XML Path Language

Andy Clark

17 Apr 2002

Overview

- Syntax for selecting nodes in an XML document
- Location paths and expressions
  - Location paths similar to UNIX paths
    - e.g. /usr/local/bin
- Result of expression can be
  - Set of nodes (“node-set”)
  - Boolean
  - Number
  - String

Location Paths

- Comprised of “steps”
  - Relative to *context node*
- Each step has three parts
  - Axis
    - e.g. parent::, attribute::, child::, descendent::, etc…
  - Node test
    - e.g. foo, bar, html:body, etc…
  - Zero or more predicates
    - e.g. [1], [foo/bar], [text()=“Andy”], [not(position()=last())], etc…

Axes

- Full list
  - ancestor::, ancestor-or-self::
  - attribute::
  - child::, descendent::, descendent-or-self::
  - following::, following-sibling::
  - namespace::
  - parent::
  - preceding::, preceding-sibling::
  - self::

Node Tests

- Full list
  - Name test
    - e.g. *, *qname*, etc…
  - Node type
    - e.g. node(), text(), etc…
  - Processing instruction test
    - e.g. processing-instruction(“xml-stylesheet”)

Predicates

- Expressions
  - Location path
    - Union of location paths
  - Variable references
    - e.g. $name, etc…
  - String and number literals
    - e.g. “Andy”, 42, etc…
  - Functions
    - e.g. text(), position(), substring(), etc…

Functions

- Node-set functions
  - e.g. position(), last(), local-name(), etc…
- String functions
  - e.g. string(), contains(), substring(), etc…
- Boolean functions
  - e.g. boolean(), not(), etc…
- Number functions
  - e.g. number(), sum(), round(), etc…

Location Path Abbreviated Syntax

- Common location paths have short form

self::node()	.
parent::node()	..
attribute::bar	@bar
child::foo	foo
/descendent::foo	//foo
descendent::foo	.//foo

Basic Examples

Path	Selects
/	Root element
foo	Element "foo"
foo/bar	Child element "bar" of element "foo"
foo//bar	Element "bar" descendent of element "foo"
foo[bar]	Element "foo" contains child "bar"
@baz	Attribute "baz"
.	This node
..	Parent node
*	Any element
@*	Any attribute

More Complex Sample (1 of 2)

- Path:
  - /book/chapter[3]/section[subsection][2]
- Selects:
  - The second *section* that contains a *subsection* in the third *chapter* of the *book*
  - In pseudo-SQL:
    - FROM note *root* SELECT element “book”, element “chapter” WHERE (position = 3), element “section” WHERE (contains element “subsection” AND position = 2);

More Complex Sample (2 of 2)

- Path:
  - * [not(preceding-sibling::* [name() = name(current())])]
- Selects:
  - The set of children elements with unique names
  - In pseudo-SQL:
    - FROM node *current* SELECT element *any* WHERE (not(SELECT element *any* on axis preceding-sibling WHERE (element *name* = SELECT node *current name)));

Useful Links

- XPath 1.0 Specification
  - http://www.w3.org/TR/xpath
- XSLT 1.0 Specification
  - http://www.w3.org/TR/xslt

XML Path Language

Andy Clark