

What is XPath ?

Author: Mihai Rotaru

Date: 1 Dec 2011

XPath is a syntax used for addressing certain parts of an XML document. It can also be used to test whether a node matches a pattern (the pattern being the XPath) - this property is used by XSLT.

It is difficult to point out strengths and weaknesses, since XPath is a relatively simple technology, designed with a very specific purpose in mind. Over the years since it was introduced, XPath has proven it's value, and is a very important part of XML. The most common usage is in the context of XSLT, to select nodes onto which to apply formatting. It is obvious that some mechanism of selecting nodes is necessary in this context, not simply desirable; and XPath is the natural consequence of this requirement.

Assuming this well-formed XML document:

```
01 <?xml version="1.0"?>
02 <movies>
03 <movie>
04 <title>Hunger</title>
05 <rating>7.6</rating>
06 <price>15.00</price>
07 <year>2008</year>
08 </movie>
09 <movie>
10 <title>Year One</title>
11 <rating>5.0</rating>
12 <price>12.00</price>
13 <year>2011</year>
14 </movie>
15 <movie>
16 <title>Pulp Fiction</title>
17 <rating>9.0</rating>
18 <price>10.00</price>
19 <year>1994</year>
20 </movie>
21 <movie>
22 <title>The Godfather</title>
23 <rating>9.2</rating>
24 <price>15.00</price>
25 <year>1972</year>
26 </movie>
27 <movie>
28 <title>Apocalypto</title>
29 <rating>7.8</rating>
30 <price>17.00</price>
31 <year>2006</year>
32 </movie>
```

```
33 </movies>
```

XPath can be used to select a particular element. For example, to view the titles of all the movie elements, the following XPath expression can be used:

```
movies/movie/title
```

To view only the titles of movies released after year 2000:

```
/movies/movie[year > '2000']
```

To view the first two movies:

```
/movies/movie[position() < '3']
```

Some of the inconveniences of XPath include the fact that it has its own syntax that one needs to learn; it's a language within a language. XPath includes quite a number of functions (<http://www.w3.org/TR/xpath/#corelib>), which means that proficiency is not easy to achieve, and can become very difficult to read. Here's an example from the XPath specification:

```
/child::doc/child::chapter[position()=5]/  
child::section[position()=2]
```

It selects the second section of the fifth chapter of the doc document element.

On the other hand, the syntax is purposefully designed to be similar to the way file and folder paths are described on Unix systems, and having some familiarity with the Unix command line can ease the process of becoming comfortable with XPath. For example, the '.' has a conceptually similar significance - in Unix, it signifies the current directory, while in XPath the current element. In a similar vein, '..' can be used on the Unix command line as a shortcut for the parent of the current folder, while in XPath it means the parent of the current element. Hierarchy is described using the slash in both cases.