

---

Subject: Passing arbitrary attributes / use of "any" fields

Posted by [Thomas Langkamm](#) on Tue, 20 Jul 2021 09:29:49 GMT

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

---

I understand that the use of "any" attributes is somewhat discouraged in railML (<https://wiki2.railml.org/wiki/Dev:usingAny>), for good reasons. However, there is a common use case where we need additional attributes on railML objects, and I would like to see a standardized way to handle them: Pass references to external objects.

Now I'll admit that it's a bit of a "debugging" feature, but I think it is essential if we want to switch from proprietary formats to railML.

My benchmark use case here is that we want to generate a railML file from sources that has a different structures than railML. A real life example: We are currently evaluating a project where we have data (mostly IS, some IL) from 3 different importable sources and we want to create a railML file out of that. More precisely, our customer has 3 different software programs that they use, each with varying details of the infrastructure. The general idea is that we build a converter, consolidating the data into a railML file that can then be edited with an editor like Rail-AiD. However, there will be inevitably inconsistencies in the data as the 3 sources occasionally have contradicting data. So what I would like to do is pass references to the source objects, which allows the "debugging" of these inconsistencies. This allows the end user, if she/he edits the railML file in the editor, to find the original objects in the other software.

For example, we have 2 sources for track sections (typically bounded by balises, isolations, signals or switches) that are not entirely consistent. From a software standpoint this is already difficult to translate into railML, as some of these will become tvd sections while other track sections will be joined into one netElement. (For example, 2 consecutive track sections split by a signal would become one netElement with a signal in the middle.) What I would like to do in this case is to annotate all netElements in the railML file with a list of track sections from either source, so that the software developer and later the user can verify how the source objects were mapped to railML netElements.

In short, I'd like a standardized way to add a list of external references to arbitrary railML objects. Is there any existing concept to do this?