## XPath 节点

在 XPath 中，有七种类型的节点：元素、属性、文本、命名空间、处理指令、注释以及文档（根）节点。XML 文档是被作为节点树来对待的。树的根被称为文档节点或者根节点。

请看下面这个 XML 文档：

<?xml version="1.0" encoding="ISO-8859-1"?>

<bookstore>

<book>

<title lang="en">Harry Potter</title>

<author>J K. Rowling</author>

<year>2005</year>

<price>29.99</price>

</book>

</bookstore>

上面的XML文档中的节点例子：

<bookstore> （文档节点）

<author>J K. Rowling</author> （元素节点）

lang="en" （属性节点）

### 基本值（或称原子值，Atomic value）

基本值是无父或无子的节点。

基本值的例子：

J K. Rowling

"en"

### 项目（Item）

项目是基本值或者节点。

### 命名空间

众所周知，XmlDocument可以进行XPath查询，但实际上这里所说的XPath查询仅限于没有命名空间（没有xmlns属性）的XML，一旦遇到有命名空间的XML，对应XPath查询都会无结果。

比如下面这个XML

<a xmlns="mgen.cnblogs.com">

<b>ccc</b>

</a>

XPath查询/a/b会返回null，而如果没有xmlns的话，会返回节点b。

<?xml version="1.0" encoding="utf-8"?>

<root xmlns="dotnet" xmlns:w="wpf">

<!-- xmlns: dotnet -->

<a>data in a</a>

<!-- xmlns: dotnet -->

<w:b>data in b</w:b>

<!-- xmlns: wpf -->

<c xmlns="silverlight">

<!-- xmlns: silverlight -->

<w:d>

<!-- xmlns: wpf -->

<e>data in e</e>

<!-- xmlns: silverlight -->

</w:d>

</c>

</root>

在XmlDocument中用XPath查询某一节点时，只要它的命名空间值不是空值，那么你必须给它一个前缀，用这个前缀代表这个节点的命名空间值！这些前缀是通过XmlNamespaceManager类添加的，使用时将XmlNamespaceManager传入SelectNodes或SelectSingleNode中即可。

<?xml version="1.0" encoding="utf-8"?>

<SOAP-ENV:Envelope

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:xsd="http://www.w3.org/2001/XMLSchema"

xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"

xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/">

<SOAP-ENV:Header>

<TransactionID xmlns="http://www.monternet.com/dsmp/schemas/">06070516230084</TransactionID>

</SOAP-ENV:Header>

<SOAP-ENV:Body>

<SyncOrderRelationReq xmlns="http://www.monternet.com/dsmp/schemas/">

<Version>1.5.0</Version>

<MsgType>SyncOrderRelationReq</MsgType>

<Send\_Address>

<DeviceType>0</DeviceType>

<DeviceID>0011</DeviceID>

</Send\_Address>

<Dest\_Address>

<DeviceType>400</DeviceType>

<DeviceID>0</DeviceID>

</Dest\_Address>

<FeeUser\_ID>

<UserIDType>1</UserIDType>

<MSISDN>13800138000</MSISDN>

<PseudoCode>

</PseudoCode>

</FeeUser\_ID>

<DestUser\_ID>

<UserIDType>1</UserIDType>

<MSISDN>13800138000</MSISDN>

<PseudoCode></PseudoCode>

</DestUser\_ID>

<LinkID>LinkID</LinkID>

<ActionID>1</ActionID>

<ActionReasonID>1</ActionReasonID>

<SPID>900000</SPID>

<SPServiceID>abcd</SPServiceID>

<AccessMode>3</AccessMode>

<FeatureStr></FeatureStr>

</SyncOrderRelationReq>

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>

string xml = "";

XmlDocument xmldoc = new XmlDocument();

xmldoc.LoadXml(xml);

XmlNamespaceManager xnm = new XmlNamespaceManager(xmldoc.NameTable);

xnm.AddNamespace("SOAP-ENV","http://schemas.xmlsoap.org/soap/envelope/");

xnm.AddNamespace("Mo","http://www.monternet.com/dsmp/schemas/");

xmldoc.SelectNodes("//SOAP-ENV:Header/Mo:TransationID",xnm);//注意，就算是用默认命名空间的节点，也要为命名空间定义一个名字，并使用这个名字。

## 语法

### 选取节点

|  |  |
| --- | --- |
| **表达式** | **描述** |
| nodename | 选取此节点的所有子节点。 |
| / | 从根节点选取。 |
| // | 从匹配选择的当前节点选择文档中的节点，而不考虑它们的位置。 |
| . | 选取当前节点。 |
| .. | 选取当前节点的父节点。 |
| @ | 选取属性。 |

### 谓语（Predicates）

谓语用来查找某个特定的节点或者包含某个指定的值的节点。

谓语被嵌在方括号中。

**实例**

在下面的表格中，我们列出了带有谓语的一些路径表达式，以及表达式的结果：

|  |  |
| --- | --- |
| **路径表达式** | **结果** |
| /bookstore/book[1] | 选取属于 bookstore 子元素的第一个 book 元素。 |
| /bookstore/book[last()] | 选取属于 bookstore 子元素的最后一个 book 元素。 |
| /bookstore/book[last()-1] | 选取属于 bookstore 子元素的倒数第二个 book 元素。 |
| /bookstore/book[position()<3] | 选取最前面的两个属于 bookstore 元素的子元素的 book 元素。 |
| //title[@lang] | 选取所有拥有名为 lang 的属性的 title 元素。 |
| //title[@lang='eng'] | 选取所有 title 元素，且这些元素拥有值为 eng 的 lang 属性。 |
| /bookstore/book[price>35.00] | 选取 bookstore 元素的所有 book 元素，且其中的 price 元素的值须大于 35.00。 |
| /bookstore/book[price>35.00]/title | 选取 bookstore 元素中的 book 元素的所有 title 元素，且其中的 price 元素的值须大于 35.00。 |

### 选取未知节点

XPath 通配符可用来选取未知的 XML 元素。

|  |  |
| --- | --- |
| **通配符** | **描述** |
| \* | 匹配任何元素节点。 |
| @\* | 匹配任何属性节点。 |
| node() | 匹配任何类型的节点。 |

### 选取若干路径

通过在路径表达式中使用“|”运算符，您可以选取若干个路径。

**实例**

在下面的表格中，我们列出了一些路径表达式，以及这些表达式的结果：

|  |  |
| --- | --- |
| **路径表达式** | **结果** |
| //book/title | //book/price | 选取 book 元素的所有 title 和 price 元素。 |
| //title | //price | 选取文档中的所有 title 和 price 元素。 |
| /bookstore/book/title | //price | 选取属于 bookstore 元素的 book 元素的所有 title 元素，以及文档中所有的 price 元素。 |