

---

Subject: [railML3] Proposal for removing "any" elements  
Posted by [christian.rahmig](#) on Tue, 22 Mar 2022 08:13:26 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Dear railML IS community,

At our last conference in Gothenburg, Sweden, our Common Coordinator raised the issue of changing the way extensions for railML can be created in railML3. Thomas presented a new approach there, which is, however, mutually exclusive with the current approach. He has created a forum post on this, which unfortunately has not received much attention so far:

<https://www.railml.org/forum/index.php?t=msg&th=638&start=0&>

The railML.org coordinators think it is a good idea to use this new technique, as it allows, among other things, to validate documents with custom extensions of the respective railML interfaces. However, since such a change will mean that existing extensions will no longer work with a new railML version, we need your feedback to check whether the advantages we see in the new way are convincing to you as well. Please let us know what you think under the thread above.

Background:

Previously, the common practice was to provide extension points in the official railML schema. This allowed an XML that contained non-railML tags at such a point to still be considered valid railML. The disadvantage of this approach is that it is not possible to specify through the extension schema itself where these non-railML tags are allowed and where they are not. An extension that is intended to record opening hours for a station building could thus also be used to record opening hours for a circulation or for a train number. From a technical point of view, this does not make sense. With the extension points it is not possible to restrict this. With the newly proposed approach (see forum post) this would no longer be a problem. In addition, code generation tools could also be used to implement code for importing and exporting railML more efficiently.

We therefore propose to replace the extension points with railML 3.2 with the new inheritance method. Please let us know at the link above whether there are any professional/technical arguments against this from your point of view.

Best regards  
Christian

---