

JDOMNamespacePointer.java - NamespacePointer.java

```

1/*
2 * $Header:
3   /home/cvs/jakarta-commons/jxpath/src/java/org/apache/commons/jxpath/ri/model/jdom/J
4   DOMNamespacePointer.java,v 1.6 2003/03/11 00:59:33 dmitri Exp $
5 * $Revision: 1.6 $
6 * $Date: 2003/03/11 00:59:33 $
7 *
8 * =====
9 * The Apache Software License, Version 1.1
10 *
11 * Copyright (c) 1999-2003 The Apache Software Foundation. All rights
12 * reserved.
13 *
14 * Redistribution and use in source and binary forms, with or without
15 * modification, are permitted provided that the following conditions
16 * are met:
17 *
18 * 1. Redistributions of source code must retain the above copyright
19 * notice, this list of conditions and the following disclaimer.
20 *
21 * 2. Redistributions in binary form must reproduce the above copyright
22 * notice, this list of conditions and the following disclaimer in
23 * the documentation and/or other materials provided with the
24 * distribution.
25 *
26 * 3. The end-user documentation included with the redistribution, if
27 * any, must include the following acknowledgement:
28 * "This product includes software developed by the
29 * Apache Software Foundation (http://www.apache.org/)."
30 * Alternately, this acknowledgement may appear in the software itself,
31 * if and wherever such third-party acknowledgements normally appear.
32 *
33 * 4. The names "The Jakarta Project", "Commons", and "Apache Software
34 * Foundation" must not be used to endorse or promote products derived
35 * from this software without prior written permission. For written
36 * permission, please contact apache@apache.org.
37 *
38 * 5. Products derived from this software may not be called "Apache"
39 * nor may "Apache" appear in their names without prior written
40 * permission of the Apache Group.
41 *
42 * THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED
43 * WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
44 * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
45 * DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR
46 * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
47 * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
48 * LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
49 * USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
50 * ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
51 * OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
52 * OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
53 * SUCH DAMAGE.
54 *
55 * =====

```

```

1/*
2 * $Header:
3   /home/cvspublic/jakarta-commons/jxpath/src/java/org/apache/commons/jxpath/ri/model/
4   dom/NamespacePointer.java,v 1.4 2002/05/29 00:40:58 dmitri Exp $
5 * $Revision: 1.4 $
6 * $Date: 2002/05/29 00:40:58 $
7 *
8 * =====
9 * The Apache Software License, Version 1.1
10 *
11 * Copyright (c) 1999-2001 The Apache Software Foundation. All rights
12 * reserved.
13 *
14 * Redistribution and use in source and binary forms, with or without
15 * modification, are permitted provided that the following conditions
16 * are met:
17 *
18 * 1. Redistributions of source code must retain the above copyright
19 * notice, this list of conditions and the following disclaimer.
20 *
21 * 2. Redistributions in binary form must reproduce the above copyright
22 * notice, this list of conditions and the following disclaimer in
23 * the documentation and/or other materials provided with the
24 * distribution.
25 *
26 * 3. The end-user documentation included with the redistribution, if
27 * any, must include the following acknowledgement:
28 * "This product includes software developed by the
29 * Apache Software Foundation (http://www.apache.org/)."
30 * Alternately, this acknowledgement may appear in the software itself,
31 * if and wherever such third-party acknowledgements normally appear.
32 *
33 * 4. The names "The Jakarta Project", "Commons", and "Apache Software
34 * Foundation" must not be used to endorse or promote products derived
35 * from this software without prior written permission. For written
36 * permission, please contact apache@apache.org.
37 *
38 * 5. Products derived from this software may not be called "Apache"
39 * nor may "Apache" appear in their names without prior written
40 * permission of the Apache Group.
41 *
42 * THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED
43 * WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
44 * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
45 * DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR
46 * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
47 * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
48 * LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
49 * USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
50 * ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
51 * OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
52 * OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
53 * SUCH DAMAGE.
54 *
55 * =====

```

JDOMNamespacePointer.java - NamespacePointer.java

```

55 * This software consists of voluntary contributions made by many
56 * individuals on behalf of the Apache Software Foundation and was
57 * originally based on software copyright (c) 2001, Plotnix, Inc,
58 * <http://www.plotnix.com/>.
59 * For more information on the Apache Software Foundation, please see
60 * <http://www.apache.org/>.
61 */
62 package org.apache.commons.xpath.ric.model.jdom;
63
64 import org.apache.commons.xpath.ric.QName;
65
66 import org.apache.commons.xpath.ric.model.NodePointer;
67 /**
68 * Represents a namespace node.
69 *
70 * @author Dmitri Plotnikov
71 * @version $Revision: 1.6 $ $Date: 2003/03/11 00:59:33 $
72 */
73 public class JDOMNamespacePointer extends NodePointer {
74     private String prefix;
75     private String namespaceURI;
76
77     public JDOMNamespacePointer(NodePointer parent, String prefix) {
78         super(parent);
79         this.prefix = prefix;
80     }
81
82     public JDOMNamespacePointer(
83         NodePointer parent,
84         String prefix,
85         String namespaceURI)
86     {
87         super(parent);
88         this.prefix = prefix;
89         this.namespaceURI = namespaceURI;
90     }
91
92     public QName getName() {
93         return new QName(getNamespaceURI(), prefix);
94     }
95
96     public Object getBaseValue() {
97         return null;
98     }
99
100     public boolean isCollection() {
101         return false;
102     }
103
104     public int getLength() {
105         return 1;
106     }

```

```

55 * This software consists of voluntary contributions made by many
56 * individuals on behalf of the Apache Software Foundation and was
57 * originally based on software copyright (c) 2001, Plotnix, Inc,
58 * <http://www.plotnix.com/>.
59 * For more information on the Apache Software Foundation, please see
60 * <http://www.apache.org/>.
61 */
62 package org.apache.commons.xpath.ric.model.dom;
63
64 import org.apache.commons.xpath.ric.Compiler;
65 import org.apache.commons.xpath.ric.QName;
66 import org.apache.commons.xpath.ric.compiler.NodeTest;
67 import org.apache.commons.xpath.ric.compiler.NodeTypeTest;
68 import org.apache.commons.xpath.ric.model.NodePointer;
69
70 /**
71 * Represents a namespace node.
72 *
73 * @author Dmitri Plotnikov
74 * @version $Revision: 1.4 $ $Date: 2002/05/29 00:40:58 $
75 */
76 public class NamespacePointer extends NodePointer {
77     private String prefix;
78     private String namespaceURI;
79
80     public NamespacePointer(NodePointer parent, String prefix) {
81         super(parent);
82         this.prefix = prefix;
83     }
84
85     public NamespacePointer(NodePointer parent, String prefix, String
namespaceURI) {
86         super(parent);
87         this.prefix = prefix;
88         this.namespaceURI = namespaceURI;
89     }
90
91     public QName getName() {
92         return new QName(getNamespaceURI(), prefix);
93     }
94
95     public Object getBaseValue() {
96         return null;
97     }

```

JDOMNamespacePointer.java - NamespacePointer.java

```

107
108 public Object getImmediateNode() {
109     return getNamespaceURI();
110 }
111
112 public String getNamespaceURI() {
113     if (namespaceURI == null) {
114         namespaceURI = parent.getNamespaceURI(prefix);
115     }
116     return namespaceURI;
117 }
118
119 public boolean isLeaf() {
120     return true;
121 }
122
123 /**
124  * Throws UnsupportedOperationException.
125  */
126 public void setValue(Object value) {
127     throw new UnsupportedOperationException("Cannot modify a namespace");
128 }
129
130 public String asPath() {
131
132     StringBuffer buffer = new StringBuffer();
133     if (parent != null) {
134         buffer.append(parent.asPath());
135         if (buffer.length() == 0
136             || buffer.charAt(buffer.length() - 1) != '/') {
137             buffer.append('/');
138         }
139         buffer.append("namespace::");
140         buffer.append(prefix);
141         return buffer.toString();
142     }
143
144 public int hashCode() {
145     return prefix.hashCode();
146 }
147
148 public boolean equals(Object object) {
149     if (object == this) {
150         return true;
151     }
152
153     if (!(object instanceof JDOMNamespacePointer)) {
154         return false;
155     }

```

```

98
99 public Object getNodeValue() {
100     return getNamespaceURI();
101 }
102
103 public String getNamespaceURI() {
104     if (namespaceURI == null) {
105         namespaceURI = parent.getNamespaceURI(prefix);
106     }
107     return namespaceURI;
108 }
109
110 public boolean isLeaf() {
111     return true;
112 }
113
114 /**
115  * Throws UnsupportedOperationException.
116  */
117 public void setValue(Object value) {
118     throw new UnsupportedOperationException("Cannot modify DOM trees");
119 }
120
121 public boolean testNode(NodeTest nodeTest) {
122     return nodeTest == null ||
123         ((nodeTest instanceof NodeTypeTest) &&
124         ((NodeTypeTest) nodeTest).getNodeTest() ==
125         Compiler.NODE_TYPE_NODE);
126 }
127
128 public String asPath() {
129     StringBuffer buffer = new StringBuffer();
130     if (parent != null) {
131         buffer.append(parent.asPath());
132         buffer.append('/');
133     }
134     buffer.append("namespace::");
135     buffer.append(prefix);
136     return buffer.toString();
137 }
138
139 public int hashCode() {
140     return prefix.hashCode();
141 }
142
143 public boolean equals(Object object) {
144     if (object == this) {
145         return true;
146     }
147
148     if (!(object instanceof NamespacePointer)) {
149         return false;
150     }

```

JDOMNamespacePointer.java - NamespacePointer.java

```
156
157 JDOMNamespacePointer other = (JDOMNamespacePointer) object;
158 return prefix.equals(other.prefix);
159 }
160
161 public int compareChildNodePointers(
162     NodePointer pointer1,
163     NodePointer pointer2)
164 {
165     // Won't happen - namespaces don't have children
166     return 0;
167 }
168 }
```

```
150
151 NamespacePointer other = (NamespacePointer) object;
152 return prefix.equals(other.prefix);
153 }
154
155 public int compareChildNodePointers(NodePointer pointer1, NodePointer
pointer2) {
156
157     // Won't happen - namespaces don't have children
158     return 0;
159 }
```