

---

Subject: [railML3] Validities without bitMask

Posted by [Milan Wölke](#) on Thu, 24 Feb 2022 16:54:58 GMT

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

---

Hi all,

recently we the timetable developer group discussed if beside the already existing ways of specifying validities in railML3 a third option should be added. Since I assume you are not too familiar with the current development state of railML3 timetable, let me shortly describe the two existing option:

- 1.) The first option is very similar to the one well known from railML2; use a bitmask to specify if a train runs or not and use some optional week patterns to specify how that bitMask was created in case someone needs to work with the system behind that.
- 2.) The other option is to specify a generic week pattern based on a 7 char-bitMask. This is intended for strategic planning when its not yet clear which exact week is described.

The newly requested 3rd option was to allow a list of such week patterns (like in the second option) each with a start and end date. Basically this means that its like the first option, just without the bitMask included.

For better understanding of the suggested extension please also refer to the attached image: [https://forum.railml.org/userfiles/2022-02-24\\_railml\\_railml3-compositeweekpatternvalidity.png](https://forum.railml.org/userfiles/2022-02-24_railml_railml3-compositeweekpatternvalidity.png)

The developer group decided not to add this 3rd option as it is easy enough to calculate the necessary resulting bitmask from a week based pattern. Since there was discussion on this point and one easily could also take the opposite position and argue that if it is so easy it does not need to be transferred with railML I am posting this, to see what the community thinks of this issue.

Please let me know what is your view on the matter. Should we have added a 3rd option? Or should we have replaced the first existing option with that new approach?

Thanks in advance for your feedback.

-----

vor kurzem haben wir in der timetable developer group darüber diskutiert, ob neben den bereits existierenden Möglichkeiten zur Angabe von Gültigkeiten in railML3 eine dritte Option hinzugefügt werden sollte. Da ich davon ausgehe, dass Sie mit dem aktuellen Entwicklungsstand von railML3-Fahrplan nicht allzu vertraut sind, möchte ich die beiden bestehenden Möglichkeiten kurz beschreiben:

- 1.) Die erste Option ist der aus railML2 bekannten sehr ähnlich; verwenden Sie eine Bitmaske, um anzugeben, ob ein Zug fährt oder nicht, und verwenden Sie einige optionale Wochenmuster, um anzugeben, wie diese Bitmaske erstellt wurde, falls jemand mit dem System dahinter arbeiten muss.
- 2.) Die andere Möglichkeit besteht darin, ein allgemeines Wochenmuster auf der Grundlage einer 7-Zeichen-Bitmaske anzugeben. Diese Option ist für die strategische Planung gedacht, wenn noch nicht klar ist, welche Woche genau beschrieben wird.

Die neu angefragte 3. Option war, eine Liste solcher Wochenmuster (wie in der zweiten Option) jeweils mit einem Start- und Enddatum zuzulassen. Im Grunde bedeutet dies, dass es wie die erste Option ist, nur ohne die bitMaske.

Zum besseren Verständnis der vorgeschlagenen Erweiterung können Sie auch das beigefügte Bild betrachten: [https://forum.railml.org/userfiles/2022-02-24\\_railml\\_railml3-compositeweekpatternvalidity.png](https://forum.railml.org/userfiles/2022-02-24_railml_railml3-compositeweekpatternvalidity.png)

Die Entwicklergruppe hat beschlossen, diese dritte Option nicht hinzuzufügen, da es problemlos möglich ist, die erforderliche resultierende Bitmaske aus einem wochenbasierten Muster zu berechnen. Da es zu diesem Punkt eine Diskussion gab und man auch leicht die gegenteilige Position einnehmen könnte und argumentieren könnte, dass, wenn es so einfach ist, es nicht mit railML übertragen werden muss, poste ich dies, um zu sehen, was die Community über dieses Thema denkt.

Bitte lassen Sie mich wissen, was Sie von dieser Sache halten. Hätten wir eine 3. Option hinzufügen sollen? Oder hätten wir die erste bestehende Option durch diesen neuen Ansatz ersetzen sollen?

Vielen Dank im Voraus für Ihr Feedback.

Best regards, Milan

---