

---

Subject: Proposal: New optional attributes on type block  
Posted by [horst.naujoks](#) on Tue, 30 Apr 2013 12:43:46 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hi all,

I like to propose the following enhancement for an upcoming railML release (not necessarily 2.2):

The physical location where a block begins and ends could be implicitly determined by inspecting its contained blockpart sequences and block parts respectively. From first found block part the attribute startOcpRef could be used and analogous from the last one the endOcpRef attribute.

But this approach requires that all the block parts are present.

We are using railML in a scenario, where present blocks in two different modes: overview or details.

In the overview mode we are delivering only blocks without containing details (i.e. block parts).

Nevertheless, we want to provide the above mentioned information about where a block starts and

ends. Today, we are using two custom attributes for that, but we are sure that it would be even

helpful for the standard railML to have two new, optional attributes

startOcpRef and analogous from

endOcpRef on the type block. Of course, these attributes are redundant in case of present block

parts.

Kind Regards,

Horst Naujoks

--

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

---

Subject: Re: Proposal: New optional attributes on type block  
Posted by [horst.naujoks](#) on Fri, 19 Jul 2013 15:32:07 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Dear Mr. Naujoks and all others,

first, to avoid misunderstandings also for other readers: I guess this is about circulations! ("Physical location where a block begins and ends" may also sound like "block" in the meaning of "tokenless train operation" along a section of track.)

- > In the overview mode we are delivering only blocks without containing details (i.e. block parts).

Actually, there is such a possibility to have <blocks> in a circulation which have no actual train/trainPart in the current RailML file. But this possibility is "one level up": At <blockParts>, not at <blocks>.

You can read it in the example explanation e. g. at [1] (which is linked e. g. from [2]):

- > Fahrt ohne Referenz auf einen Zugteil: mission=fullRun oder emptyRun;
- > trainPartRef darf nicht angegeben sein, startOcpRef, endOcpRef, begin und end müssen angegeben sein.
- > Dienst (der keine Fahrt ist): mission =shunting (Rangierdiensten), =maintenance (Wartungsdiensten), =standBy (Bereitschafts- und Reservedienste), =preheating (Vorheizdiensten), =refuel (Auffüllen von Vorräten, Tanken usw.); trainPartRef darf nicht angegeben sein, startOcpRef, endOcpRef, begin und end müssen angegeben sein, startOcpRef=endOcpRef.

Run w/o reference to a train part: mission=fullRun or emptyRun; trainPartRef must not be set, startOcpRef, endOcpRef, begin and end have to be set.

Duty which is no actual run: mission = shunting, maintenance, standBy, preheating, refuel; trainPartRef must not be set, startOcpRef, endOcpRef, begin and end have to be set, startOcpRef=endOcpRef.

So the solution is: Create a simple place-holder <blockParts> for the desired <block> and then create a <block> referring this place-holder <blockPart>.

(I'm aware that this may be easier written than done. It is not meant cynically. But is there a real need to create a second solution where already one exists or did I misunderstood the situation?)

With best regards,  
Dirk Bräuer.

[1] [www.irfp.de/download/railml\\_beispiel\\_umlauf.pdf](http://www.irfp.de/download/railml_beispiel_umlauf.pdf)  
[2] [www.wiki.railml.org/index.php?title=TT:block](http://www.wiki.railml.org/index.php?title=TT:block)

---

Am 30.04.2013, 14:43 Uhr, schrieb Horst Naujoks <[horst.naujoks@qnamic.com](mailto:horst.naujoks@qnamic.com)>:

> Hi all,  
>  
> I'd like to propose the following enhancement for an upcoming railML  
> release  
> (not necessarily 2.2):  
>  
> The physical location where a block begins and ends could be implicitly  
> determined by inspecting its  
> contained blockpart sequences and block parts respectively. From first  
> found block part the attribute  
> startOcpRef could be used and analogous from the last one the endOcpRef  
> attribute.  
> But this approach requires that all the block parts are present.  
> We are using railML in a scenario, where present blocks in two different  
> modes: overview or details.  
> In the overview mode we are delivering only blocks without containing  
> details (i.e. block parts).  
> Nevertheless, we want to provide the above mentioned information about  
> where a block starts and  
> ends. Today, we are using two custom attributes for that, but we are sure  
> that it would be even  
> helpful for the standard railML to have two new, optional attributes  
> startOcpRef and analogous from  
> endOcpRef on the type block. Of course, these attributes are redundant  
> in case of present block  
> parts.  
>  
> Kind Regards,  
>  
> Horst Naujoks

---

---

Subject: Re: Proposal: New optional attributes on type block  
Posted by [horst.naujoks](#) on Mon, 16 Sep 2013 15:01:32 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Dear Mr. Br  uer

First of all, thanks for your reply.

You are right, my input is about Blocks/BlockPart in the context of  
circulations.

Your suggested solution for my request (using startOcpRef/endOcpRef on  
'place-holder'  
BlockParts) is feasible, but not matching my primary intention.

My suggested new optional startOcpRef/endOcpRef attributes on the Block  
element should

provide a convenient way to deliver useful information at the block level even if its redundant to some information a level below (BlockParts). In fact, I wish to declare where a block starts, respectively ends, completely without the use of BlockParts.

Kind Regards,

Horst Naujoks

Dirk Br  uer wrote:

>  
> Dear Mr. Naujoks and all others,  
>  
> first, to avoid misunderstandings also for other readers: I guess this is  
> about circulations! ("Physical location where a block begins and ends" may  
> also sound like "block" in the meaning of "tokenless train operation"  
> along a section of track.)  
>  
>> In the overview mode we are delivering only blocks without containing  
>> details (i.e. block parts).  
>  
> Actually, there is such a possibility to have <blocks> in a circulation  
> which have no actual train/trainPart in the current RailML file. But this  
> possibility is "one level up": At <blockParts>, not at <blocks>.  
>  
> You can read it in the example explanation e. g. at [1] (which is linked  
> e. g. from [2]):  
>  
>> Fahrt ohne Referenz auf einen Zugteil: mission=fullRun oder emptyRun;  
>> trainPartRef darf nicht angegeben sein, startOcpRef, endOcpRef, begin  
>> und end m  ssen angegeben sein.  
>  
>> Dienst (der keine Fahrt ist): mission =shunting (Rangierdiensten),  
>> =maintenance (Wartungsdiensten), =standBy (Bereitschafts- und  
>> Reservedienste), =preheating (Vorheizdiensten), =refuel (Auff  llen von  
>> Vorr  tten, Tanken usw.); trainPartRef darf nicht angegeben sein,  
>> startOcpRef, endOcpRef, begin und end m  ssen angegeben sein,  
>> startOcpRef=endOcpRef.  
>  
> Run w/o reference to a train part: mission=fullRun or emptyRun;  
> trainPartRef must not be set, startOcpRef, endOcpRef, begin and end have  
> to be set.  
>  
> Duty which is no actual run: mission = shunting, maintenance, standBy,  
> preheating, refuel; trainPartRef must not be set, startOcpRef, endOcpRef,  
> begin and end have to be set, startOcpRef=endOcpRef.

>

> So the solution is: Create a simple place-holder <blockParts> for the  
 > desired <block> and then create a <block> referring this place-holder  
 > <blockPart>.

>

> (I'm aware that this may be easier written than done. It is not meant  
 > cynically. But is there a real need to create a second solution where  
 > already one exists or did I misunderstood the situation?)

>

> With best regards,  
 > Dirk Br  uer.

>

> [1] [www.irfp.de/download/railml\\_beispiel\\_umlauf.pdf](http://www.irfp.de/download/railml_beispiel_umlauf.pdf)  
 > [2] [www.wiki.railml.org/index.php?title=TT:block](http://www.wiki.railml.org/index.php?title=TT:block)

>

>

> ---

> Am 30.04.2013, 14:43 Uhr, schrieb Horst Naujoks <[horst.naujoks@qnamic.com](mailto:horst.naujoks@qnamic.com)>:

>

>> Hi all,

>>

>> I like to propose the following enhancement for an upcoming railML  
 >> release  
 >> (not necessarily 2.2):

>>

>> The physical location where a block begins and ends could be implicitly  
 >> determined by inspecting its  
 >> contained blockpart sequences and block parts respectively. From first  
 >> found block part the attribute  
 >> startOcpRef could be used and analogous from the last one the endOcpRef  
 >> attribute.

>> But this approach requires that all the block parts are present.

>> We are using railML in a scenario, where present blocks in two different  
 >> modes: overview or details.

>> In the overview mode we are delivering only blocks without containing  
 >> details (i.e. block parts).

>> Nevertheless, we want to provided the above mentioned information about  
 >> where a block start and  
 >> ends. Today, we are using two custom attributes for that, but we are sure  
 >> that it would be even  
 >> helpful for the standard railML to have to new, optional attributes  
 >> startOcpRef and analogous from  
 >> endOcpRef on the type block. Of course, these attributes where redundant  
 >> in case of present block  
 >> parts.

>>

>> Kind Regards,

>>

>> Horst Naujoks

>

>

--

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

---