MicroKernel Revision Model

MVCC by Example

stefan@apache.org
Topics

- MicroKernel Object Model
- MVCC in action
- Pros/Cons of MVCC approach
Commit

parentId
rootNodeId
commitTS
msg

Head

symbolic reference to most recent commit
Node

Map<String, String> properties
Map<String, ChildNodeEntry> childEntries

ChildNodeEntry

name
id
an example in 4 steps...
step 0: create an empty root node
step 1: add nodes /a, /a/c, a/d and /b

+ /a : { c:{}, d:{} }  
+ /b : {}
step 2: add node /b/e

+ /b/e {}
empty repository
+/a: { c:{}, d:{} }
+/b: {}
step 3: move /a to /x

> /a : /x
empty repository
+/a:{ c:{}, d:{} }
+/b:{}
+/b/e:{}

writes
repository tree
HEAD
MVCC pros

- writers don't interfere with readers and vice versa
- snapshot isolation (repeatable reads)
- improved concurrency
- minimal and very narrow point of synchronization on commit
- cheap copy & move
MVCC cons

- *heavy* on resources
- no stable identifiers across revisions (except for path), i.e. jcr-style references require an index
- non-trivial garbage collection/revision compacting problem...