

---

Subject: operatingDay

Posted by [tobias](#) on Tue, 23 Aug 2005 08:29:04 GMT

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

---

In section 2.3.2 of the timetable specifications, the operatingDay element is described. I feel the specifications does not fully explain how the dayTypes element should be interpreted. In particular, I don't quite understand the priority order of the dayTypes.

1. First consider Christmas. The 25th and 26th of December are both holidays. The 27th is both afterHoliday and afterAfterHoliday. I presume that in this case, afterHoliday takes preference?

2. Next consider Easter. Both Good Friday (in 2005 the 25/3) and Easter Day (27/3) are holidays. Is the 26/3 afterHoliday or beforeHoliday?

Regards,  
Tobias Bende

---

---

Subject: Re: operatingDay

Posted by [Joachim.Rubröder](#) on Thu, 25 Aug 2005 13:53:41 GMT

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

---

<br><font size=2 face="sans-serif">I agree that this is a weak point if you use this kind of description instead of bitmasks. <br>But this is unfortunately the way, the daytypes are used within DB.<br><br>

It's still better than within SBB, there the 27th of December is an afterHoliday and therefore must be a monday. ;-)</font>

<br>

<br><font size=2 face="sans-serif">The right order should be (strongest first):</font>

<br><font size=2 face="sans-serif"><br>

1. holiday<br>

2. afterHoliday</font>

<br><font size=2 face="sans-serif">3. beforeHoliday</font>

<br><font size=2 face="sans-serif">4. afterAfterHoliday</font>

<br>

<br><font size=2 face="sans-serif">Regards,</font>

<br><font size=2 face="sans-serif">Joachim Rubröder</font>

---

---

Subject: Re: operatingDay

Posted by [Vasco Paul Kolmorgen](#) on Mon, 29 Aug 2005 01:07:10 GMT

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

---

Joachim, please stop to send messages in HTML-format, only plain text will be allowed at the RailML-Newsserver to avoid active or executable parts of messages (may be virus or spyware!)

I have translated your message in plain text:

On Thu, 25 Aug 2005 13:53:41 +0000 (UTC), j.rubroeder@sma-partner.ch wrote:

> I agree that this is a week point if you use this kind of description instead of bitmasks.  
> But this is unfortunately the way, the daytypes are used within DB.  
>  
> It's still better than within SBB, there the 27th of December is an afterHoliday and therefore must be a monday. ;-)  
>  
> The right order should be (strongest first):  
>  
> 1. holiday  
> 2. afterHoliday  
> 3. beforeHoliday  
> 4. afterAfterHoliday  
>  
> Regards,  
> Joachim Rubröder

Best regards,

--

Dipl.-Ing. Vasco Paul Kolmorgen  
RailML-Konsortium  
Telefon: +49-351-46676939 Telefax: +49-351-46676940  
Zeunerstrasse 38; D-01069 Dresden [www.railml.org](http://www.railml.org)

---

Subject: Re: operatingDay  
Posted by [martin.weichert](#) on Thu, 01 Sep 2005 16:01:25 GMT  
[View Forum Message](#) <> [Reply to Message](#)

Hello,  
I hope it is OK to just step into the discussion as a newcomer with some questions about railML.

I am reacting to the message (originally by Joachim Rubröder):  
>> I agree that this is a week point if you use this kind of description instead of bitmasks.  
>> But this is unfortunately the way, the daytypes are used within DB.  
>>  
>> It's still better than within SBB, there the 27th of December is an afterHoliday and therefore must be a monday. ;-)  
>>

>> The right order should be (strongest first):

>>  
>> 1. holiday  
>> 2. afterHoliday  
>> 3. beforeHoliday  
>> 4. afterAfterHoliday  
>>  
>> Regards,  
>> Joachim Rubröder

Now my comments/questions:

Thus:

- X = holiday : if X is listed as a <holiday> entry
- X = afterHoliday : if X != holiday, X-1 = holiday.
- X = beforeHoliday : if X and X-1 != holiday, X+1 = holiday.
- X = afterAfterHoliday: if X and X-1 and X+1 != holiday, X-2 = holiday.
- X = regularday : if X and X-1 and X+1 and X-2 != holiday.

Do I understand this correctly?

Then the following problem remains: I cannot distinguish between  
- "holiday-before-holiday" (a holiday that is followed by another holiday), and  
- "holiday-not-before-holiday" (a holiday that is followed by a non-holiday).

Consider a timetable with a simple rule that a "free-day" (any Saturday, Sunday or holiday) shall be run as a Saturday if it is followed by another free-day;  
but as a Sunday if it is followed by "work-day" (any day that is not a Saturday, Sunday or holiday).

I try to define something like this:

```
<operatingPeriods>  
<holidays>  
<holiday holidayDate="2006-12-25" description="1st Christmas day, 2006 (a Monday)" />  
<holiday holidayDate="2006-12-26" description="2nd Christmas day, 2006 (a Tuesday)" />  
<holiday holidayDate="2007-12-25" description="1st Christmas day, 2007 (a Tuesday)" />  
<holiday holidayDate="2007-12-26" description="2nd Christmas day, 2007 (a Wednesday)" />
```

```
<!-- and some more... -->
</holidays>
```

```
<service serviceID="free-day" description="any Saturday, Sunday or
holiday" startDate="2006-01-01" endDate="2008-01-01">
<operatingDay operatingCode="1111111" dayType="holiday"/>
<operatingDay operatingCode="0000011" dayType="afterHoliday"/>
<operatingDay operatingCode="0000011" dayType="beforeHoliday"/>
<operatingDay operatingCode="0000011" dayType="regularday"/>
</service>
```

```
<service serviceID="work-day" description="anything that is NOT a
free-day" startDate="2006-01-01" endDate="2008-01-01">
<operatingDay operatingCode="0000000" dayType="holiday"/>
<operatingDay operatingCode="1111100" dayType="afterHoliday"/>
<operatingDay operatingCode="1111100" dayType="beforeHoliday"/>
<operatingDay operatingCode="1111100" dayType="regularday"/>
</service>
```

```
<service serviceID="free-day before free-day" description="(includes
Saturdays and 1st Christmas day...)" startDate="2006-01-01"
endDate="2008-01-01">
```

```
<!-- NOTE the question marks here! -->
<operatingDay operatingCode="????11?" dayType="holiday"/>
<operatingDay operatingCode="000001?" dayType="afterHoliday"/>
<operatingDay operatingCode="0000011" dayType="beforeHoliday"/>
<operatingDay operatingCode="0000010" dayType="regularday"/>
</service>
```

```
<service serviceID="free-day before work-day" description="(includes
most Sundays and 2nd Christmas day...)" startDate="2006-01-01"
endDate="2008-01-01">
```

```
<!-- NOTE the question marks here! -->
<operatingDay operatingCode="????00?" dayType="holiday"/>
<operatingDay operatingCode="000000?" dayType="afterHoliday"/>
<operatingDay operatingCode="0000000" dayType="beforeHoliday"/>
<operatingDay operatingCode="0000001" dayType="regularday"/>
</service>
```

```
</operatingPeriods>
```

But I still have question marks in some places.

Both dates 2006-12-26 and 2007-12-25 are Tuesdays, and both of them are holiday. By any definition that is based on dayType and day of week, they will be treated the same.

Yet 2006-12-26 should fall into "free-day before work-day" and 2007-12-25 into "free-day before free-day".

It seems that the definitions with "holiday", "beforeHoliday", etc. cannot make this distinction. Correct?

The rules are complicated, but still not complicated enough!?

Maybe I should just skip all the <holiday> definitions and restrict

myself to only using explicit bitmasks?

- Another question about dayTypes:

If I am not interested in the dayType "afterAfterHoliday" and want to treat

all such days as "regularday", can I define that in a simple way or do I always have to list an element

```
<operatingDay ... dayType="afterAfterHoliday" />
```

alongside the

```
<operatingDay ... dayType="regularday" />
```

with the same operatingCode?

If I don't list "afterAfterHoliday" in the <service ...> element,

will all "afterAfterHoliday" days be excluded from that service?

Best regards,  
Martin Weichert

---

---

Subject: Re: operatingDay

Posted by [Joachim.Rubröder](#) on Wed, 14 Sep 2005 16:42:29 GMT

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

---

Hello Martin,

- X = holiday : if X is listed as a <holiday> entry
- X = afterHoliday : if X != holiday, X-1 = holiday.
- X = beforeHoliday : if X and X-1 != holiday, X+1 = holiday.
- X = afterAfterHoliday: if X and X-1 and X+1 != holiday, X-2 = holiday.
- X = regularday : if X and X-1 and X+1 and X-2 != holiday.

seems to be a correct definition.

Now to your Problem with "holiday-before-holiday" and "holiday-not-before-holiday":

The 25.12. is both a holiday and a beforeHoliday but holiday is stronger.

If you like to define a train driving on all holidays but not on the holidays followed by other holidays, you have to use:

```
<operatingDay operatingCode="1111111" dayType="holiday"/> (on all holidays)
```

```
<special type="exclude" date="2005-12-24"/> (but not on 25.12.)
```

Your "free-day before free-day" should look like:

```
<service serviceID="free-day before free-day" description="(includes Saturdays and 1st Christmas day...)" startDate="2006-01-01" endDate="2008-01-01">
```

```
  <operatingDay operatingCode="0000011" dayType="beforeHoliday"/>
```

(all Saturdays and Sundays before Holidays)

```
<operatingDay operatingCode="0000010" dayType="regularday"/> (all regular Saturdays)
```

```
<special type="include" date="2005-12-24"/> (also on holiday 25.12., even on Mo-Fr)
```

```
</service>
```

Maybe you should just skip all the <holiday> definitions and restrict yourself to only using explicit bitmasks?

Now to your second question. Every day is either a holiday (if listed as holiday) or a regularday.

If a service has no other operatingDay defined - that's it.

If a service has a beforeHoliday definition, then every day can be (for this service) either holiday or beforeHoliday or regularday.

Kind regards,  
Joachim

---