OASIS ebXML Registry

Proposal: Use of PATH Expressions

Category: Improvements to existing specifications

Date: September 28, 2001

Author: Farrukh Najmi, Nikola Stojanovic

Status of this Document

This document is a draft proposal whose purpose is to solicit additional input.

1 Abstract

There are two places in ebXML registry specifications where paths are used to identify an element or node in a classification scheme.

1. The first use is in representing an absolute path from the ClassificationScheme to a specific ClassificationNode for the purposes of identifying the ClassificationNode.
2. The second use is in describing a path used within ClassificationNode related filter queries to select one or more ClassificationNodes.

This document proposes to provide some missing details and clarity to the use of PATH syntax used in RIM 1.1 and RS 1.0 to identify or query for specific ClassificationNode.

2 Motivation

The following motivations drive this proposal:

1. Define detailed specification of a canonical path representation that is returned by the getPath method of ClassificationNode
2. Define detailed specification of path syntax that is valid in ClassificationNode queries

2.1 Assumptions

The following assumptions are made in this proposal:

1. Issues dealing with multiple co-operating registries are not considered. These issues are deferred to the Inter Registry Cooperation (IRC) team.

3 Canonical Path Representation

The ClassificationNode class defines a getPath method. This method returns an absolute path in a canonical representation that uniquely identifies the path leading from the ClassificationScheme to that ClassificationNode. The canonical path representation is similar to the familiar file system paths in many operating systems such as Unix.

The canonical path representation is defined by the following BNF grammar:

```
canonicalPath ::= '/' schemeURN nodePath
nodePath      ::=     '/' nodeCode
              |       '/' nodeCode ( nodePath )?
```

3.1 Scheme URN

The schemeURN in the canonical path syntax is the URN for the ClassifificationScheme as defined by a proposed new attribute name *urn* to be added to the ClassificationScheme class. It follows URN syntax as defined by http://www.ietf.org/rfc/rfc2141.txt

3.2 Example of Canonical Path Representation

The following canonical path represents what the getPath method would return for the ClassificationNode with code ‘United States’ in the sample Geography scheme in section 3.2.1.

```
/GeographyURN/NorthAmerica/UnitedStates
```

3.2.1 Sample Geography Scheme

Note that in the following examples, the ID attribute has been chosen for ease of memorization and is therefore not a valid URN or UUID.

```
<ClassificationScheme id='Geography id' name="Geography" urn=”GeographyURN”/>

<ClassificationNode id="North America id" parent="Geography" code=NorthAmerica" />
<ClassificationNode id="United States id" parent="North America" code="UnitedStates" />
```

```
<ClassificationNode id="Asia id" parent="Geography" code="Asia" />
<ClassificationNode id="Japan id" parent="Asia" code="Japan" />
<ClassificationNode id="Tokyo id" parent="Japan" code="Tokyo" />
```

4 Path Filter Expressions in Classification Queries

Path expressions are used in Classification related filter queries within the HasPathBranch element to find one or more ClassificationNodes in a ClassificationScheme. Such *Path Filter* expressions may also be used to match scheme elements in an external ClassificationScheme.

The path filter expressions are based on a very small sub-set of XPATH location path syntax. The path filter syntax extends the canonical path syntax grammar to include:

- Use of wildcard support using ‘*’ in place of nodeCode
- Use of ‘//’ syntax to denote any descendent of a node

It is defined by the following BNF grammar:

```
pathFilter     ::= '/' schemeURN  nodePath
nodePath       ::= slashes nodeCode
               |   slashes ‘*’
               |   slashes nodeCode ( nodePath )?
Slashes        ::= ‘/’ | ‘//’
```

4.1 Example of Path Filters

The following table lists various use cases and examples using the the sample Geography scheme in section 3.2.1.

Use Case	XPATH Expression	Description
Match all nodes in first level that have a specified value	/GeographyURN/NorthAmerica	Find all first level nodes with code 'NorthAmerica'
Match all nodes that have a specified value regardless of	/GeographyURN//Japan	Find all nodes with code "Japan”

level		
Match all nodes in the second level that have a specified value	/GeographyURN/*/Japan	Find all second level nodes with code 'Japan'
Match all nodes in the 3rd level that have a specified value	/GeographyURN /*/*/Tokyo	Find all third level nodes with code 'Tokyo'

5 Issues

We need to see if there is a better way to handle URNs.