Subject: unique IDs
Posted by Joachim Buechse on Thu, 25 Sep 2003 13:21:23 GMT
View Forum Message <> Reply to Message

On yesterdays meeting we discussed unique IDs (for tracks). I would like to extend the focus and make a suggestion regarding IDs.

XML defines the concept of ID and IDREF. IDs need to be unique within an XML document (not just within a namespace or 'tagspace'). IDREFs are the closest you can get to pointers with standard XML. They are sometimes used for XML-encoding of directed acyclic graphs (DAGs) or other data structures that can not be represented as a tree.

For purely practical reasons - or should I say from bad experience - we at Ergon try to keep ID values free of semantics (ie arbitrary strings with min/max length). Please note that this is in sharp contrast to SQL where primary keys or foreign keys are often choosen as compound values.

<uniquekeygeneration>
It is simple to generate (globally) unique keys. Our typical approach is to use the 32-bit ip address of the host that creates the node/key concatenated with 64-bit currentTimeMillis (ie. the time in Milliseconds since 1.1.1970). As long as the creation rate is (conceptually) lower than 1000 nodes per second the chance of a collision is very low. If the creation rate can be higher than 1000 nodes per second we simply add another 32-bit local counter (which may or may not be reset every millisecond) which allows for the creation of 4 billion unique IDs per Millisecond with an ID length of 128bit = 16 byte binary = 22 base64 symbols.

For a standard like RailML which is here to stay for a while it might be advisable to allow IPv6 addresses (128-bit). The Unsigned currentTimeMillis will overflow around 2106. (How old are the oldest tracks?). But even after an overflow collisions are very unlikely.
</uniquekeygeneration>

Hence my suggestion is:

RailML should use IDs (attribute with the name ID) for main elements like track, line etc. IDs MUST be of type string. IDs SHOULD have a minimal length of 8 and a maximal length of 32 symbols. Applications SHOULD create IDs that are globally unique. Applications SHOULD preserve IDs when importing and reexporting a data set with RailML. The content of IDs MAY be arbitrarily choosen but SHOULD NOT be semanticly interpreted by an application. IDs SHOULD NOT be used to order elements.

Please note: I do not suggest the IDs should be used to replace attributes like lineId, trackId, etc in the current schema [except where thoose are only used to reference elements].

Please excuse the lengthy mail.

Best Regards,
Joachim Buechse

--
buechse@ergon.ch, Phone +41 1 268 89 58, Fax +41 1 260 20 65
Ergon Informatik AG, Kleinstrasse 15, 8008 Zuerich, Switzerland
http://www.ergon.ch

_____

e r g o n smart people - smart software