

---

Subject: Re: [railML3] Train number/identification systems

Posted by [Milan Wölke](#) on Mon, 14 Feb 2022 10:56:21 GMT

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

---

Hallo zusammen,

bei unserer letzten Entwickler Telco haben wir dieses Thema diskutiert. Dabei haben wir entschieden, dass wir die verschiedenen Identitätstypen der Taf Tap Tsi vollständig in das entsprechende railML3 Enum mit aufnehmen, also TR (train), PR (path request), PA (path), RO (route) und CR (case reference).

Was ScopeSecondary, ScopeSecondaryStart, ScopeSecondaryEnd und ScopeSecondaryInner betrifft, haben wir entschieden, dass wir diese zunächst nicht mit in railML3 aufnehmen wollen. Der Hintergrund hier ist, dass dieses Zugnummernschema nur bei sehr wenigen Bahnen im Einsatz ist und der größte dieser bereits angekündigt hat, sich von dieser Praxis zu verabschieden. Insofern wollen wir kein Konzept in den neuen railML 3 Standard aufnehmen, dass in näherer Zukunft sowieso für obsolet erklärt wird.

Für jene, die noch auf eine Nutzung angewiesen sind, ist das Enum der Zugnummerntypen in railML 3 aber erweiterbar modelliert, so dass auch andere als die vordefinierten verwendet werden können.

Sofern diesbezüglich gegensätzliche Ansichten existieren, dann bitte ich darum diese hier zu äußern. Noch wurde railML 3.2 nicht veröffentlicht und kann daher noch entsprechend einfach angepasst werden.

-----

We discussed this topic at our last developer telco. We decided to include the different identity types of the Taf Tap Tsi in the corresponding railML3 enum, i.e. TR (train), PR (path request), PA (path), RO (route) and CR (case reference).

As far as ScopeSecondary, ScopeSecondaryStart, ScopeSecondaryEnd and ScopeSecondaryInner are concerned, we have decided not to include them in railML3 for the time being. The background here is that this train numbering scheme is only used by a very few railways and the largest of these has already announced that it will abandon this practice. We therefore do not want to include a concept in the new railML 3 standard that will be declared obsolete in the near future anyway.

However, for those who still need to use it, the enum of train number types in railML 3 is modelled in an extensible way, so that others than the predefined ones can be used.

If there are opposing views on this, please express them here. RailML 3.2 has not yet been published and can therefore still be easily adapted accordingly.

Best regards, Milan

---