

KEP: 3 Title: Introduction of 'subevent' sub-tag for 'exclusion' from 'recurrence' Version: \$Revision\$  
Last-Modified: \$Date\$ Author: Georg Greve <greve@kolabsys.com> Status: Draft Type: Design  
Content-Type: text/x-rst Created: 2010-11-17

## Abstract

The Kolab storage format models allows recurrence to have 'non-occurrence' exceptions already, but it does not allow to have 'modification' exceptions, e.g. change of time or location. The way this has been handled is to create a 'non-occurrence' exception, and then create a separate event for that day with the default values of the recurring event.

This has been understood as a suboptimal solution for a long time because the clients cannot easily connect that new event with the recurring event for future modifications, while the user logically sees them as connected.

The solution is to introduce a hierarchically nested 'subevent' sub-tag for an exception, which takes all default values from the recurring event itself, and only contains the differences to these default values. [1]

## Update to the XML Format

The type for datetime storage in Kolab XML is modified as follows:

- A 'subevent' XML tag **MAY** be added hierarchically nested within an 'exclusion' to a 'recurrence' of an 'event' object.
- There **MUST** be only one 'subevent' per 'exclusion'.
- All event values of the 'subevent' default to the 'event' within which it is nested. Values within 'subevent' change these values for this 'exception' from the 'recurrence' only.

## Example

Moving one instance of a recurring event from date1, a normal date in the recurrence cycle, to new-date1 would express itself as follows

```
<event> ...
  <recurrence> ...
    <exclusion>date1
      <subevent>
        <start-date>new-date1</start-date>
      </subevent>
    </exclusion>
  </recurrence>
</event>
```

## Upgrade Path

When this KEP becomes active, the version number of the Kolab Storage Format specification will be updated to 2.1.

New clients that correspond to 2.1 will be fully compatible with older data sets.

Older clients can continue to behave as before at their own choice, and will remain consistent with the 2.0 specification and no data will be lost or corrupted. So whilst this is important, it is not urgent. Older clients will however not be able to display recurrence exceptions of newer clients properly, so it is highly recommended to move to the newer 2.1 format when feasible.

# References

## Copyright

This document has been placed in the public domain.

---

1

event XML 1.1 (fix recurrances),  
(<http://kolab.org/pipermail/kolab-format/2008-December/000876.html>)

Konold