R4X: Convenient XML Manipulation for R

Romain Francois - Mango Solutions
rfrancois@mango-solutions.com

August 9, 2008

Abstract

The R4X package enhances the `XML` package by providing a simple mechanism to create, read and manipulate XML structures from within R. The functionality of the package is based on the E4X (EcmaScript for XML) add-on to the javascript specifications.

R4X implements templating of XML structures based on the `brew` package and convenient extraction of XML content using an XPath-like syntax.

The XML format is used to transfer data accross many applications, specially web applications. The R4X package came from the need of a convenient tool to manipulate XML data when R is part of a bigger application. The presentation will describe key features of the R4X package, including examples of using the functionality to build a simple RSS reader and a tag cloud generated by using description of all current CRAN packages.

Keywords: XML, templates, data manipulation

Creation of XML

The generic `xml` function allows creation of XML structures as defined by the `XML` package from character vectors, connections. Users can write other `xml` methods to convert different formats to XML structures

```
R> x <- xml( '
    <test>
        <foo blah="1"/>
        <bar/>
    </test>' )
R> x

<test>
 <foo blah="1"/>
 <bar/>
</test>

R> class( x)

[1] "XMLNode"
```

The character string used by the `xml` function may contain R code embedded in curly brackets.

```
R> y <- "something"
R> x <- xml( '
    <test>
       <foo blah="{y}"/>
       <bar/>
    </test>' )
R> x

<test>
 <foo blah="something"/>
 <bar/>
</test>
```

Extracting XML content

The `[` and `[[` methods from the `XML` package are masked by R4X to provide an XPath-like mechanism for extracting and modifying content of an XML structure.

```
R> x <- xml( '
   <root>
     <child id="1">
       <subchild id = "sub1" >foo</subchild>
       <subchild id = "sub2" >bar</subchild>
     </child>
     <child id="2">
       <subchild id="a">blah</subchild>
       <subchild id="b">bob</subchild>
       <something id="c" />
     </child>
     <fruits>
        <fruit>banana</fruit>
        <fruit>mango</fruit>
     </fruits>
   </root>
   ' )
R> # extracting a single XMLNode
R> x[ "fruits" ]

<fruits>
 <fruit>banana</fruit>
 <fruit>mango</fruit>
</fruits>

R> # extracting the text content of each <subchild>
R> x[ "child/subchild/#" ]

child.subchild child.subchild child.subchild child.subchild
         "foo"          "bar"         "blah"          "bob"
```

```
R> # extracting the id attribute of each <child>
R> x[ "child/@id" ]

child child
  "1"   "2"

R> # extracting the <subchild> of the second <child>
R> x[ "child[2]/subchild" ]

$subchild
<subchild id="a">blah</subchild>

$subchild
<subchild id="b">bob</subchild>
```

Example - Tag Cloud Generator

Figure 1: Tag cloud generated from words used in descriptions of CRAN packages using the R4X package.