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, zlip, 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 tho 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.
