Json parsers

There are some libraries at Json.org . I compare all of them below

	needs additional libraries	liciense	builds by	comment
jost	+	Public Domain	make	
cajun	-	BSD	make	
jsoncpp	-	Public Domain	scons	
zoolib	-	MIT	make	cross-platform framework
jaula	-	LGPL	make	needs FlexLexer lib
json-spirit	+	MIT		developed for windows

I think the best solution is cajun. Here is sample of code, that shows how does it work. We have such json request in file example.in

```
{
    "description": "subscribe to channel",
    "type": "object",
    "properties": {
        "user": {"type":"string"},
        "channel":{"type":"string"}
    }
}
```

Below you can see programm that prints its data to std::cout.

```
#include "elements.h"
#include "quick.h"
#include "writer.h"
#include "reader.h"
#include "cast.h"
#include "exception.h"
#include <iostream>
#include <fstream>

using namespace json;

int main(int argc,char** argv){
    std::ifstream in("example.in");
    Element elemRoot;//Root element of request
    Reader::Read(elemRoot,in); // Reading document into elemRoot
    QuickInterpreter interpeter(elemRoot);// This class allows us to get
access to elemnts of json request like in std::map
    const String& description=interpeter["description"];
    const String& type=interpeter["type"];
    const String& type1=interpeter["properties"]["user"]["type"];
    const String& type2=interpeter["properties"]["channel"]["type"];
    std::cout << "description:  " << std::string(description) << std::endl;
```

```
    std::cout << "type:  " << std::string(type) << std::endl;
        std::cout << "properties/user/type:  " << std::string(type1) <<
std::endl;
        std::cout << "properties/channel/type:  " <<  std::string(type2) <<
std::endl;
    return 0;
}
```

From:
http://wiki.osll.ru/ - **Open Source & Linux Lab**

Permanent link:
http://wiki.osll.ru/doku.php/etc:blog:json_parsers?rev=1258285732

Last update: **2009/11/15 14:48**