How to use EP-QRisk3 in a Visual Studio application.

1. Create a new console app, targeting .NET Core 6.0
2. Add a reference to the EP\_Qrisk3.ddl and the ep\_models.dll supplied by Endeavour Health

A screenshot of a computer

Description automatically generated

Your dependencies should now look like this:

A screenshot of a computer program

Description automatically generated

1. Replace the contents of the program.cs file with the following

using ep\_models;

using EP\_QRisk3;

using static Core.EPStandardDefinitions;

using static ep\_models.EngineResultModel;

EP\_QRisk3.Controller controller = new EP\_QRisk3.Controller();

var model = new EPInputModel();

model.age = 33;

model.CVD = false;

model.sex = Core.EPStandardDefinitions.Gender.Male;

model.requestedEngines = new List<Core.EPStandardDefinitions.Engines>();

model.requestedEngines.Add(Engines.QRisk3);

var result = controller.GetScore(EPInputModel: model);

var firstResult = result.EngineResults.First();

Console.WriteLine("Engine Name: " + firstResult.EngineName);

Console.WriteLine("Engine Version: " + firstResult.EngineVersion);

foreach (var predictionResult in firstResult.Results)

{

Console.WriteLine("------------------------");

Console.WriteLine("Score Id: " + predictionResult.id);

Console.WriteLine("Score score: " + predictionResult.score);

Console.WriteLine("Score typicalScore: " + predictionResult.typicalScore);

}

Console.WriteLine("------------------------");

Console.WriteLine("Press any key to finish.");

Console.ReadKey();

1. Run the program, you should see the following output:

A screenshot of a computer program

Description automatically generated

Full details of the EPInputModel can be found in the supplied HelpFiles.

How to run the Batch Processor.

The batch processor will process a folder of JSON Input Files. Example files are provided in the TestPack/Params and TestPack/First200 folders

Open a command prompt and call the EXE passing in the path to your input files, for example:



The batch processor will try and process any file in the folder with the “input.json” extension.

Any file that does not have this extension will be skipped.

See below for expected output when pointed to the TestPack/First200 folder:

A screen shot of a computer

Description automatically generated