

Implementation details | Library references | Acknowledgements:

- The application is developed in Visual Studio, tested on both VS2010 and VS2013.
- For JSON parsing in VS platform, [JsonCPP](#) library is used. Following three files generated from JsonCPP are included in the code repository:
 - o json.cpp
 - o json/json.h
 - o json/json-forwards.h
- As the first time JSON user, I leveraged this [tutorial code snippets](#) for the profile schema parsing.

The relevant implementations are reflected in following files in the code repository:

- o IJsonDeserializable.h
- o CJsonDeserializer.h
- o CJsonDeserializer.cpp
- Used information on StackOverflow to find out pertinent solution with using the JSON API.

How to use the quote application:

- When running from Visual Studio, filename (containing no space) should be given considering the relative path to the input file from the project directory. For instance, as the `inputs` directory is put under the project source directory, to provide profile input we could provide `"inputs\Rectangle.json"`.
- However, when running the application from the command line (either from the `Debug` or the `Release` directory that are also located under the project source directory), the input filename should be given as `"..\inputs\Rectangle.json"`. In either case, an absolute path should work fine.
- A sample execution from the command line:

```
C:\Users\Mohammad>cd Desktop\PlethoraQuoting\Debug
C:\Users\Mohammad\Desktop\PlethoraQuoting\Debug>PlethoraQuoting.exe ..\inputs\Rectangle.json
profile name: ..\inputs\Rectangle.json
Quote: 14.10
C:\Users\Mohammad\Desktop\PlethoraQuoting\Debug>PlethoraQuoting.exe ..\inputs\ExtrudeCircularArc.json
profile name: ..\inputs\ExtrudeCircularArc.json
Quote: 4.47
C:\Users\Mohammad\Desktop\PlethoraQuoting\Debug>PlethoraQuoting.exe ..\inputs\CutCircularArc.json
profile name: ..\inputs\CutCircularArc.json
Quote: 4.06
```

How to improve the implementation:

- A sophisticated approach that computes the convex hull of a set of straight and circular line segments can be computed by the proposed technique I came to learn from following paper.

Convex hull of a planar set of straight and circular line segments, *International Journal for Computer Aided Engineering and Software*, Vol 16, Issue 8, 1999.

<http://www.emeraldinsight.com/doi/pdfplus/10.1108/02644409910304086>

- Once the convex hull is computed, an oriented minimum boundary box can be computed using different techniques, including the rotating calipers algorithm. I came to learn some good implementation approach from the below article.

<http://www.geometrictools.com/Documentation/MinimumAreaRectangle.pdf>