
Subject: what about compressed RailML files?

Posted by _____ on Thu, 05 Jul 2012 16:39:04 GMT

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

Dear all,

with further circulation of RailML, we have increasing problems with RailML files which are sent uncompressed as E-Mail attachments. They become quickly larger than suitable for attachments, and also they are sometimes misunderstood by browsers or so as XHTML or whatever.

I therefore want to make a suggestion to provide an official supported way to pack a RailML file. I am aware that EXI is a possible solution but I fear that it is too complicated for a general acceptance.

So I would suggest to 'allow' or 'recommend' to put a RailML file into a simple ZIP file. That means, to pack it with the default Deflate compression algorithm and surround it with the local/common/central file headers of the ZIP file format.

The advantage of such compressed RailML files would be (possibly against EXI):

- That it is still possible to read or edit them with a common text editor after extracting with a common zip extractor. No special software is needed.
- That there are plenty possibilities to include the packing & unpacking in the own software either by own programming or a 'used' library. Both file format and Deflate algorithm are Public Domain. There are many programming solutions (libraries) already existing for the common platforms such as java.util.zip, zip, deflate.obj.

Of course, 'allowing' or 'recommending' compressed RailML files shall not mean to exclude uncompressed: Every software reading RailML shall accept both compressed and uncompressed (in the best case) or at least uncompressed (hopefully in a temporarily case only).

A RailML writing software can or shall make the output of compressed RailML files as the default. It should also allow the output of uncompressed RailML files, possibly on explicit user setting. It does not need to provide compressed output (as the user can pack it manually).

There are some questions we should consider:

- Do we recommend file extensions and if so, which?
- Do we enforce Deflate compression algorithm or do we allow others?
- Do we allow more than one RailML file in one ZIP file?
- 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).

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

For the moment, I would start with easy solutions and recommend:

- only Deflate compression algorithm,
- only one RailML file in a ZIP file,
- only UTF-8 file names as we also recommend UTF-8 for the coding of the RailML file.

To allow more can easily be done later, to allow less would be difficult...

I would prefer to define file extensions for both compressed and uncompressed RailML files. (So far, we use 'xml' as the file extension for RailML files only.) It should be unique file extensions, so no common ones, to prevent the user from mixing too much at his hard disc. (When providing a file-open dialog box for a RailML file, I would prefer to show the user the real RailML files only, no other XML or ZIP files.) Some possible extensions are *.railml for uncompressed and *.railmlx for compressed RailML files.

What do you think?

With best regards,
Dirk.

Subject: Re: what about compressed RailML files?
Posted by [Nilo Menezes](#) on Mon, 15 Oct 2012 10:49:32 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Dirk,

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.

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).

For the file extension:

..railml for uncompressed files

..railml.gz for gzipped RailML files (following Unix tradition like

..tar.gz or .tar.bz2)

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.

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

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

> - 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?

> - 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.

> - 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.

Best Regards,

Nilo Menezes

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

Subject: Re: what about compressed RailML files?

Posted by [Susanne Wunsch railML](#) on Tue, 06 Nov 2012 08:25:14 GMT

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

Sorry for responding to my own posting. I missed an important use case that is already practiced.

Susanne Wunsch <coord@common.railml.org> writes:

> 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.

Use Case E

Transferring relatively small single railML files from a server to mobile devices

These files may be best compressed using the EXI algorithm. [1]

-> useCaseE.railml (uncompressed)

-> useCaseE.railml.exi (EXI compressed)

Kind regards...

Susanne

[1] <http://www.w3.org/XML/EXI/>

--

Susanne Wunsch

Schema Coordinator: railML.common

Subject: Re: what about compressed RailML files?
Posted by [christian.wermelinger](#) on Fri, 07 Dec 2012 16:53:54 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello,

This is my first post. I am working at Qnamic in HÃgendorf, Switzerland.
Qnamic mainly uses
RailML for exchanging timetable and infrastructure data. Further
information can be found on the
developers page: <http://www.railml.org//index.php/developers.html?show=35>
Following my thoughts regarding file compression and file name extensions.

1. ZIP

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

From my point of view RailML standard should NOT define whether and how to
use compressed
ZIP archives in context of RailML. In the end it depends on the use-case
whether ZIP compression
shall be used, whether one or multiple files shall be included in a ZIP
file, which algorithm fits best
etc. Defining a standard leads to additional (and in the worst case even
unnecessary)
development effort.

2. File extension

>> - Do we recommend file extensions and if so, which?
>> .railml for uncompressed files
>> .railml.gz for gzipped RailML files (following Unix tradition like
>> .tar.gz or .tar.bz2)

That sounds good to me and follows a common pattern.

Regards
Christian

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

Subject: Re: what about compressed RailML files?
Posted by [coordination](#) on Sun, 16 Aug 2015 13:41:46 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear all,

some time has passed and a lot of trains departed since Dirk Braeuer of iRFP started this discussion about compressed railML files in 2012. In the meantime some programmes got certified, railML's usage has spreaden wider and a lot of partners had joined railML.org.

The issue of file compression was described in a ticket (<http://trac.railml.org/ticket/181>) and some programmes using *.railml for uncompressed and *.railmlx for ZIP compressed RailML 2.2 files. Dir Braeuer described the current state in railML's wiki at <http://wiki.railml.org/index.php?title=CO:fileConventions>. To enlarge and finish this wiki page I want to ask al the railML developers (and users too) the following questions:

- 1) Do you use file compression in your programmes exports or do you read compressed files? If not, do you plan to use in near future or why not?
- 2) Do you use ZIP compression only or one of the other discussed compression algorithms (TAR, GZ, EXI
- 3) Do you allow any one railML file per archive or multiple? What's with exports of separted part schemes (TT/IS/RS in separate files)?
- 4)What experiences did you make or what feedback do you got?
- 5) Other questions or ideas regarding this issue?

We'll collect all the meanings and will report during the next railML conference about this issue.

Best regards,

--

MSc. Vasco Paul Kolmorgen

Phone: +49-351-46676939

D-01069 Dresden; Germany www.railml.org

--

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

Subject: Re: what about compressed RailML files?
Posted by [Ferri Leberl](#) on Wed, 19 Apr 2017 12:49:22 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear all,

Does https://wiki.railml.org/index.php?title=CO:fileConventions#C_ompressed_railML_files reflect the currant approach towards file compression?

How did the attitude towards compressing several .railml-files in a single .railmlx-file develop?

Thank you in advance for the answer.

Ferri Leberl

Subject: Re: what about compressed RailML files?
Posted by [Ferri Leberl](#) on Wed, 26 Apr 2017 13:20:23 GMT
[View Forum Message](#) <> [Reply to Message](#)

Ticket #181 has been closed.

Subject: Re: what about compressed RailML files?
Posted by _____ on Thu, 18 May 2017 13:23:16 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear Ferri,

Am 19.04.2017 um 14:49 schrieb Ferri Leberl:

- > Does
- > https://wiki.railml.org/index.php?title=CO:fileConventions#Compressed_railML_files
- > reflect the current approach towards file compression?

From our side: It does.

- > How did the attitude towards compressing several
- > railml-files in a single .railmlx-file develop?

From our side: Currently not supported. One .railml into one .railmlx file only. No known demand on anything else.

Best regards,
Dirk.
