

---

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

Posted by [Thomas Nygreen](#) on Thu, 27 Feb 2020 18:08:27 GMT

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

---

Dear Christian,

Dear all,

It looks like we are finally near to closing this issue. I just have a few final questions/comments:

<lock> was introduced after I did my review of all elements with @dir. I suggest that we deprecate @dir also for <lock>.

We agree to deprecate @dir on <trackCircuitBorder>. But what about <trainDetector>? Are there axle counters that only detect axles going in one direction? I have a feeling that we should treat <trackCircuitBorder> and <trainDetector> the same way.

As I understand Christian's proposals, there will no longer be three different sets of values for @dir. All occurrences will allow "up", "down" and "both". I think that is a good solution, as it also allows equal interpretation across elements of a missing @dir as unknown. I assume we will have to keep three different types in the XSD, where tLaxDirection will have two deprecated values ("unknown" and "none"), [the badly named] tDelimitedDirection will have one new value ("both") and one deprecated ("unknown") and tStrictDirection will have one new value ("both").

And then, finally, there are the <\*Change> elements.

Den 13.01.2020 15:09, skrev Christian Rahmig:

>>> By standard, the change elements' orientation (not their  
>>> application direction!) shall be always in direction of track  
>>> orientation (from trackBegin towards trackEnd).

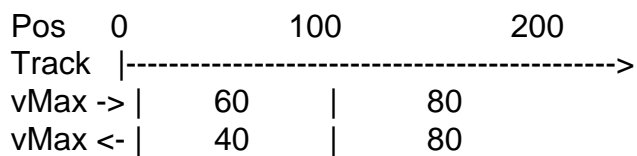
Den 26.02.2020 11:05, skrev Christian Rahmig:

> A change element always contains a change of a feature from  
> an old value to a new value. E.g. referring to an  
> <electrificationChange> there is a state of electrification  
> before the change as well as after the change. By setting  
> the "change element orientation" I want to specify that the  
> old value applies for the track until <\*Change>@pos and the  
> new value (as provided by the change element) applies from  
> <\*Change>@pos ongoing.

Just to make sure we are thinking alike here:

For the <\*Change> elements where we deprecate @dir (axleWeightChange, clearanceGaugeChange, electrificationChange, gaugeChange, ownerChange, powerTransmissionChange and radiusChange) this means that the value is simply a property of the infrastructure, regardless of the direction of traffic. The new value given in the <\*Change> element describes the infrastructure from the point it is placed (@pos) towards the track end. This is as documented in the wiki for <radiusChange>.

For the <\*Change> elements where we keep @dir (gradientChange, operationModeChange, speedChange, trainProtectionChange and trainRadioChange), @dir only describes which direction of traffic the new value applies to. The value is applied to the infrastructure in the same way as for the other <\*Change> elements. To reiterate a previous example, this means that



would be exported as:

```
<speedChange pos="0" dir="up" vMax="60"/>  
<speedChange pos="0" dir="down" vMax="40"/>  
<speedChange pos="100" dir="both" vMax="80"/>
```

Best regards,  
Thomas

---