Subject: Re: what about compressed RailML files?
Posted by Susanne Wunsch railML on Mon, 05 Nov 2012 22:19:37 GMT
View Forum Message <> Reply to Message

Hello Nilo and Dirk,

Nilo Menezes <menezes@multitel.be> writes:

> This is my first post on the RailML group. I wrote an internal tool
> that reads RailML 2.1 files and provide some operations on it (time
> table extraction and track export based on route). I work at Multitel,
> Belgium, at the Certification Laboratory.

Welcome Nilo at the railML community.

Please register as a railML developer if you already have worked with railML. [1] To many people think, railML is only used in German-speaking countries. ;-)

> Regarding your message, may I suggest using Gzip instead of zip?
>
> Why:
> 1) GZip is streaming friendly, you can read the compressed file
> directly, no need to decompress first. This also make GZip files very
> welcome on command line applications.
> 2) You can only add a single file to it. In fact, GZip does not
> specify internal files, all you have a single stream. To get the file
> name, we process the .gz file name itself.
> 3) The overhead is very small.
> 4) Most software libraries and languages provide GZip
> compression/decompression (Python, Ruby, C/ZLib, Java, C#, etc).

Thank you for your suggestion. It sounds very helpful.

> For the file extension:
> .railml for uncompressed files

+1

> .railml.gz for gzipped RailML files (following Unix tradition like
> .tar.gz or .tar.bz2)

+1

For one railML (instance) file gzip would be a nice option for saving file size and enabling streaming.

For multiple railML files, including an extension XML Schema file and/or

separated railML instance files (e.g. for <infrastructure> or <rollingstock>) the "normal" zip archive (RFC 1950) would help out.

All files in the archive should validate without any further files other than:

* railML XML schema files
* Dublin Core XML schema files
(* MathML XML schema files)

> Regarding the points you listed:
> On 05/07/2012 18:39, Dirk Bräuer wrote:
>
>> There are some questions we should consider:
>> - Do we recommend file extensions and if so, which?

> It is a very good idea. Anything different from .xml would be nice.
> I have a lot of problems opening large RailML xml files with the wrong
> tools on Windows. With .xml it is harder to create a specific file
> association too.

What do you think about Dirks suggestion to use *.railmlx for zipped files?

I would have no problems with this idea.

>> - Do we enforce Deflate compression algorithm or do we allow others?

> If we use gzip, this question would be already answered.

The deflate compression algorithm could be recommended for "normal" zip archives.

>
>> - Do we allow more than one RailML file in one ZIP file?

> I recommend only one file. If the user needs more files, he can create
> a tar or use another program for that.
> Maybe I'm missing something here, but what do you mean by more than
> one file? Would they share the same references? Is this grouping a
> kind of context somehow?

I hope to clarified this a bit. If this question keeps already not fully answered, please, give me a hint.

A tar archive has the disadvantage that one has to decompress the whole archive in order to get only single files from it. If we use the zip archive one could only extract and decompress single files from the archive.

>> - Do we enforce UTF-8 file names in the ZIP file or do we allow also the
>> older but default Ansi-437 ? (Bit 11 of GeneralPurposeBitFlag of the
>> CommonFileHeader of ZIP would allow to distinguish between both).

> UTF-8 is widely spread. Enforcing ANSI-437 can be annoying for
> international use. The European page for example is the 850. I'm not
> sure if these code pages are ANSI standards, I think they are just
> code pages created by IBM and Microsoft.
> UTF-8 is welcome on Windows, Mac OS X and Linux. So I think we would
> make everybody happy. If we adopt the GZip format, any problems
> regarding file name encoding would be solved by a simple rename.

That sounds good to me.

>> - Do we 'allow' or 'recommend' the compressed RailML files?

> It is very easy to accept both. On my tool, if you decide to use .gz,
> it will change very few lines of code. Uncompressed files are great
> when we are tweaking them. Compressed files are great for transmission
> and storage. I work with 150Mb XML files... I would not like to
> compress and uncompress them every time I change a letter or
> something.

+1

I would prefer a "good practice" style. There are multiple use cases
that may "feel blocked" or "unofficial" if we would _recommend_ "single
zip files".

Use Case A:

One large railML file containing pure railML without any extensions,
validating against the officially published railML XML Schemas.

-> useCaseA.railml (uncompressed)
-> useCaseA.railml.gz (gzipped)

Use Case B:

One large railML file containing railML and some extensions,
validating against the officially published railML XML Schemas
together with the extension XML Schema.

-> useCaseB.railml (uncompressed)
  useCaseB.xsd (extension XML Schema)

-> useCaseB.railmlx (compressed zip archive containing both files)

Use Case C:

Multiple railML files, which base on the same separated railML files, validating against the officially published railML XML Schemas

-> useCaseC_rollingstock.railml (uncompressed)
useCaseC_infrastructure.railml (uncompressed)
useCaseC_timetable_variant1.railml (uncompressed)
useCaseC_timetable_variant2.railml (uncompressed)

-> useCaseC.railmlx (compressed zip archive containing all above files)

Use Case D...

Variants of the above mentioned use cases.

Any further comments appreciated.

Kind regards...
Susanne

[1] http://www.railml.org//index.php/developers.html

--
Susanne Wunsch
Schema Coordinator: railML.common