
Subject: Dublin Core Metadata

Posted by [Susanne Wunsch railML](#) on Thu, 17 Sep 2009 19:08:31 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello,

railML now references Dublin Core Metadata.

See <<http://dublincore.org/>> for more information about it.

railML uses "dc.xsd", so you get up to 15 different metadata elements.

You can use it, like shown in the example "TT_CNL.xml":

```
<railml xmlns="http://www.railml.org/schemas/2007"
xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.railml.org/schemas/2007
../schema/railML.xsd">
<metadata>
  <dc:title>CNL night train example</dc:title>
  <dc:creator>Joachim Rubröder</dc:creator>
  <dc:source>manually build</dc:source>
</metadata>
...
</railml>
```

Feel free to try and comment the new railML code.

Download SVN revision [165] from assembla.com as Zip Archive
(ca. 900 KB):

<http://trac2.assembla.com/railML/changeset/165/trunk?old_path=%2F&old=165&format=zip>

Alternatively download folders "schema" and "examples" separately
(smaller size):

<<http://trac2.assembla.com/railML/browser/trunk?rev=165>>

Kind regards ...

Susanne

--

Susanne Wunsch

Schema Coordinator: railML.common

Subject: coding of dc:language

Posted by on Fri, 30 Mar 2012 17:53:30 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hallo,

I have added some examples on using Dublin Core Metadata Set to the Wiki pages. There was one response from Susanne concerning the item <dc:language>.

Generally, this item shall be used to code character set of the names (e.. g. station names a. s. o.) of the RailML file. This value is of importance in case the containing Unicode names have to be converted into a non-Unicode-string by the reading software.

Originally, I wanted it to contain the Codepage Number of the data <dc:language>1252</dc:language> ;(1252=ANSI - Lateinisch I) because, from my experience, one does need a Codepage Number do convert non-Unicode strings.

This did not enjoy Susanne who rather would prefer a coding like ISO 15924:
<dc:language>de-CH</dc:language>

The problem is that there is no 'conversion table' or something like that (as far as I know) to convert Codepage Numbers into ISO 15924 codes or vice versa. There is, unfortunately, no standardisation of Codepages at all. So if we do not allow the non-standardised Codepage Numbers we cannot tell the reading software how to convert the UTF-8 strings of a RailML file into non-Unicode strings. This leaves a reading software with the need to 'scan' the names for special characters and deduce a Codepage from this - a more empiric solution.

The problem with the ISO 15924 codes is not only that there is no 'conversion table'. It is also that typically a RailML file contains names of more than one language, e. g. some foreign station names also. This is normally no problem because one Codepage normally allows languages of neighboring countries. Our 'middle-European' Codepage (1250) allows German

what should we write into <dc:language> if a RailML file contains all of these three and the writing programme only know that it is CodePage 1250?

Anyway, it is not a big problem because it only applies to non-Unicode software and there should be not much non-Unicode software nowadays. It's only that we do not know...

So, from my opinion we have two possible solutions:

- a) either to skip this <dc:language> at all and delete it from all examples
- b) or still to allow and recommend a Codepage number (!) there because it

costs nothing, may help someone, and there is no other need for this element.

It does not make sense to code it with ISO 15924 since, as I did explain, there is normally not `_one_` source language for all the RailML file.

@Susanne: If nobody answers this 'post' in a near future you can tell me at any time to delete this `<dc:language>` from the examples without further objection from me. I leave it up to you, doing nothing more from my side..

With best regards,
Dirk.

Subject: Re: coding of dc:language
Posted by [Joerg von Lingen](#) on Sun, 01 Apr 2012 04:18:31 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Dirk,

I would not delete the `<dc:language>` in any case. If there is a need for code page information this shall be additional.

The original thought about `<dc:language>` was to identify the language used for that name, especially when you have a station like Bautzen/Budyšin with several names in different languages.

--

Best regards,
Joerg v. Lingen

On 30.03.2012 19:53, Dirk Bräuer wrote:

- > Hallo,
- >
- > I have added some examples on using Dublin Core Metadata Set to the Wiki pages.
- > There was one response from Susanne concerning the item `<dc:language>`.
- >
- > Generally, this item shall be used to code character set of the names (e. g.
- > station names a. s. o.) of the RailML file. This value is of importance in case
- > the containing Unicode names have to be converted into a non-Unicode-string by
- > the reading software.
- >
- > Originally, I wanted it to contain the Codepage Number of the data
- > `<dc:language>1252</dc:language>` ;(1252=ANSI - Lateinisch I)
- > because, from my experience, one does need a Codepage Number do convert
- > non-Unicode strings.
- >
- > This did not enjoy Susanne who rather would prefer a coding like ISO 15924:

> <dc:language>de-CH</dc:language>
>
> The problem is that there is no 'conversion table' or something like that (as
> far as I know) to convert Codepage Numbers into ISO 15924 codes or vice versa.
> There is, unfortunately, no standardisation of Codepages at all. So if we do not
> allow the non-standardised Codepage Numbers we cannot tell the reading software
> how to convert the UTF-8 strings of a RailML file into non-Unicode strings. This
> leaves a reading software with the need to 'scan' the names for special
> characters and deduce a Codepage from this - a more empiric solution.
>
> The problem with the ISO 15924 codes is not only that there is no 'conversion
> table'. It is also that typically a RailML file contains names of more than one
> language, e. g. some foreign station names also. This is normally no problem
> because one Codepage normally allows languages of neighboring countries. Our

> and Sorbian/Polish 'striked-out L's'. But what should we write into
> <dc:language> if a RailML file contains all of these three and the writing
> programme only know that it is CodePage 1250?
>
> Anyway, it is not a big problem because it only applies to non-Unicode software
> and there should be not much non-Unicode software nowadays. It's only that we do
> not know...
>
> So, from my opinion we have two possible solutions:
> a) either to skip this <dc:language> at all and delete it from all examples
> b) or still to allow and recommend a Codepage number (!) there because it costs
> nothing, may help someone, and there is no other need for this element.
>
> It does not make sense to code it with ISO 15924 since, as I did explain, there
> is normally not one source language for all the RailML file.
>
> @Susanne: If nobody answers this 'post' in a near future you can tell me at any
> time to delete this <dc:language> from the examples without further objection
> from me. I leave it up to you, doing nothing more from my side.
>
> With best regards,
> Dirk.

Subject: Re: coding of dc:language
Posted by _____ on Mon, 02 Apr 2012 10:26:54 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Jörg,

> I would not delete the <dc:language> in any case

Yes, of course, I cannot delete it since it comes with Dublin Core

Metadata Set. My recommendation to Susanne was

> to delete this <dc:language> from the examples

with the pronunciation at "from the examples" ;-)

- > The original thought about <dc:language> was to identify the language
- > used for
- > that name, especially when you have a station like Bautzen/Budyšin with
- > several
- > names in different languages.

I agree that it is (still) indented to identify the language of names, remarks a.s.o. which do not have a individual 'xml:lang' attribute such as the general 'name' of a station. An additional name (like Budyšin) already has its 'xml:lang' filed.

But I think that we should not add a 'xml:lang' attribute to all fields which may contain a kind of plain text (of any language). This would mean to add 'xml:lang' to all fields defined with type 'string' - which would expand both the XSD and RailML files in an unnecessary way. This is the point where from my opinion <dc:language> comes into play...

Best regards,
Dirk.
