**Election Process Exercise**

**Description**

The exercise has been coded in C# in a .NET 4.5 environment. The code is part of Visual Studio 2012 version.

A simple GUI is presented with a start button and a table showing the part results in seat won order and the sum of the vote share. The top three are shown and others are shown in fourth place using a code of OTHER.

The files are ingested from a processing directory and if successfully processed put into a done directory. Any failures are put into a failed directory. These are configurable via constants.

Schema validation on the files are carried out via a generated xsd file.

Failures are emailed to a supervisor. But for the purposes of this exercise if the email is not setup then a simple message box is presented on screen to show a failure has occurred.

An assumption has been made if there is a Party Code without any voting information then this would count as zero votes. In an actual work situation this, of course, would be clarified.

Also the minimum of information is obtained from the xml to carry out the exercise. In a real situation, it is likely, all information would be recorded and stored so can be used for a variety of purposes.

**Setup Required**

In the ElectionProcess.cs file there are constants which can be changed.

The mandatory constants, for directory locations, to change are the following:

// specific file location constants to run this exercise

const string PROCESS\_DIRECTORY = @"C:\election\process";

const string DONE\_DIRECTORY = @"C:\election\done";

const string FAILED\_DIRECTORY = @"C:\election\failed";

const string ELECTION\_XSD = @"c:\election\election.xsd";

There are other constants for email setup for gmail, but not mandatory to do as indicated above.

The zip file contains the solution for the code and election.xsd schema file used for validation.