router will be new WorkerService which could be a Windows Service
* .Net Core 3.1 side uses SeriLog which redirects Microsoft Log4Net logging style to text file
* .Net Core 3.1 side uses Unit Testing
* .Net Core 3.1 side can migrate data to SQL
* Client side .Net 3.5 can shell to .Net Core 3.1 command line exe which is use transfer XML
* Modern (03/2020) Blazer Application on Internet with extremely light and portable across multiple devices on client.
* Ability to received requests from server or modern application
* Demo that works on Free Azure web platform with simulated in memory database because of restrictions I found on Azure that limited operations. It not a big deal since Item data and Category can be seed and information stays for lifetime of the web application. A real implementation would have a SQL server database.

### ***Lessons learn***

Many lessons learn why development of this project and I found that technology that I thought was new in the past can be replace with better option. For me, I found that development in past of web application was extremely difficult but with new Microsoft Blazor, I found that it just extension of programming that was done in better and best of all the user interfaces can truly independent of the business logic and most importantly the database. The best example is recent finding that when working with Azure, that Azure SQL database kept loose setting in between builds. But with a set of data services that are not connected directly with user interface, it was rather simple to remove the Web Application depending on SQL Server and for demo purposes swap it out within a hour or two for in memory implementation.

Also, one of fun parts of Microsoft Blazor is the fact that the pages can be component driven and run off Bootstrap 4.4, which has many advantages over the past because this example has absolutely NO JavaScript that was directly written for client. I have found that in a lot of ways this pages that are extended html with bootstrap are like data when they are generated for client. Requiring a browser but no code on the client with real work handle on the server.

This has been a wonderful learning experience and it could never happen with out the excellent videos from Tim Covey on You Tube and especially the Blazor in Depth course.

### ***Other Documents provided***

***Legacy-Delivery Application***

This document description the aspects of legacy C++ Application in this project. Besides the inclusion of screen of the C++ application it describes some of goals. This was initially done, and the same application was creating but in real world this could be like in my previous jobs, a huge application that been developed over the years and running on old system and hard to maintain.

***Modern-Delivery Application***

This document in based on lessons learn from pass experience with the struggles of developing on legacy software and then in contrast experiencing the advance in technology especially with what I found with Microsoft .Net Core 3.1 and Blazor web applications.

***Modernizing Legacy Software***

This document I hope is what this entire project is all about. In a real world, Legacy application maybe something totally different than what is provided in this project. But my hope is that this project gives developers some insights on what is possible. In an ideal world, it may not possible to start from scratch on a project but if it can change so over time older software can use new development platforms which allow other developers to share in the project and make enhancements.