
Subject: Re: [railML3] Handling changes between minor versions
Posted by on Thu, 13 Aug 2020 13:41:59 GMT

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

Hello,

> The coordinators are proposing the > following three alternatives:> > Follow the path of railML 2.x in

railML 3.x: downward> compatibility will be guaranteed. Changes may lead to> elements or attributes being deprecated, but not removed.> Changes may lead to elements or attributes being> deprecated. Elements and attributes that are deprecated in> one minor version will be removed in the following minor> version. Compatibility is only guaranteed between one minor> version and the next.> Changes may lead to elements or attributes being removed in> a new minor version, without first being deprecated. No> compatibility guaranteed.

We would prefer option 3 (give up compatibility between minor versions). The current compatibility rule often stands in the way of further development of the schema and, in our view, has little advantage when implementing the interface software.

Furthermore, backward compatibility does not only apply to the removal of attributes / elements: Adding a new attribute can also change the semantics of other existing attributes, for example, by semantically overwriting or negating the contents of other attributes. For this reason, it is not always predictable anyway whether a change will break downward compatibility or not.

However, different revisions of an (already released) version should remain backwards compatible.

Best regards

Christian Rößiger

--

iRFP e. K. · Institut für Regional- und Fernverkehrsplanung

Hochschulstr. 45, 01069 Dresden

Tel. +49 351 4706819 · Fax. +49 351 4768190 · www.irfp.de

Registergericht: Amtsgericht Dresden, HRA 9347
