

---

Subject: Re: Identification in the XML list files and its references

Posted by on Mon, 10 Dec 2012 15:49:08 GMT

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

---

Dear Susanne and all others,

- > The separate XML list files should be an easier to maintain replacement
- > for schema-internal enumeration lists.

+1

I think it is a compromise between a free string and an XML enumeration type.

- > \* Should a railML file be meaningful without this list file?
- >
- > That would mean to refer to the /meaningful/ 'code' value, that is not
- > unique.

Between the contradictory aspects of "uniqueness" and "easy to read at text file level", I would prefer a technical uniqueness in case of doubt. This means: Rather refer to /id/ than to /code/ in favour of consequence.

If anybody wants to create an "easy to read" XML file he may use a meaningful /id/ for test purposes.

- > \* Should a railML file be meaningful only with knowledge of the list
- > file, only in cases, where its attributes are used?
- >
- > That would mean to refer to the \_\_unique\_\_ but not /meaningful/ 'id'
- > value.

Again the same answer from my side.

- > \* Should both possibilities be provided? If the list file is present, it
  - > may be looked up for further details, if not, the value is
  - > /meaningful/ anyway.
  - >
  - > That would mean to refer to both values.
- > <designator register="RL100" registerRef="registers.xml#d1e51" entry="..."/>

As we have already agreed to avoid redundancies as far as possible, this should not be an option. What should a reading software do if there are to contradictory references at one element? I. e. an <ocp> refers to a <register> using a /registerRef/ but the attribute /register/ of the <ocp> is different than the appropriate attribute in the additional XML file?

I think RailML is more a technical data exchange format than a text format for intuitive reading. So, the readability has to step back in cases of doubt. There is still the possibility to create "easy readable" files using meaningful /id/s. This possibility is good, but it should be left to explicit test cases. For all-day work, interoperability should be the main aim.

With best regards,  
Dirk.

---