

Down Translating XML

Alberto Manuel Brandao Simões

ambs@cpan.org

Down translation is the process of converting XML (eXtensible Markup Language) data streams into another format, such as HTML or plain text. The Comprehensive Perl Archive Network (CPAN) contains lot of XML processing tools, and a new user can get lost in all of them. The `XML::DT` module, originally written by José João Almeida and currently maintained by José and I, is a handy tool for this transformation, even if it isn't the best tool for every job.

History

The `XML::DT` module stands for XML Down Translation, and was first developed by José João Almeida at the University of Minho, Portugal in 1999. He stole some features from other XML and SGML tools like OmniMark and Balise. It ran using `XML::Parser` as base XML parsing engine. Later, as one of José's students, I modified it to support a subset of XPATH and ported it to `XML::LibXML`, and so I became a co-developer of the module.

We ported it to `XML::LibXML` because `XML::Parser` development was stalled, libxml2 is faster, and libxml2 also provides ways of parsing HTML (non-XHTML even) files. When I install `XML::DT`, I can choose the expat (`XML::Parser`) or the libxml2 (`XML::LibXML`) backend.

A Simple Down Translation

Suppose I have my address book in XML, and I have the name, city, country, and email address for each person. The root element `book` comprises `address` elements which describe each person.

```
<?xml version="1.0"?>
<book>
  <address>
    <name>Alberto Simoes</name>
    <city>Braga</city>
```

```
<country>Portugal</country>
  <email>ambs@cpan</email>
</address>
<address>
  <name>Jose Joao Almeida</name>
  <city>Braga</city>
  <country>Portugal</country>
  <email>jjoao@cpan</email>
</address>
</book>
```

Now, I want to convert this to plain text. I start with `XML::DT's mkdtskel` (make down translation skeleton) script which creates an initial script for me. I supply the name of an XML file, and based on the structure it finds, it dumps a skeleton of a perl script that uses `XML::DT` to process it. For my address book it outputs this short script.

```
#!/usr/bin/perl
use XML::DT;
my $filename = shift;

%handler=(
  # '-outputenc' => 'ISO-8859-1',
  # '-default'   => sub{"<$q>$c</$q>"},

  'address' => sub{ "$q:$c" },
                # occurred 2 times
  'book'    => sub{ "$q:$c" },
                # occurred 1 times
  'city'    => sub{ "$q:$c" },
                # occurred 2 times
  'country' => sub{ "$q:$c" },
                # occurred 2 times
  'email'   => sub{ "$q:$c" },
                # occurred 2 times
  'name'    => sub{ "$q:$c" },
                # occurred 2 times
);
print dt($filename,%handler);
```

This script has two main parts: the hash where I specify the desired actions for each element I process, and the `XML::DT` function that uses the hash and an XML file to do the down translation.