

---

Subject: Re: the use of @dir in railML.

Posted by on Mon, 09 Mar 2020 10:39:06 GMT

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

---

Dear Thomas, Christian and all,

sorry, Thomas, for not answering your question from 24.10.2018 so far. Somehow, I lost connection...

Concerning <speedChange>s, I want to record as a summary from our side:

- I understand Thomas' suggestion to interpret "simply a property of the infrastructure, regardless of the direction of traffic". I have full sympathy for this kind of usage and would normally not plead against it.

- However, unfortunately the usage was different in the last decade or more: Our (FBS) railML export interface clearly produces speedChanges in the following way:

> Die Geschwindigkeitswechsel sind entsprechend der Gültigkeitsrichtung ihres Geschwindigkeitsprofils angegeben. Im Falle eines Geschwindigkeitsprofils, das für beide Fahrrichtungen gültig ist, sind die Informationen redundant doppelt vorhanden – je einmal pro Fahrrichtung.

(The <speedChange>s are exported depending on the validity direction of the <speedProfile>. In case of a <speedProfile> valid for both directions, the <speedChange>s are exported redundantly doubled once per driving direction.)

> Je nach Gültigkeitsrichtung des Geschwindigkeitsprofils ist an der Position des Streckenanfangs (<trackBegin>.pos) bzw. Streckenendes (<trackEnd>.pos) immer ein Geschwindigkeitswechsel enthalten.

(Depending on the validity direction of the <speedProfile>, there is always a <speedChange> at the position of <trackBegin>.pos or <trackEnd>.pos.)

- The reason for this (possibly unexpected) behaviour is, as far as I remember, an objection of Susanne Wunsch in the time of publishing and certifying our FBS-RailML2-Export (Susanne was common coordinator in that time). You can still read Susannes opinion in [1].

I want to repeat that I have full sympathy for Thomas' kind of interpretation, which is different than our current usage. But a specification by railML now may not lead to existing valid railML files becoming invalid at once. So, unfortunately, the only solution I see is either to clarify the "new" interpretation as a future recommendation only, clearly allowing the "old" interpretation for backward compatibility in existing use cases, or to limit this discussion on railML 3.x.

However, the "advantage" of the old interpretation is that it keeps to the "rule": railML infrastructure orientates on what you can touch in the outside world. There is a speed change sign (post) at the beginning of each speed change depending on the driving direction, not depending on the definition direction. There are two speed change signs in case of tracks which are used in

both directions. Therefore, the old interpretation may be more complicated and redundant, but at least it is closer to the outside world and therefore may be easier to understand by non-informatics, beginners etc.

Best regards,  
Dirk.

[1] [https://www.railml.org/forum/index.php?t=msg&th=120&goto=276&#msg\\_276](https://www.railml.org/forum/index.php?t=msg&th=120&goto=276&#msg_276)

---