

Dear Andreas,

Thank you for teaching me how to recognize the point of Join / Split.
Your valid condition description helps me very much. :-)

> Sent per mail.

Thank you, too.

There isn't any trainPartSequence that sequence isn't "1" in your sample railML file, and every point of Join / Split in it seems to be either start ocp or end ocp of a train.

If there is a trainPartSequence that sequence isn't "1", it means there is either Join / Split even in the course of the train as

<http://wiki.railml.org/index.php?title=TT:trainCouplingAndSh> aring examples and we can find out the point with corresponding ocpTTs. If you send me another railML example file that has some Join / Split even in the course of some trains, I appreciate it. ;-)

Best regards,
Utah (Yutaka Manchu)

Andreas Tanner wrote:

>

> Am 12.09.2014 07:14, schrieb Yutaka Manchu:

>>

>> About Split / Join, there're examples that don't have data under

>> /trainPartRef/s;

>> <http://wiki.railml.org/index.php?title=TT:trainCouplingAndSh> aring

>>

>> May I confirm you if where the Split / Join occurs (i.e.

>> "Lille", "SanAntonio" in above examples) are known from the /ocpRef/ of

>> <ocpTT>s of <trainPart> referred from the /trainPartRef/ or not?

>

> yes, the location of the split/join is at the first / last ocpTT of the

> referenced trainParts.

> The following are assumed:

> - the sequences of ocpTT of all trainParts within one trainPartSequence

> coincide (at least when the operation periods are overlapping)

> - the sequence of ocpTT within a train, that is, along the

> trainPartSequences, form a valid train path. This means that there are

> no location breaks and the times are ascending.

>

>> If you have a "full railML timetable file" that contains Join / Split,

>> would you mind sending it for me?

> Sent per mail.

>
> Best, Andreas.
>
>

--
-----== posted via PHP Headliner ==-----
