JcrContentModelAreas

Areas (Authoring, Staging, Live)

This section refers to areas as abstract concepts, not necessarily modeled as separated content areas.

Requirements:

- independent from each other (changes to /authoring/foo don't influence staging/foo)
- self-contained

See also:

- http://www.mail-archive.com/sling-dev@incubator.apache.org/msg05812.html (Felix Meschberger from the Sling team in favor of the multi-workspaces approach)

Options:

Workspaces

Advantages:

- Physically separated nodes with same UUID possible ("corresponding nodes" concept)
- Separated, independent
- Self-contained
- Easy migration to an environment with separate authoring/staging/live repository servers + replication

Issues:

- Changing the site structure requires to remove the involved nodes from all non-authoring areas (like now)
- Copying access control definitions?
- `Node.update(workspace)` copies the subtree:
  - either nodes must be kept in a flat structure (no hierarchy)
  - or nodes and properties must be copied manually
  - or use referenceable child nodes

```
/de
  /foo < mix:referenceable
    /lenya:document < mix:referenceable
      /jcr:content = "Hello"
    /bar
```

Publish single node:

```
getItem("/de/foo/lenya:document").update("staging");
```

Publish subtree:

```
getItem("/de/foo").update("staging");
```

Separated Areas in a Single Workspace

Advantages:

- Simple access

Issues:

- Copying access control definitions required
- Different UUIDs:
  - Either rewrite links
  - Or don't use JCR UUIDs but custom String properties for UUIDs

Separated Areas in a Single Node

Example:

```
/foo
  /lenya:authoring
    /jcr:content = "Hello"
  /lenya:staging
    /jcr:content = "Hello"
  /lenya:live
    /jcr:content = "Hello"
  /bar
    /lenya:authoring
    /…
```

Advantages:

- Simple
- Site structure can be changed without removing staged or published content

Issues:

- Areas not self-contained
- No direct URI-to-path mapping possible
- Changes to site structure would be visible in all areas immediately

Labelled Versions

- Issues:
  - All changes applied immediately, without calling `Session.save()`
  - Access control to version history?

Submit:

```
VersionHistory history = node.getVersionHistory();
Version base = node.getBaseVersion();
boolean move = true;
history.addVersionLabel(base.getName(), "staging", move);
```

Obtaining live version:

```
VersionHistory history = node.getVersionHistory();
Node liveVersion = history.getVersionByLabel("live");
Node liveNode = liveVersion.getChild("jcr:frozenNode");
```