



| | |
|---------------------|---|
| PROJECT REVIEWED | GEOJSON PARSER (WRITTEN BY TEAM E) |
| REVIEWED BY | (TEAM D) |
| SUBJECT | SECURE CODING PRINCIPLES AND PRACTICES |

GeoJSON - Introduction

An open standard format for encoding collections of simple geographical features along with their non-spatial attributes using JavaScript Object Notation. The features include :

- points (addresses/locations)
- line strings (streets/highways)
- polygons (countries) and multi-part collections of these types

GeoJSON feature not only represent entities of physical world

E.g mobile routing and navigation apps describe their service coverage using GeoJSON

GeoJSON (contd..)

Unlike GIS Standards which is written and maintained by formal standards organization, GeoJSON is maintained by an *Internet working group of developers*.

GeoJson specification available <http://geojson.org/geojson-spec.html>

(parts 2.1.2, 2.1.3, 2.1.4, 2.1.5, 2.1.6 and 2.1.7)

A notable offspring of GeoJSON is TopoJSON, an extension of GeoJSON that encodes geospatial topology and that typically provides smaller file sizes

GeoJSON Parser

Supported types -

Point, MultiPoint, LineString, MultiLineString, Polygon, MultiPolygon

Target Platform -

Windows 8.1, compiled using Visual Studio 2012 ,Language: C++11

Dependencies -

STD library, Boost library

Input -

Input GeoJson format file. E.g. { "type": "Point", "coordinates": [2, 0] }

Output -

File parsed correctly (if format is correct), Otherwise Error: Message

GeoJSON Parser –Manual Review

| | |
|------------------------------|--|
| Code compilation | <i>Compiled & Run successfully, Need Boost library path to be included</i> |
| Code commenting- | <i>Inline, Descriptive, Function/Class comments missing</i> |
| Sanity checks- | <i>Input Sanity Checks are not exhaustive</i> |
| Error handling- | <i>Fairly good</i> |
| Limited format- | <i>Covers only 6 objects, Specifications has many more</i> |
| 'GeometryType' cases- | <i>Enumerators GEOMETRYCOLLECTION, FEATURE, FEATURECOLLECTION, GEOMETRY_TYPE_COUNT not handled</i> |
| Coding conventions | <i>Conventions like Variable naming could have added for better code Readability</i> |
| Size of File Name | <i>char filenameetemp[65536] ? (Windows limits file names to 260 chars)</i> |

GeoJSON Parser - Input and Output

Program crashed for following Inputs

1. Empty Input file with .json extension
2. Improper format checking
 - a.

```
{ "type": "MultiLineString",  
  "coordinates": [ [100.0, 0.0], [101.0, 1.0] ]  
                  [100.0, 0.0], [101.0, 1.0]  
                }  
                }
```
 - b.

```
{ "type": "MultiPoint",  
  "coordinates": [ [100.0, 0.0] ] [ [101.0, 1.0] ]  
                }
```

Valid Input rejection

```
{ "type": "Point", "coordinates": [2, 0] },  
{ "type": "Point", "coordinates": [2, 0] }
```

Output - Error: Invalid format, File not parsed

Sample Input (<http://geojson.org/geojson-spec.html#examples>)

```
{ "type": "FeatureCollection",  
  "features": [  
    { "type": "Feature",  
      "geometry": { "type": "Point", "coordinates": [102.0, 0.5] },  
      "properties": { "prop0": "value0" }  
    },  
    { "type": "Feature",  
      "geometry": {  
        "type": "LineString",  
        "coordinates": [  
          [102.0, 0.0], [103.0, 1.0], [104.0, 0.0], [105.0, 1.0]  
        ]  
      },  
      "properties": {  
        "prop0": "value0",  
        "prop1": 0.0  
      }  
    },  
  ]  
}
```

Analysis using Automated Tools - PREFast Output

1. main.cpp(72): warning C4820: 'Array' : '3' bytes padding added after data member 'Array::mIsPointer'
2. main.cpp(35): warning C4265: '_Node' : class has virtual functions, but destructor is not virtual instances of this class may not be destructed correctly (*line no. 66,80 also has similar warnings*)
3. main.cpp(102): warning C4365: 'argument' : conversion from 'int' to 'unsigned int',signed/unsigned mismatch (*Total 33 instances of similar class of warnings found at different line numbers*)
4. main.cpp(726): warning C4365: 'initializing' : conversion from 'unsigned int' to 'int', signed/unsigned mismatch
5. main.cpp(805): warning C4571: Informational: catch(...) semantics changed since Visual C++ 7.1; structured exceptions (SEH) are no longer caught

PREFast Output (contd..)

6. main.cpp(945): warning C4061: enumerator 'GEOMETRY_TYPE_COUNT' in switch of enum 'GeometryType' is not explicitly handled by a case label
7. main.cpp(945): warning C4061: enumerator 'FEATURE_COLLECTION' in switch of enum 'GeometryType' is not explicitly handled by a case label
8. main.cpp(945): warning C4061: enumerator 'FEATURE' in switch of enum 'GeometryType' is not explicitly handled by a case label
9. main.cpp(945): warning C4061: enumerator 'GEOMETRY_COLLECTION' in switch of enum 'GeometryType' is not explicitly handled by a case label
10. main.cpp(542): warning C4100: 'envp' : unreferenced formal parameter (similar warnings for argc and argv)

Analysis using Automated Tools - PReFast Output

1. main.cpp(72): warning C4820: 'Array' : '3' bytes padding added after data member 'Array::mIsPointer'
2. main.cpp(35): warning C4265: '_Node' : class has virtual functions, but destructor is not virtual instances of this class may not be destructed correctly (*line no. 66,80 also has similar warnings*)
3. main.cpp(102): warning C4365: 'argument' : conversion from 'int' to 'unsigned int',signed/unsigned mismatch (*Total 33 instances of similar class of warnings found at different line numbers*)
4. main.cpp(726): warning C4365: 'initializing' : conversion from 'unsigned int' to 'int', signed/unsigned mismatch
5. main.cpp(805): warning C4571: Informational: catch(...) semantics changed since Visual C++ 7.1; structured exceptions (SEH) are no longer caught

Specific Warnings(by Code Analysis) -PREFast

1. C6262 Excessive stack usage- Function uses '66208' bytes of stack: exceeds /analyze: stacksize '16384'. Consider moving some data to heap. Parser - main.cpp (Line 542)
2. C28182 Dereferencing a copy of a null pointer- Dereferencing NULL pointer. 'parent' contains the same NULL value as 'node->mParent' did.
Line 185: 'parent' may be NULL
Line 187: 'parent' is dereferenced, but may still be NULL
Parser - main.cpp (Line 187)

Analysis using Automated Tools - CPPCheck Output

1. [Parser\main.cpp:199]: (style) C-style pointer casting (*Total 11 instances of similar class of warnings found at different line numbers*)
2. [Parser\main.cpp:674]: (style) Scope of the variable 'node' can be reduced.
3. [Parser\main.cpp:676]: (style) Scope of the variable 'previous' can be reduced.
4. [Parser\main.cpp:678]: (style) Scope of the variable 'bColonFlag' can be reduced.
5. [Parser\main.cpp:679]: (style) Scope of the variable 'nColonCnt' can be reduced
6. [Parser\main.cpp:796]: (style) Variable 'bColonFlag' is assigned a value that is never used.
7. [Parser\main.cpp:47]: (performance) Variable 'mLeftSide' is assigned in constructor body. Consider performing initialization in initialization list.

A silver metal spiral binding is visible on the left edge of the page, consisting of a series of loops that hold the pages together.

THANK YOU