

```

1  /**
2   * Copyright (C) 2011 Nokia Inc. All rights reserved.
3   * Copyright (C) 2012 Google Inc. All rights reserved.
4   *
5   * This library is free software; you can redistribute it and/or
6   * modify it under the terms of the GNU Library General Public
7   * License as published by the Free Software Foundation; either
8   * version 2 of the License, or (at your option) any later version.
9   *
10  * This library is distributed in the hope that it will be useful,
11  * but WITHOUT ANY WARRANTY; without even the implied warranty of
12  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
13  * Library General Public License for more details.
14  *
15  * You should have received a copy of the GNU Library General Public License
16  * along with this library; see the file COPYING.LIB. If not, write to
17  * the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor,
18  * Boston, MA 02110-1301, USA.
19  *
20  */
21
22  #include "config.h"
23  #include "QuotesData.h"
24
25  namespace WebCore {
26
27  static size_t sizeForQuotesDataWithQuoteCount(unsigned count)
28  {
29      return sizeof(QuotesData) + sizeof(std::pair<String, String>) * count;
30  }
31
32  Ref<QuotesData> QuotesData::create(const Vector<std::pair<String, String>>& quotes)
33  {
34      void* slot = fastMalloc(sizeForQuotesDataWithQuoteCount(quotes.size()));
35      return adoptRef(*new (NotNull, slot) QuotesData(quotes));
36  }
37
38  QuotesData::QuotesData(const Vector<std::pair<String, String>>& quotes)
39      : m_quoteCount(quotes.size())
40  {
41      for (unsigned i = 0; i < m_quoteCount; ++i)

```

```

42         new (NotNull, &m_quotePairs[i]) std::pair<String, String>(quotes[i]);
43     }
44
45     QuotesData::~QuotesData()
46     {
47         for (unsigned i = 0; i < m_quoteCount; ++i)
48             m_quotePairs[i].~pair<String, String>();
49     }
50
51     const String& QuotesData::openQuote(unsigned index) const
52     {
53         if (!m_quoteCount)
54             return emptyString();
55
56         if (index >= m_quoteCount)
57             return m_quotePairs[m_quoteCount - 1].first;
58
59         return m_quotePairs[index].first;
60     }
61
62     const String& QuotesData::closeQuote(unsigned index) const
63     {
64         if (!m_quoteCount)
65             return emptyString();
66
67         if (index >= m_quoteCount)
68             return m_quotePairs[m_quoteCount - 1].second;
69
70         return m_quotePairs[index].second;
71     }
72
73     bool operator==(const QuotesData& a, const QuotesData& b)
74     {
75         if (a.m_quoteCount != b.m_quoteCount)
76             return false;
77
78         for (unsigned i = 0; i < a.m_quoteCount; ++i) {
79             if (a.m_quotePairs[i] != b.m_quotePairs[i])
80                 return false;
81         }
82
83         return true;

```

```
84 }  
85  
86 } // namespace WebCore  
87
```