

## Proposal for a new reserved attribute - xml:version (xml:sn)

### Why?

Let's say, we are developing a database, a database which is closely bound to the XML data format because both control commands and stored data are in XML (it's a kind of an XML/object database with Web Services interface). In that database we implement various commands - besides the administration, we know four basic operations: creating, modifying, deleting and querying. These operations are easy to implement and perform in a single user environment, but nowadays it is usual to have multi-user capabilities.

The thing what we talk about here is the behavior of the database in the case when multiple users are trying to modify the same data. The update should be allowed only for one of the users (usually for the quickest).

To ensure that behavior, we need to do a locking - e.g. who comes first, gets the lock and the others are denied. This approach has two problems - others can not view the data (unless they do not agree with read only access) and in the case that the user who gets the lock fails and does not unlock the item, all that data stays in read only mode forever (in reality - until a user with higher privileges clears locks).

A better approach would be that all users are handled same way in the reading phase, and after one of them saves the modified data, the modification by others are denied. To implement this behavior, we need an attribute for each data item which will hold the version of that item. Whenever the data gets modified, the version gets incremented.

The important thing is, that the modification is allowed only when the version of the data is same as the version which was sent to user at the moment of reading - when the version of data in the database is higher, it means that the data were modified by somebody other and this second user is informed by the application about that (and can take various actions - force overwriting, retrieve the new version, compare them, merge changes, etc).

The attribute which holds the serial number is however not related to the stored data, but to the implementation of the database itself, in other words it belongs to the system and that's why it would be nice to have it in the reserved xml namespace.

### Alternatives

The attribute name does not need to be called xml:version, because that can be mistaken with the version of the XML recommendation. Another possible name is then xml:sn (serial number).

A totally different solution would be to use an encapsulation method (as is seen in SOAP), where we could make a two element structure where one will store the data and the other the version of that data, but this is not as efficient as the proposed attribute (if the attribute will belong to the xml reserved namespace, it will never interfere with other namespaces). The efficiency of the attribute is more visible in the example where it is present in every element - using the encapsulation for that purpose will create very big file and unnecessary double markup.

### Examples

An example of an object retrieved from our database (every user knows that a contact is atomic and is not possible to update just the first name):

```
<?xml version="1.0" encoding="UTF-8" ?>
<contact xml:sn="123" xmlns="schemas.rozsnyo.com/grouware/addressbook">
  <firstname>Daniel</firstname>
  <lastname>Rozsnjó</lastname>
</contact>
```

When the contact isn't atomic, we need the versions of all elements as the protection against sub-update from other users:

```
<?xml version="1.0" encoding="UTF-8" ?>
<contact xml:sn="123" xmlns="schemas.rozsnyo.com/grouware/addressbook">
  <firstname xml:sn="456">Daniel</firstname>
  <lastname xml:sn="789">Rozsnjó</lastname>
</contact>
```