Package javax.xml.xpath

it will be BEST for low memory device

5 - 11 - 08

This package provides an *object-model neutral* API for the evaluation of XPath expressions and access to the evaluation environment.

8 - Jan -09

in next round, see example for the use of function / variable resolver

See:

Description

Interface Summary

XPath	`XPath` provides access to the XPath evaluation environment and expressions.
XPathExpression	`XPathExpression` provides access to compiled XPath expressions.
XPathFunction	`XPathFunction` provides access to XPath functions.
XPathFunctionResolver	`XPathFunctionResolver` provides access to the set of user defined `XPathFunction`s.
XPathVariableResolver	`XPathVariableResolver` provides access to the set of user defined XPath variables.

Class Summary

XPathConstants	XPath constants.
XPathFactory	An `XPathFactory` instance can be used to create `XPath` objects.

Exception Summary

XPathException	`XPathException` represents a generic XPath exception.
XPathExpressionException	`XPathExpressionException` represents an error in an XPath expression.
XPathFactoryConfigurationException	`XPathFactoryConfigurationException` represents a configuration error in a `XPathFactory` environment.
XPathFunctionException	`XPathFunctionException` represents an error with an XPath function.

Package javax.xml.xpath Description

This package provides an *object-model neutral* API for the evaluation of XPath expressions and access to the evaluation environment.

~~The following XML standards apply:~~

- XML Path Language (XPath) Version 1.0

XPath Overview

The XPath language provides a simple, concise syntax for selecting nodes from an XML document. XPath also provides rules for converting a node in an XML document object model (DOM) tree to a boolean, double, or string value. XPath is a W3C-defined language and an official W3C recommendation; the W3C hosts the XML Path Language (XPath) Version 1.0 specification.

XPath started in life in 1999 as a supplement to the XSLT and XPointer languages, but has more recently become popular as a stand-alone language, as a single XPath expression can be used to replace many lines of DOM API code.

XPath Expressions

An XPath *expression* is composed of a *location path* and one or more optional *predicates*. Expressions may also include XPath variables.

The following is an example of a simple XPath expression:

```
/foo/bar
```

This example would select the `<bar>` element in an XML document such as the following:

```
<foo>
<bar/>
</foo>
```

The expression `/foo/bar` is an example of a location path. While XPath location paths resemble Unix-style file system paths, an important distinction is that XPath expressions return *all* nodes that match the expression. Thus, all three `<bar>` elements in the following document would be selected by the `/foo/bar` expression:

```
<foo>
<bar/>
<bar/>
<bar/>
</foo>
```

A special location path operator, `//`, selects nodes at any depth in an XML document. The following example selects all `<bar>` elements regardless of their location in a document:

```
//bar
```

A wildcard operator, *, causes all element nodes to be selected. The following example selects all children elements of a `<foo>` element:

```
/foo/*
```

In addition to element nodes, XPath location paths may also address attribute nodes, text nodes, comment nodes, and processing instruction nodes. The following table gives examples of location paths for each of these node types:

Location Path	Description
`/foo/bar/@id`	Selects the attribute `id` of the `<bar>` element
`/foo/bar/text()`	Selects the text nodes of the `<bar>` element. No distinction is made between escaped and non-escaped character data.
`/foo/bar/comment()`	Selects all comment nodes contained in the `<bar>` element.
`/foo/bar/processing-instruction()`	Selects all processing-instruction nodes contained in the `<bar>` element.

Predicates allow for refining the nodes selected by an XPath location path. Predicates are of the form `[expression]`. The following example selects all `<foo>` elements that contain an `include` attribute with the value of `true`:

```
//foo[@include='true']
```

Predicates may be appended to each other to further refine an expression, such as:

```
//foo[@include='true'][@mode='bar']
```

Using the XPath API

The following example demonstrates using the XPath API to select one or more nodes from an XML document:

```
XPath xpath = XPathFactory.newInstance().newXPath();
String expression = "/widgets/widget";
InputSource inputSource = new InputSource("widgets.xml");
NodeList nodes = (NodeList) xpath.evaluate(expression, inputSource, XPathConstants.
NODESET);
```

XPath Expressions and Types

While XPath expressions select nodes in the XML document, the XPath API allows the selected nodes to be coalesced into one of the following other data types:

- `Boolean`
- `Number`
- `String`

The desired return type is specified by a `QName` parameter in method call used to evaluate the expression, which is either a call to `XPathExpression.evalute(...)` or to one of the `XPath.evaluate(...)` convenience methods. The allowed QName values are specified as constants in the `XPathConstants` class; they are:

- `XPathConstants.NODESET`
- `XPathConstants.NODE`

- XPathConstants.STRING
- XPathConstants.BOOLEAN
- XPathConstants.NUMBER

When a Boolean return type is requested, Boolean.TRUE is returned if one or more nodes were selected; otherwise, Boolean.FALSE is returned.

The String return type is a convenience for retrieving the character data from a text node, attribute node, comment node, or processing-instruction node. When used on an element node, the value of the child text nodes is returned.

The Number return type attempts to coalesce the text of a node to a double data type.

XPath Context

XPath location paths may be relative to a particular node in the document, known as the context. Consider the following XML document:

```
<widgets>
<widget>
<manufacturer/>
<dimensions/>
</widget>
</widgets>
```

The <widget> element can be selected with the following XPath API code:

```
// parse the XML as a W3C Document
DocumentBuilder builder = DocumentBuilderFactory.newInstance().newDocumentBuilder();
Document document = builder.parse(new File("/widgets.xml"));

XPath xpath = XPathFactory.newInstance().newXPath();
String expression = "/widgets/widget";
Node widgetNode = (Node) xpath.evaluate(expression, document, XPathConstants.NODE);
```

With a reference to the <widget> element, a relative XPath expression can now written to select the <manufacturer> child element:

```
XPath xpath = XPathFactory.newInstance().newXPath();
String expression = "manufacturer";
Node manufacturerNode = (Node) xpath.evaluate(expression, widgetNode, XPathConstants.
NODE);
```

- Author Ben Galbraith
- Author Norman Walsh
- Author Jeff Suttor
- See XML Path Language (XPath) Version 1.0
- Since 1.5

---

5 - 11 - 08

7 - Jan -09

it has 12 methods

2 - getter / setter for XPathVariableResolver
2 - getter / setter for XPathFunctionResolver
2 - getter / setter for NamespaceContext
4 - evaluate methods
1 - reset method
1 - compile XPath Expression

javax.xml.xpath

Interface XPath

use XPathExpression instead of xpath for evaluation. it give reusability

`public interface XPath`

`XPath` provides access to the XPath evaluation environment and expressions.

Evaluation of XPath Expressions.	
context	If a request is made to evaluate the expression in the absence of a context item, an empty document node will be used for the context. For the purposes of evaluating XPath expressions, a DocumentFragment is treated like a Document node.
variables	If the expression contains a variable reference, its value will be found through the `XPathVariableResolver` set with `setXPathVariableResolver(XPathVariableResolver resolver)`. An `XPathExpressionException` is raised if the variable resolver is undefined or the resolver returns `null` for the variable. The value of a variable must be immutable through the course of any single evaluation. ohh.... if return null ...throwing exception
functions	If the expression contains a function reference, the function will be found through the `XPathFunctionResolver` set with `setXPathFunctionResolver(XPathFunctionResolver resolver)`. An `XPathExpressionException` is raised if the function resolver is undefined or the function resolver returns `null` for the function. ohh.... if return null ...throwing exception
QNames	QNames in the expression are resolved against the XPath namespace context set with `setNamespaceContext(NamespaceContext nsContext)`.
result	This result of evaluating an expression is converted to an instance of the desired return type. Valid return types are defined in `XPathConstants`. Conversion to the return type follows XPath conversion rules.

An XPath object is not thread-safe and not reentrant. In other words, it is the application's responsibility to make sure that one `XPath` object is not used from more than one thread at any given time, and while the `evaluate` method is invoked, applications may not recursively call the `evaluate` method.

Since:

1.5

Version:

$Revision: 1.6 $, $Date: 2005/11/03 19:34:17 $

Author:

Norman Walsh, Jeff Suttor

See Also:

XML Path Language (XPath) Version 1.0

---

this is best practice instead directly evaluate on XPATH itself

Method Summary

`XPathExpression`	**`compile`**`(java.lang.String expression)` Compile an XPath expression for later evaluation.
`java.lang.String`	**`evaluate`**`(java.lang.String expression, InputSource source)` Evaluate an XPath expression in the context of the specified `InputSource` and return the result as a `String`.
`java.lang.Object`	**`evaluate`**`(java.lang.String expression, InputSource source, QName returnType)` Evaluate an XPath expression in the context of the specified `InputSource` and return the result as the specified type.
`java.lang.String`	**`evaluate`**`(java.lang.String expression, java.lang.Object item)` Evaluate an XPath expr e specified context and return the result as a `String`.
`java.lang.Object`	**`evaluate`**`(java.lang.String expression, java.lang.Object item, QName returnType)` Evaluate an `XPath` expression in the specified context and return the result as the specified type.

it will return text of matched element

it can be casted to Document / Element / Attr accordingly

this is DOM node

this is DOM node

i would say, use inputSource overloaded method **instead** load dom factor / builder then create document finnaly use to evaluate

NamespaceContext	**getNamespaceContext**() Return the current namespace context.
XPathFunctionResolver	**getXPathFunctionResolver**() Return the current function resolver.
XPathVariableResolver	**getXPathVariableResolver**() Return the current variable resolver.
void	**reset**() Reset this XPath to its original configuration.
void	**setNamespaceContext**(NamespaceContext nsContext) Establish a namespace context.
void	**setXPathFunctionResolver**(XPathFunctionResolver resolver) Establish a function resolver.
void	**setXPathVariableResolver**(XPathVariableResolver resolver) Establish a variable resolver.

By default, All of these THREE objects are **null** (namespace context, function resolver, variable resolver)

Method Detail

reset

```
void reset()
```

Reset this XPath to its original configuration.

XPath is reset to the same state as when it was created with XPathFactory.newXPath(). reset() is designed to allow the reuse of existing XPaths thus saving resources associated with the creation of new XPaths.

The reset XPath is not guaranteed to have the same XPathFunctionResolver, XPathVariableResolver or NamespaceContext Objects, e.g. Object.equals(Object obj). It is guaranteed to have a functionally equal XPathFunctionResolver, XPathVariableResolver and NamespaceContext.

that's ok

---

setXPathVariableResolver

```
void setXPathVariableResolver(XPathVariableResolver resolver)
```

Establish a variable resolver.

A `NullPointerException` is thrown if `resolver` is `null`.

Parameters:

`resolver` - Variable resolver.

Throws:

`java.lang.NullPointerException` - If `resolver` is `null`.

---

getXPathVariableResolver

```
XPathVariableResolver getXPathVariableResolver()
```

Return the current variable resolver.

`null` is returned in no variable resolver is in effect.

Returns:

Current variable resolver.

---

setXPathFunctionResolver

```
void setXPathFunctionResolver(XPathFunctionResolver resolver)
```

Establish a function resolver.

A `NullPointerException` is thrown if `resolver` is `null`.

Parameters:

`resolver` - XPath function resolver.

Throws:

`java.lang.NullPointerException` - If `resolver` is `null`

---

getXPathFunctionResolver

`XPathFunctionResolver` **`getXPathFunctionResolver`**`()`

Return the current function resolver.

`null` is returned in no function resolver is in effect.

Returns:

Current function resolver.

---

setNamespaceContext

`void` **`setNamespaceContext`**`(``NamespaceContext` `nsContext)`

Establish a namespace context.

A `NullPointerException` is thrown if `nsContext` is `null`

Parameters:

`nsContext` - Namespace context to use.

Throws:

`java.lang.NullPointerException` - If `nsContext` is `null`.

---

getNamespaceContext

`NamespaceContext` **`getNamespaceContext`**`()`

Return the current namespace context.

`null` is returned in no namespace context is in effect.

Returns:

Current Namespace context.

---

compile

```
XPathExpression compile(java.lang.String expression)
                        throws XPathExpressionException
```

Compile an XPath expression for later evaluation.

it will be better, if any expression has external function or variable, first compile and keep reference of it for whole app. life

If `expression` contains any `XPathFunction`s, they must be available via the `XPathFunctionResolver`. An `XPathExpressionException` will be thrown if the `XPathFunction` cannot be resovled with the `XPathFunctionResolver`.

If `expression` contains any variables, the `XPathVariableResolver` in effect **at compile time** will be used to resolve them.

If `expression` is `null`, a `NullPointerException` is thrown.

Parameters:

`expression` - The XPath expression.

Returns:

Compiled XPath expression.

Throws:

`XPathExpressionException` - If `expression` cannot be compiled.

`java.lang.NullPointerException` - If `expression` is `null`.

---

evaluate

```
java.lang.Object evaluate(java.lang.String expression,
                          java.lang.Object item,
                          QName returnType)
                          throws XPathExpressionException
```

Evaluate an `XPath` expression in the specified context and return the result as the specified type.

See Evaluation of XPath Expressions for context item evaluation, variable, function and `QName` resolution and return type conversion.

If `returnType` is not one of the types defined in `XPathConstants` (`NUMBER`, `STRING`, `BOOLEAN`, `NODE` or `NODESET`) then an `IllegalArgumentException` is thrown.

If a `null` value is provided for `item`, an empty document will be used for the context. If `expression` or `returnType` is `null`, then a `NullPointerException` is thrown.

Parameters:

`expression` - The XPath expression.
`item` - The starting context (a node, for example).
`returnType` - The desired return type.

Returns:

Result of evaluating an XPath expression as an `Object` of `returnType`

Throws:

`XPathExpressionException` - If `expression` cannot be evaluated.
`java.lang.IllegalArgumentException` - If `returnType` is not one of the types defined in `XPathConstants`.
`java.lang.NullPointerException` - If `expression` or `returnType` is `null`.

---

evaluate

```
java.lang.String evaluate(java.lang.String expression,
                          java.lang.Object item)
                          throws XPathExpressionException
```

Evaluate an XPath expression in the specified context and return the result as a `String`.

This method calls `evaluate(String expression, Object item, QName returnType)` with a `returnType` of `XPathConstants.STRING`.

See Evaluation of XPath Expressions for context item evaluation, variable, function and QName resolution and return type conversion.

If a `null` value is provided for `item`, an empty document will be used for the context. If `expression` is `null`, then a `NullPointerException` is thrown.

Parameters:

`expression` - The XPath expression.

`item` - The starting context (a node, for example).

Returns:

The `String` that is the result of evaluating the expression and converting the result to a `String`

Throws:

`XPathExpressionException` - If `expression` cannot be evaluated.

`java.lang.NullPointerException` - If `expression` is `null`.

---

evaluate

```
java.lang.Object evaluate(java.lang.String expression,
                          InputSource source,
                          QName returnType)
                          throws XPathExpressionException
```

Evaluate an XPath expression in the context of the specified `InputSource` and return the result as the specified type.

This method builds a data model for the `InputSource` and calls `evaluate(String expression, Object item, QName returnType)` on the resulting document object.

See Evaluation of XPath Expressions for context item evaluation, variable, function and QName resolution and return type conversion.

If `returnType` is not one of the types defined in `XPathConstants`, then an `IllegalArgumentException` is thrown.

If `expression`, `source` or `returnType` is `null`, then a `NullPointerException` is thrown.

Parameters:

`expression` - The XPath expression.

`source` - The input source of the document to evaluate over.

`returnType` - The desired return type.

Returns:

The `Object` that encapsulates the result of evaluating the expression.

Throws:

`XPathExpressionException` - If expression cannot be evaluated.

`java.lang.IllegalArgumentException` - If `returnType` is not one of the types defined in `XPathConstants`.

`java.lang.NullPointerException` - If `expression`, `source` or `returnType` is `null`.

---

evaluate

```
java.lang.String evaluate(java.lang.String expression,
                          InputSource source)
                          throws XPathExpressionException
```

Evaluate an XPath expression in the context of the specified `InputSource` and return the result as a `String`.

This method calls `evaluate(String expression, InputSource source, QName returnType)` with a `returnType` of `XPathConstants.STRING`.

See Evaluation of XPath Expressions for context item evaluation, variable, function and QName resolution and return type conversion.

If `expression` or `source` is `null`, then a `NullPointerException` is thrown.

Parameters:

`expression` - The XPath expression.

`source` - The `InputSource` of the document to evaluate over.

Returns:

The `String` that is the result of evaluating the expression and converting the result to a

`String`

Throws:

`XPathExpressionException` - If expression cannot be evaluated.

`java.lang.NullPointerException` - If `expression` or `source` is `null`.

---

5 - 11 - 08

javax.xml.xpath

Class XPathConstants

```
java.lang.Object
  └ javax.xml.xpath.XPathConstants
```

```
public class XPathConstants
```

extends java.lang.Object

XPath constants.

Since:

1.5

Version:

$Revision: 1.3 $, $Date: 2005/11/03 19:34:18 $

Author:

Norman Walsh, Jeff Suttor

See Also:

XML Path Language (XPath) Version 1.0

Field Summary

static QName	**BOOLEAN** The XPath 1.0 boolean data type.
static java.lang.String	**DOM_OBJECT_MODEL** The URI for the DOM object model, "http://java.sun.com/jaxp/xpath/dom".
static QName	**NODE** The XPath 1.0 NodeSet data type.

XPathFactory.DEFAULT_OBJECT_MODEL_URI

static QName	**NODESET** The XPath 1.0 NodeSet data type.
static QName	**NUMBER** The XPath 1.0 number data type.
static QName	**STRING** The XPath 1.0 string data type.

Method Summary

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

NUMBER

```
public static final QName NUMBER
```

The XPath 1.0 number data type.

Maps to Java `Double`.

STRING

```
public static final QName STRING
```

The XPath 1.0 string data type.

Maps to Java `String`.

BOOLEAN

```
public static final QName BOOLEAN
```

The XPath 1.0 boolean data type.

Maps to Java `Boolean`.

NODESET

```
public static final QName NODESET
```

The XPath 1.0 NodeSet data type.

Maps to Java `NodeList`.

NODE

```
public static final QName NODE
```

The XPath 1.0 NodeSet data type.

Maps to Java `Node`.

DOM_OBJECT_MODEL

```
public static final java.lang.String DOM_OBJECT_MODEL
```

The URI for the DOM object model, "http://java.sun.com/jaxp/xpath/dom".

See Also:

Constant Field Values

javax.xml.xpath

Class XPathException

```
java.lang.Object
  └java.lang.Throwable
      └java.lang.Exception
          └javax.xml.xpath.XPathException
```

All Implemented Interfaces:

java.io.Serializable

Direct Known Subclasses:

XPathExpressionException, XPathFactoryConfigurationException

```
public class XPathException
```

extends java.lang.Exception

`XPathException` represents a generic XPath exception.

Since:

1.5

Version:

$Revision: 1.3 $, $Date: 2005/11/03 19:34:16 $

Author:

Norman Walsh, Jeff Suttor

See Also:

Serialized Form

Constructor Summary

XPathException(java.lang.String message) Constructs a new XPathException with the specified detail message.
XPathException(java.lang.Throwable cause) Constructs a new XPathException with the specified cause.

Method Summary

java.lang.Throwable	**getCause**() Get the cause of this XPathException.
void	**printStackTrace**() Print stack trace to System.err.
void	**printStackTrace**(java.io.PrintStream s) Print stack trace to specified PrintStream.
void	**printStackTrace**(java.io.PrintWriter s) Print stack trace to specified PrintWriter.

Methods inherited from class java.lang.Throwable
fillInStackTrace, getLocalizedMessage, getMessage, getStackTrace, initCause, setStackTrace, toString

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

XPathException

public **XPathException**(java.lang.String message)

Constructs a new XPathException with the specified detail message.

The `cause` is not initialized.

If `message` is `null`, then a `NullPointerException` is thrown.

Parameters:
`message` - The detail message.
Throws:
`java.lang.NullPointerException` - When `message` is `null`.

XPathException

```
public XPathException(java.lang.Throwable cause)
```

Constructs a new `XPathException` with the specified `cause`.

If `cause` is `null`, then a `NullPointerException` is thrown.

Parameters:
`cause` - The cause.
Throws:
`java.lang.NullPointerException` - if `cause` is `null`.

Method Detail

getCause

```
public java.lang.Throwable getCause()
```

Get the cause of this XPathException.

Overrides:
`getCause` in class `java.lang.Throwable`
Returns:
Cause of this XPathException.

---

printStackTrace

```
public void printStackTrace(java.io.PrintStream s)
```

Print stack trace to specified `PrintStream`.

Overrides:
`printStackTrace` in class `java.lang.Throwable`

Parameters:
`s` - Print stack trace to this `PrintStream`.

---

printStackTrace

```
public void printStackTrace()
```

Print stack trace to `System.err`.

Overrides:
`printStackTrace` in class `java.lang.Throwable`

---

printStackTrace

```
public void printStackTrace(java.io.PrintWriter s)
```

Print stack trace to specified `PrintWriter`.

Overrides:
`printStackTrace` in class `java.lang.Throwable`

Parameters:
`s` - Print stack trace to this `PrintWriter`.

---

5 - 11 - 08 7 - Jan -09 it has only 4 evaluate methods

javax.xml.xpath

Interface XPathExpression

public interface **XPathExpression**

`XPathExpression` provides access to compiled XPath expressions.

Evaluation of XPath Expressions.	
context	If a request is made to evaluate the expression in the absence of a context item, an empty document node will be used for the context. For the purposes of evaluating XPath expressions, a DocumentFragment is treated like a Document node.
variables	If the expression contains a variable reference, its value will be found through the `XPathVariableResolver`. An `XPathExpressionException` is raised if the variable resolver is undefined or the resolver returns `null` for the variable. The value of a variable must be immutable through the course of any single evaluation.
functions	If the expression contains a function reference, the function will be found through the `XPathFunctionResolver`. An `XPathExpressionException` is raised if the function resolver is undefined or the function resolver returns `null` for the function.
QNames	QNames in the expression are resolved against the XPath namespace context.
result	This result of evaluating an expression is converted to an instance of the desired return type. Valid return types are defined in `XPathConstants`. Conversion to the return type follows XPath conversion rules.

An XPath expression is not thread-safe and not reentrant. In other words, it is the application's responsibility to make sure that one `XPathExpression` object is not used from more than one thread at any given time, and while the `evaluate` method is invoked, applications may not recursively call

the `evaluate` method.

Since:

1.5

Version:

$Revision: 1.5 $, $Date: 2005/11/03 19:34:17 $

Author:

Norman Walsh, Jeff Suttor

See Also:

XML Path Language (XPath) Version 1.0, Expressions

Method Summary

`java.lang.String`	**`evaluate`**`(InputSource source)` Evaluate the compiled XPath expression in the context of the specified `InputSource` and return the result as a `String`.
`java.lang.Object`	**`evaluate`**`(InputSource source, QName returnType)` Evaluate the compiled XPath expression in the context of the specified `InputSource` and return the result as the specified type.
`java.lang.String`	**`evaluate`**`(java.lang.Object item)` Evaluate the compiled XPath expression in the specified context and return the result as a `String`.
`java.lang.Object`	**`evaluate`**`(java.lang.Object item, QName returnType)` Evaluate the compiled XPath expression in the specified context and return the result as the specified type.

this is DOM node

this is DOM node

Method Detail

evaluate

```
java.lang.Object evaluate(java.lang.Object item,
                          QName returnType)
                          throws XPathExpressionException
```

Evaluate the compiled XPath expression in the specified context and return the result as the specified type.

See Evaluation of XPath Expressions for context item evaluation, variable, function and QName resolution and return type conversion.

If `returnType` is not one of the types defined in `XPathConstants`, then an `IllegalArgumentException` is thrown.

If a `null` value is provided for `item`, an empty document will be used for the context. If `returnType` is `null`, then a `NullPointerException` is thrown.

Parameters:

`item` - The starting context (a node, for example).

`returnType` - The desired return type.

Returns:

The `Object` that is the result of evaluating the expression and converting the result to `returnType`

Throws:

`XPathExpressionException` - If the expression cannot be evaluated.

`java.lang.IllegalArgumentException` - If `returnType` is not one of the types defined in `XPathConstants`

`java.lang.NullPointerException` - If `returnType` is `null`.

---

evaluate

```
java.lang.String evaluate(java.lang.Object item)
                          throws XPathExpressionException
```

Evaluate the compiled XPath expression in the specified context and return the result as a `String`

This method calls `evaluate(Object item, QName returnType)` with a `returnType` of `XPathConstants.STRING`

See Evaluation of XPath Expressions for context item evaluation, variable, function and QName resolution and return type conversion.

If a `null` value is provided for `item`, an empty document will be used for the context.

Parameters:

`item` - The starting context (a node, for example).

Returns:

The `String` that is the result of evaluating the expression and converting the result to a `String`

Throws:

`XPathExpressionException` - If the expression cannot be evaluated.

---

evaluate

```
java.lang.Object evaluate(InputSource source,
                          QName returnType)
                          throws XPathExpressionException
```

Evaluate the compiled XPath expression in the context of the specified `InputSource` and return the result as the specified type.

This method builds a data model for the `InputSource` and calls `evaluate(Object item, QName returnType)` on the resulting document object.

See Evaluation of XPath Expressions for context item evaluation, variable, function and QName resolution and return type conversion.

If `returnType` is not one of the types defined in `XPathConstants`, then an `IllegalArgumentException` is thrown.

If `source` or `returnType` is `null`, then a `NullPointerException` is thrown.

Parameters:

`source` - The `InputSource` of the document to evaluate over.

`returnType` - The desired return type.

Returns:

The `Object` that is the result of evaluating the expression and converting the result to

`returnType`

Throws:

`XPathExpressionException` - If the expression cannot be evaluated.

`java.lang.IllegalArgumentException` - If `returnType` is not one of the types defined in `XPathConstants`.

`java.lang.NullPointerException` - If `source` or `returnType` is `null`.

---

evaluate

```
java.lang.String evaluate(InputSource source)
                          throws XPathExpressionException
```

Evaluate the compiled XPath expression in the context of the specified `InputSource` and return the result as a `String`.

This method calls `evaluate(InputSource source, QName returnType)` with a `returnType` of `XPathConstants.STRING`.

See Evaluation of XPath Expressions for context item evaluation, variable, function and QName resolution and return type conversion.

If `source` is `null`, then a `NullPointerException` is thrown.

Parameters:

`source` - The `InputSource` of the document to evaluate over.

Returns:

The `String` that is the result of evaluating the expression and converting the result to a `String`

Throws:

`XPathExpressionException` - If the expression cannot be evaluated.

`java.lang.NullPointerException` - If `source` is `null`

---

javax.xml.xpath

Class XPathExpressionException

```
java.lang.Object
  └java.lang.Throwable
      └java.lang.Exception
          └javax.xml.xpath.XPathException
              └javax.xml.xpath.XPathExpressionException
```

All Implemented Interfaces:

java.io.Serializable

Direct Known Subclasses:

XPathFunctionException

```
public class XPathExpressionException
```

extends XPathException

`XPathExpressionException` represents an error in an XPath expression.

Since:

1.5

Version:

$Revision: 1.3 $, $Date: 2005/11/03 19:34:17 $

Author:

Norman Walsh, Jeff Suttor

See Also:

Serialized Form

Constructor Summary

XPathExpressionException(java.lang.String message) Constructs a new XPathExpressionException with the specified detail message.
XPathExpressionException(java.lang.Throwable cause) Constructs a new XPathExpressionException with the specified cause.

Method Summary

Methods inherited from class javax.xml.xpath.XPathException
getCause, printStackTrace, printStackTrace, printStackTrace

Methods inherited from class java.lang.Throwable
fillInStackTrace, getLocalizedMessage, getMessage, getStackTrace, initCause, setStackTrace, toString

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

XPathExpressionException

```
public XPathExpressionException(java.lang.String message)
```

Constructs a new XPathExpressionException with the specified detail message.

The cause is not initialized.

If message is null, then a NullPointerException is thrown.

Parameters:

`message` - The detail message.

Throws:

`java.lang.NullPointerException` - When `message` is `null`.

XPathExpressionException

`public` **`XPathExpressionException`**`(java.lang.Throwable cause)`

Constructs a new `XPathExpressionException` with the specified `cause`.

If `cause` is `null`, then a `NullPointerException` is thrown.

Parameters:

`cause` - The cause.

Throws:

`java.lang.NullPointerException` - if `cause` is `null`.

---

5 - 11 - 08 6 - Jan -09

it has 9 methods

3 - new instance methods
2 - getter / setter Feature
1 - is check model support
1 - new XPath object
1 - setter for XPathFunctionResolver
1 - setter for XPathVariableResolver

javax.xml.xpath

Class XPathFactory

```
java.lang.Object
  └ javax.xml.xpath.XPathFactory
```

```
public abstract class XPathFactory
```

extends java.lang.Object

An `XPathFactory` instance can be used to create `XPath` objects.

~~See `newInstance(String uri)` for lookup mechanism.~~

The `XPathFactory` class is not thread-safe. In other words, it is the application's responsibility to ensure that at most one thread is using a `XPathFactory` object at any given moment. Implementations are encouraged to mark methods as `synchronized` to protect themselves from broken clients.

`XPathFactory` is not re-entrant. While one of the `newInstance` methods is being invoked, applications may not attempt to recursively invoke a `newInstance` method, even from the same thread.

~~Since:~~

~~1.5~~

~~Version:~~

~~$Revision: 1.5 $, $Date: 2006/05/19 01:08:43 $~~

~~Author:~~

~~Norman Walsh, Jeff Suttor~~

Field Summary

`static java.lang.String`	**`DEFAULT_OBJECT_MODEL_URI`** Default Object Model URI. ✓ DOM

static java.lang.String	**DEFAULT_PROPERTY_NAME** The default property name according to the JAXP spec.

Constructor Summary

protected	**XPathFactory**() ~~Protected constructor as~~ newInstance() ~~or~~ newInstance(String uri) ~~or~~ newInstance(String uri, String factoryClassName, ClassLoader classLoader) ~~should be used to create a new instance of an~~ XPathFactory.

Method Summary

abstract boolean	**getFeature**(java.lang.String name) Get the state of the named feature.
abstract boolean	**isObjectModelSupported**(java.lang.String objectModel) Is specified object model supported by this XPathFactory?
static XPathFactory	**newInstance**() Get a new XPathFactory instance using the default object model, DEFAULT_OBJECT_MODEL_URI, the W3C DOM.
static XPathFactory	**newInstance**(java.lang.String uri) Get a new XPathFactory instance using the specified object model.
static XPathFactory	**newInstance**(java.lang.String uri, java.lang.String factoryClassName, java.lang.ClassLoader classLoader) Obtain a new instance of a XPathFactory from a factory class name.
abstract XPath	**newXPath**() Return a new XPath using the underlying object model determined when the XPathFactory was instantiated.
abstract void	**setFeature**(java.lang.String name, boolean value) Set a feature for this XPathFactory and XPaths created by this factory.
abstract void	**setXPathFunctionResolver**(XPathFunctionResolver resolver) Establish a default function resolver.
abstract void	**setXPathVariableResolver**(XPathVariableResolver resolver) Establish a default variable resolver.

only setter not getter

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

DEFAULT_PROPERTY_NAME

public static final java.lang.String **DEFAULT_PROPERTY_NAME**

The default property name according to the JAXP spec.

it just system property key name of this factory

See Also:
Constant Field Values

DEFAULT_OBJECT_MODEL_URI

public static final java.lang.String **DEFAULT_OBJECT_MODEL_URI**

Default Object Model URI.

See Also:
Constant Field Values

Constructor Detail

XPathFactory

protected **XPathFactory**()

Protected constructor as newInstance() or newInstance(String uri) or newInstance (String uri, String factoryClassName, ClassLoader classLoader) should be used to create a new instance of an XPathFactory.

Method Detail

newInstance

public static final XPathFactory **newInstance**()

Get a new `XPathFactory` instance using the default object model, `DEFAULT_OBJECT_MODEL_URI`, the W3C DOM.

This method is functionally equivalent to:

```
newInstance(DEFAULT_OBJECT_MODEL_URI)
```

Since the implementation for the W3C DOM is always available, this method will never fail.

Returns:

Instance of an `XPathFactory`.

Throws:

`java.lang.RuntimeException` - When there is a failure in creating an `XPathFactory` for the default object model.

---

newInstance

```
public static final XPathFactory newInstance(java.lang.String uri)
                                      throws XPathFactoryConfigurationException
```

Get a new `XPathFactory` instance using the specified object model.

To find a `XPathFactory` object, this method looks the following places in the following order where "the class loader" refers to the context class loader:

1. If the system property `DEFAULT_PROPERTY_NAME` + ":uri" is present, where uri is the parameter to this method, then its value is read as a class name. The method will try to create a new instance of this class by using the class loader, and returns it if it is successfully created.
2. ${java.home}/lib/jaxp.properties is read and the value associated with the key being the system property above is looked for. If present, the value is processed just like above.
3. The class loader is asked for service provider provider-configuration files matching `javax.xml.xpath.XPathFactory` in the resource directory META-INF/services. See the JAR File Specification for file format and parsing rules. Each potential service provider is required to implement the method:

   `isObjectModelSupported(String objectModel)`

   The first service provider found in class loader order that supports the specified object model is returned.
4. Platform default `XPathFactory` is located in a platform specific way. There must be a platform

default XPathFactory for the W3C DOM, i.e. DEFAULT_OBJECT_MODEL_URI.

If everything fails, an XPathFactoryConfigurationException will be thrown.

Tip for Trouble-shooting:

See Properties.load(java.io.InputStream) for exactly how a property file is parsed. In particular, colons ':' need to be escaped in a property file, so make sure the URIs are properly escaped in it. For example:

```
http\://java.sun.com/jaxp/xpath/dom=org.acme.DomXPathFactory
```

Parameters:

uri - Identifies the underlying object model. The specification only defines the URI DEFAULT_OBJECT_MODEL_URI, http://java.sun.com/jaxp/xpath/dom for the W3C DOM, the org.w3c.dom package, and implementations are free to introduce other URIs for other object models.

Returns:

Instance of an XPathFactory.

Throws:

XPathFactoryConfigurationException - If the specified object model is unavailable.
java.lang.NullPointerException - If uri is null.
java.lang.IllegalArgumentException - If uri is null or uri.length() == 0.

---

newInstance

```
public static XPathFactory newInstance(java.lang.String uri,
                                       java.lang.String factoryClassName,
                                       java.lang.ClassLoader classLoader)
                                throws XPathFactoryConfigurationException
```

Obtain a new instance of a XPathFactory from a factory class name. XPathFactory is returned if specified factory class supports the specified object model. This function is useful when there are multiple providers in the classpath. It gives more control to the application as it can specify which provider should be loaded.

Tip for Trouble-shooting

Setting the jaxp.debug system property will cause this method to print a lot of debug messages to System.err about what it is doing and where it is looking at.

If you have problems try:

```
java -Djaxp.debug=1 YourProgram ....
```

Parameters:

`uri` - Identifies the underlying object model. The specification only defines the URI `DEFAULT_OBJECT_MODEL_URI`,`http://java.sun.com/jaxp/xpath/dom` for the W3C DOM, the org.w3c.dom package, and implementations are free to introduce other URIs for other object models.

`factoryClassName` - fully qualified factory class name that provides implementation of `javax.xml.xpath.XPathFactory`.

`classLoader` - `ClassLoader` used to load the factory class. If `null` current `Thread`'s context classLoader is used to load the factory class.

Returns:

New instance of a `XPathFactory`

Throws:

`XPathFactoryConfigurationException` - if `factoryClassName` is `null`, or the factory class cannot be loaded, instantiated or the factory class does not support the object model specified in the `uri` parameter.

`java.lang.NullPointerException` - If `uri` is `null`

`java.lang.IllegalArgumentException` - If `uri` is `null` or `uri.length() == 0`

Since:

1.6

See Also:

`newInstance()`, `newInstance(String uri)`

---

isObjectModelSupported

```
public abstract boolean isObjectModelSupported(java.lang.String objectModel)
```

Is specified object model supported by this `XPathFactory`?

Parameters:

`objectModel` - Specifies the object model which the returned `XPathFactory` will understand.

Returns:

`true` if `XPathFactory` supports `objectModel`, else `false`

Throws:

`java.lang.NullPointerException` - If `objectModel` is `null`

`java.lang.IllegalArgumentException` - If `objectModel.length() == 0`

---

setFeature

```
public abstract void setFeature(java.lang.String name,
                                boolean value)
                         throws XPathFactoryConfigurationException
```

Set a feature for this `XPathFactory` and `XPath`s created by this factory.

Feature names are fully qualified `URI`s. Implementations may define their own features. An `XPathFactoryConfigurationException` is thrown if this `XPathFactory` or the `XPath`s it creates cannot support the feature. It is possible for an `XPathFactory` to expose a feature value but be unable to change its state.

All implementations are required to support the `XMLConstants.FEATURE_SECURE_PROCESSING` feature. When the feature is `true`, any reference to an external function is an error. Under these conditions, the implementation must not call the `XPathFunctionResolver` and must throw an `XPathFunctionException`.

Parameters:

`name` - Feature name.

`value` - Is feature state `true` or `false`.

Throws:

`XPathFactoryConfigurationException` - if this `XPathFactory` or the `XPath`s it creates cannot support this feature.

`java.lang.NullPointerException` - if `name` is `null`.

---

getFeature

```
public abstract boolean getFeature(java.lang.String name)
                           throws XPathFactoryConfigurationException
```

Get the state of the named feature.

Feature names are fully qualified `URI`s. Implementations may define their own features. An `XPathFactoryConfigurationException` is thrown if this `XPathFactory` or the `XPath`s it creates cannot support the feature. It is possible for an `XPathFactory` to expose a feature value but be unable to change its state.

Parameters:

`name` - Feature name.

Returns:

State of the named feature.

Throws:

XPathFactoryConfigurationException - if this XPathFactory or the XPaths it creates cannot support this feature.

java.lang.NullPointerException - if name is null.

---

setXPathVariableResolver

no getter method for this

```
public abstract void setXPathVariableResolver(XPathVariableResolver resolver)
```

Establish a default variable resolver.

Any XPath objects constructed from this factory will use the specified resolver by default.

A NullPointerException is thrown if resolver is null.

Parameters:

resolver - Variable resolver.

Throws:

java.lang.NullPointerException - If resolver is null.

---

setXPathFunctionResolver

no getter method for this

```
public abstract void setXPathFunctionResolver(XPathFunctionResolver resolver)
```

Establish a default function resolver.

Any XPath objects constructed from this factory will use the specified resolver by default.

A NullPointerException is thrown if resolver is null.

Parameters:

resolver - XPath function resolver.

Throws:

java.lang.NullPointerException - If resolver is null.

---

newXPath

```
public abstract XPath newXPath()
```

Return a new `XPath` using the underlying object model determined when the `XPathFactory` was instantiated.

Returns:

New instance of an `XPath`

---

javax.xml.xpath

Class XPathFactoryConfigurationException

```
java.lang.Object
  └java.lang.Throwable
      └java.lang.Exception
          └javax.xml.xpath.XPathException
              └javax.xml.xpath.XPathFactoryConfigurationException
```

All Implemented Interfaces:

java.io.Serializable

```
public class XPathFactoryConfigurationException
```

extends XPathException

`XPathFactoryConfigurationException` represents a configuration error in a `XPathFactory` environment.

Since:

1.5

Version:

$Revision: 1.3 $, $Date: 2005/11/03 19:34:16 $

Author:

Norman Walsh, Jeff Suttor

See Also:

Serialized Form

Constructor Summary

XPathFactoryConfigurationException(java.lang.String message) Constructs a new XPathFactoryConfigurationException with the specified detail message.
XPathFactoryConfigurationException(java.lang.Throwable cause) Constructs a new XPathFactoryConfigurationException with the specified cause.

Method Summary

Methods inherited from class javax.xml.xpath.XPathException
getCause, printStackTrace, printStackTrace, printStackTrace

Methods inherited from class java.lang.Throwable
fillInStackTrace, getLocalizedMessage, getMessage, getStackTrace, initCause, setStackTrace, toString

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

XPathFactoryConfigurationException

public **XPathFactoryConfigurationException**(java.lang.String message)

Constructs a new XPathFactoryConfigurationException with the specified detail message.

The cause is not initialized.

If message is null, then a NullPointerException is thrown.

Parameters:

`message` - The detail message.

Throws:

`java.lang.NullPointerException` - When `message` is `null`.

XPathFactoryConfigurationException

`public` **`XPathFactoryConfigurationException`**`(java.lang.Throwable cause)`

Constructs a new `XPathFactoryConfigurationException` with the specified `cause`.

If `cause` is `null`, then a `NullPointerException` is thrown.

Parameters:

`cause` - The cause.

Throws:

`java.lang.NullPointerException` - if `cause` is `null`.

---

5 - 11 - 08

it has only one method

javax.xml.xpath

Interface XPathFunction

It is used only for Extension Function of XSL

```
public interface XPathFunction
```

`XPathFunction` provides access to XPath functions.

Functions are identified by QName and arity in XPath.

Since:

1.5

Version:

$Revision: 1.3 $, $Date: 2005/11/03 19:34:17 $

Author:

Norman Walsh, Jeff Suttor

Method Summary

java.lang.Object	**evaluate**(java.util.List args) Evaluate the function with the specified arguments.

Method Detail

evaluate

```
java.lang.Object evaluate(java.util.List args)
                          throws XPathFunctionException
```

Evaluate the function with the specified arguments.

To the greatest extent possible, side-effects should be avoided in the definition of extension functions. The implementation evaluating an XPath expression is under no obligation to call extension functions in any particular order or any particular number of times.

Parameters:

`args` - The arguments, `null` is a valid value.

Returns:

The result of evaluating the `XPath` function as an `Object`.

Throws:

`XPathFunctionException` - If `args` cannot be evaluated with this `XPath` function.

---

javax.xml.xpath

Class XPathFunctionException

```
java.lang.Object
  └java.lang.Throwable
      └java.lang.Exception
          └javax.xml.xpath.XPathException
              └javax.xml.xpath.XPathExpressionException
                  └javax.xml.xpath.XPathFunctionException
```

All Implemented Interfaces:

java.io.Serializable

```
public class XPathFunctionException
```

extends XPathExpressionException

`XPathFunctionException` represents an error with an XPath function.

Since:

1.5

Version:

$Revision: 1.3 $, $Date: 2005/11/03 19:34:18 $

Author:

Norman Walsh, Jeff Suttor

See Also:

Serialized Form

Constructor Summary

XPathFunctionException(java.lang.String message) Constructs a new XPathFunctionException with the specified detail message.
XPathFunctionException(java.lang.Throwable cause) Constructs a new XPathFunctionException with the specified cause.

Method Summary

Methods inherited from class javax.xml.xpath.XPathException
getCause, printStackTrace, printStackTrace, printStackTrace

Methods inherited from class java.lang.Throwable
fillInStackTrace, getLocalizedMessage, getMessage, getStackTrace, initCause, setStackTrace, toString

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

XPathFunctionException

```
public XPathFunctionException(java.lang.String message)
```

Constructs a new XPathFunctionException with the specified detail message.

The cause is not initialized.

If message is null, then a NullPointerException is thrown.

Parameters:

`message` - The detail message.

Throws:

`java.lang.NullPointerException` - When `message` is `null`.

XPathFunctionException

`public` **`XPathFunctionException`**`(java.lang.Throwable cause)`

Constructs a new `XPathFunctionException` with the specified `cause`.

If `cause` is `null`, then a `NullPointerException` is thrown.

Parameters:

`cause` - The cause.

Throws:

`java.lang.NullPointerException` - if `cause` is `null`.

---

5 - 11 - 08

6 - Jan -09

for 98% we dont write our own function. only built-in function itself is more than enough

Mostly i dont use this class

javax.xml.xpath

Interface XPathFunctionResolver

only one method

this will be used only for Extended function. that is our own XSL function

```
public interface XPathFunctionResolver
```

`XPathFunctionResolver` provides access to the set of user defined `XPathFunctions`.

XPath functions are resolved by name and arity. The resolver is not needed for XPath built-in functions and the resolver ***cannot*** be used to override those functions.

In particular, the resolver is only called for functions in an another namespace (functions with an explicit prefix). This means that you cannot use the `XPathFunctionResolver` to implement specifications like XML-Signature Syntax and Processing which extend the function library of XPath 1.0 in the same namespace. This is a consequence of the design of the resolver.

prefix like xml, xmlns

If you wish to implement additional built-in functions, you will have to extend the underlying implementation directly.

Since:

1.5

Version:

$Revision: 1.3 $, $Date: 2005/11/03 19:34:17 $

Author:

Norman Walsh, Jeff Suttor

See Also:

XML Path Language (XPath) Version 1.0, Core Function Library

Method Summary

XPathFunction	**resolveFunction**(QName functionName, int arity) Find a function in the set of available functions.

Where i will write function and store ??

Method Detail

resolveFunction

```
XPathFunction resolveFunction(QName functionName,
                              int arity)
```

Find a function in the set of available functions.

If `functionName` or `arity` is `null`, then a `NullPointerException` is thrown.

Parameters:

`functionName` - The function name.

`arity` - The number of arguments that the returned function must accept.

Returns:

The function or `null` if no function named `functionName` with `arity` arguments exists.

Throws:

`java.lang.NullPointerException` - If `functionName` or `arity` is `null`.

5 - 11 - 08

6 - Jan -08

javax.xml.xpath

it has only one method

Interface XPathVariableResolver

it is only for access
User defined Variable

`public interface XPathVariableResolver`

`XPathVariableResolver` provides access to the set of user defined XPath variables.

The `XPathVariableResolver` and the XPath evaluator must adhere to a contract that cannot be directly enforced by the API. Although variables may be mutable, that is, an application may wish to evaluate the same XPath expression more than once with different variable values, in the course of evaluating any single XPath expression, a variable's value ***must*** not change.

Since:

1.5

Version:

$Revision: 1.4 $, $Date: 2005/11/03 19:34:18 $

Author:

Norman Walsh, Jeff Suttor

Method Summary

java.lang.Object	**resolveVariable**(QName variableName) Find a variable in the set of available variables.

Method Detail

resolveVariable

`java.lang.Object` **`resolveVariable`**`(QName variableName)`

Find a variable in the set of available variables.

If `variableName` is `null`, then a `NullPointerException` is thrown.

Parameters:

`variableName` - The `QName` of the variable name.

so, need null check on returned object

Returns:

The variables value, or `null` if no variable named `variableName` exists. The value returned must be of a type appropriate for the underlying object model.

Throws:

`java.lang.NullPointerException` - If `variableName` is `null`.

Overview Package Class Use Tree Deprecated Index Help

PREV CLASS NEXT CLASS **FRAMES** **NO FRAMES** **All Classes**

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD
