

Report for JSON parser

Overview

This is a library for parsing JSON. It is contained in json.h and json.cpp. There is also a main that allows you to parse something from standard input.

Implementation

We have written a code with not very much state. Thanks to some modern C++11 features, e.g. smart pointers, the code should not be error prone.

The code is not much stateful, which should make it easy for reviewing. In fact, most of the code could be rewritten to Haskell very straightforwardly. There is an exception in encoding conversions. We use iconv library, because we haven't found anything better in C++11 implemented in G++.

The parsing starts by calling `Json::Value::parseValue`, which decides the particular type to delegate. Recursive structures (e.g. `[[[[[[[[[[...]]]]]]]]`) are handled by recursion.

Features, possible pitfalls for users and known issues

The parser is sometimes more permissive than the specification. For example, we allow `[1, 2,]`. This is not guaranteed in future versions. However, even these cases should not lead to memory corruption or similar issues.

We assume correct UTF-8 on input with an UTF-8 locale. If this is not satisfied, library might output an incorrect UTF-8 string. However, even this should not lead to memory corruption.

The library does not filter special control characters. If these are an issue for you, you have to handle them. If you fail to do so, it may lead to control character injection in console application, DoS in XML and so on.

When generating a JSON output, strings with some special characters are not generated correctly. We consider this to be a minor issue. Since the main task was to parse it, not to serialize it, we have left this issue unfixed.

Multiple values in object with the same key may lead to unspecified results, but it is safe from memory consistency point of view.

Arrays does not seem to be well pretty-printed, but this is a minor issue.