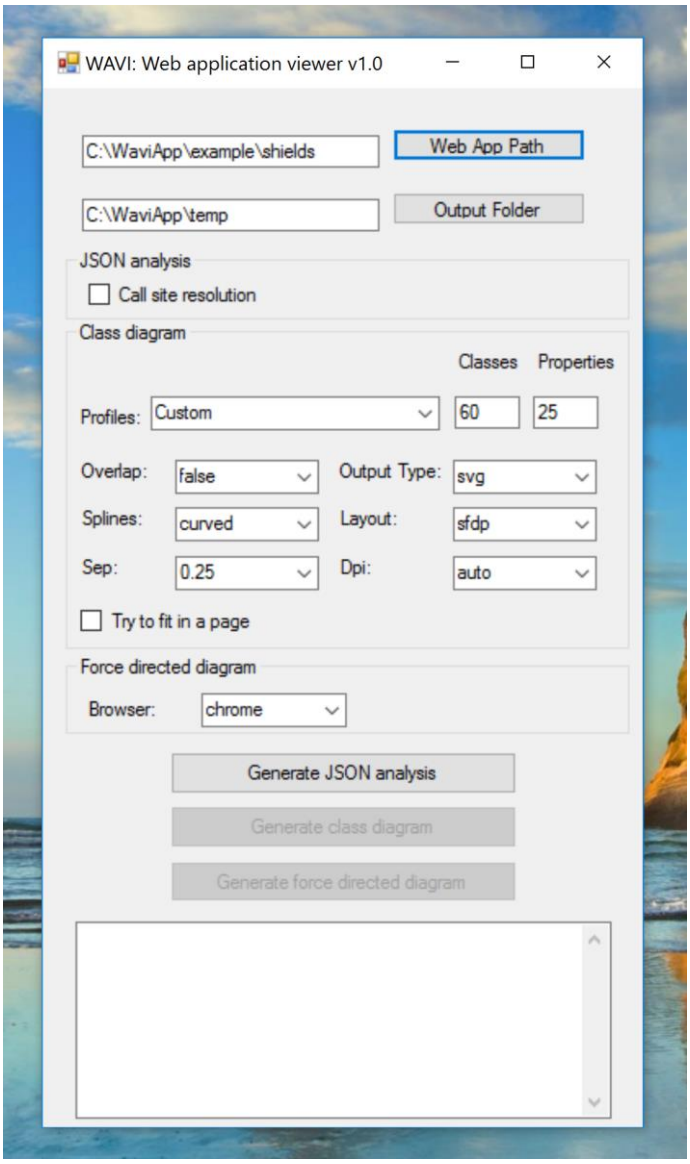
WAVI STANDALONE - QUICK MANUAL



SYSTEM REQUIREMENTS

- Windows 7/8/10 64bit
- Microsoft .NET Framework 4.5.2 or better

QUICKSTART

1. Execute “app.exe”

GENERATE JSON ANALYSIS (TEXTUAL REPRESENTATION)

1. Select the path to the web application you want to reverse engineer.
2. Select the output folder (where textual and graphical representation will be saved).
3. Choose if you want to include call site resolution (link between function call site and function declaration) by checking the « call site resolution » checkbox.
4. Click on Generate JSON analysis.

GENERATE FORCE DIRECTED DIAGRAM

1. First you need to generate the textual representation.
2. You can choose between two browsers to show the force directed diagram (firefox or chrome) in the browser dropdown.
3. Click on “Generate force directed diagram” button.

GENERATE CLASS DIAGRAM

1. First you need to generate the textual representation.
1. You can choose a profile in the profile dropdown (profile change the amount of element that form a class and a property). There is also a possibility to change it manually with a number from 0 to 100 where 0 mean all elements will be classes and 100 only important element will be classes. 0 properties means that all remaining elements will be shown as properties while 100 means no elements will be shown as properties.
2. There are six Graphviz properties[1]:
    a. **Overlap**: Determines if and how node overlaps should be removed. Nodes are first enlarged using the sep attribute. If "true" , overlaps are retained. If the value is "scale", overlaps are removed by uniformly scaling in x and y. If the value converts to "false", and it is available, Prism, a proximity graph-based algorithm, is used to remove node overlaps. This can also be invoked explicitly with "overlap=prism". This technique starts with a small scaling up, controlled by the overlap_scaling attribute, which can remove a significant portion of the overlap. The prism option also accepts an optional non-negative integer suffix. This can be used to control the number of attempts made at overlap removal. By default, overlap="prism" is equivalent to overlap="prism1000". Setting overlap="prism0" causes only the scaling phase to be run.
    b. **Splines:** Controls how, and if, edges are represented. If true, edges are drawn as splines routed around nodes; if false, edges are drawn as line segments. If set to none or "", no edges are drawn at all. The values line and spline can be used as synonyms for false and true, respectively. In addition, the value polyline specifies that edges should be drawn as polylines. The value ortho specifies edges should be routed as polylines of axis-aligned segments. The value curved specifies edges should be drawn as curved arcs.
    c. **Sep:** Specifies margin to leave around nodes when removing node overlap. This guarantees a minimal non-zero distance between nodes. If the attribute begins with a

[1] Source : http://www.graphviz.org/doc/info/attrs.html

plus sign '+', an additive margin is specified. That is, "+w,h" causes the node's bounding box to be increased by w points on the left and right sides, and by h points on the top and bottom. Without a plus sign, the node is scaled by 1 + w in the x coordinate and 1 + h in the y coordinate. If only a single number is given, this is used for both dimensions.

d. **Layout:** Specifies the name of the layout algorithm to use, such as "dot" or "neato". Normally, graphs should be kept independent of a type of layout. In some cases, however, it can be convenient to embed the type of layout desired within the graph. For example, a graph containing position information from a layout might want to record what the associated layout algorithm was.

e. **Dpi :** This specifies the expected number of pixels per inch on a display device. For bitmap output, this guarantees that text rendering will be done more accurately, both in size and in placement. For SVG output, it is used to guarantee that the dimensions in the output correspond to the correct number of points or inches.

f. **Output :** The output is the format of the file “svg”,”png”,”pdf”,”jpeg”. This should be noted that big diagram should be only generated in svg because Graphviz is limited in image dimension.

3. Click “Generate class diagram”.