

Concept - MetaData as mixin

Purpose

The Problem

The fact that Magnolia stores MetaData on a separate subnode creates additional effort at various places

- makes complex queries (JCR_SQL2/JQOM) more complicated (if not outright impossible) - we have to use joins to access MetaData (e.g. for sorting by values in metadata)
- doubles the amount of nodes in repo
- duplicates properties jcr:created and jcr:createdBy
- forces to treat MetaData separately in Java
- makes exported xml harder to read

Goals

Simplify accessing MetaData information.

Proposal

Design

The MetaData subnode hosts a range of properties - this chart details what will happen with them:

old property	new property (defined by)	remark
MetaData/mgnl:title	<none>	not in use anymore, will not be supported
MetaData/mgnl:creationdate	mgnl:created (mgnl:created)	can't use jcr:created -> it's AUTOCREATED and PROTECTED
<none>	mgnl:createdBy (mgnl:created)	new property keeping who originally created something (as with mgnl:created it will be set by us)
MetaData/mgnl:templatetype	<none>	not in use anymore, will not be supported
MetaData/mgnl:lastaction	mgnl:lastActivated (mgnl:activatable)	
MetaData/mgnl:activatorid	mgnl:lastActivatedBy (mgnl:activatable)	
MetaData/mgnl:activated	mgnl:activationStatus (mgnl:activatable)	boolean - jcr is not powerful enough to have it auto-generated
MetaData/mgnl:template	mgnl:template (mgnl:renderable)	we'll create an additional mixin: mgnl:renderable
MetaData/mgnl:authorid	mgnl:lastModifiedBy (mgnl:lastModified)	
MetaData/mgnl:lastmodified	mgnl:lastModified (mgnl:lastModified)	
MetaData/mgnl:comment	mgnl:comment (mgnl:versionable)	

Even though JCR defines a mixin, [mix:lastModified](#), we use our own mixin to have greater control. The JCR specification does not clearly define when these properties should be updated, JCR 3.7.11.8.

So the MetaData subnode will be replaced by the following:

[mgnl:lastModified]

MIXIN

- mgnl:lastModified (DATE)
- mgnl:lastModifiedBy (STRING)

[mgnl:activatable]

MIXIN

- mgnl:lastActivated (DATE)
- mgnl:lastActivatedBy (STRING)
- mgnl:activationStatus (STRING)

[mgnl:renderable]

MIXIN

- mgnl:template (STRING)

[mgnl:created]

MIXIN

- mgnl:created (DATE)
- mgnl:createdBy (STRING)

[mgnl:versionable]

MIXIN

- mgnl:comment (STRING)

Implementation details

agreed

- single class NodeTypes hosting our mixins as inner classes
 - Mixins define the constants for the names of the properties defined by the corresponding mixing
 - Mixins define convenience methods to set properties
 - Property setting methods on Mixins will check for appropriate node type - no checks on legacy MetaData + MetaDataUtil
- constants for jcr-1.0 defined NodeTypes or Properties can be taken from JcrConstants (jackrabbit common)
 - if we realize we'd need to keep jcr-2.0 defined stuff, we'd add an interface for it

to be discussed

- MetaData will be adapted to be backwards compatible (use new property names and wrap workingNode - no longer a subnode)
- Property-files for PropertyImportExport (e.g. for tests) have to be adapted -> provide script to detect?

Migration

This step would require a very well prepared migration. Hence we'd have to consider:

- migration effort
 - update custom node types
 - add mixin
 - remove subnode
- update utils and add filter to filter the mgnl:* properties
- search in templates (and other places)
 - will have to be re-written
 - have a query-wrapper which warns
- STK
 - we have to update all models and utils which used meta data (mainly templates)
- Communication
 - this is considered a change in the templating
 - we have to communicate that
- Provide a crawler

- similar as in integration test
- check there's no warnings (related to this change)

Content Migration

- bootstrap files
 - filter which changes the files on the fly
 - put warnings in the log
 - Q: can we provide a tool for converting the bootstrap files and get them cleaned once and for all?
 - A: will try to do so
- update task for migrating live content, in all workspaces
 - not using recursion (some clients have deep hierarchy)
 - safe after each page modification)

Hint: similar migration was done between Magnolia 3.5.9 and 3.6 - see

Error rendering macro 'jira'

Unable to locate Jira server for this macro. It may be due to Application Link configuration.

for details.

Backwards compatibility in Templating

TemplatingFunctions.metadata(Content/Node)

JspTemplatingFunctions.metadataProperty(Node/ContentMap, String)

JspTemplatingFunctions.metadataTemplate(Node/ContentMap)

Templates query for properties using the old names, both with and without the prefixes.

Templates that access the MetaData nodes directly will not work.

Templates that access the MetaData via ContentModel will still work, they will get the old MetaData object.

Q: Do we need to support both styles of property names and map old ones to new names?

A: we'll support it but warn users there's old style templates. Consider refactoring the above three methods to use common code base.

Lifecycle operations

When a node is deleted the `mgnl:deleted` mixin is added, it has properties `mgnl:deletedBy` and `mgnl:deletedOn`.

Q: Should these properties be defined on the mixin in nodetypes.xml?

A: Yes

Q: Can we rename `mgnl:deletedOn` to `mgnl:deleted`?

A: we'll use consistent naming for our Date properties (lastModified, lastActivated, created, deleted)

When a node is versioned or deleted we add a property `mgnl:comment` on the MetaData node

Q: Where do we store this now? Should it be part of the mixins?

A: mixin `mgnl:versionable` -> property `mgnl:comment` (String) (Do a text search to find occurrences)

Activation

`info.magnolia.cms.exchange.ActivationUtil` will be used to set and query properties related to activation.

Versioning

old property	new property (defined by)	remark
--------------	---------------------------	--------

mgnl:comment	mgnl:comment	just added explicitly
--------------	--------------	-----------------------

[mgnl:versionable]

MIXIN

- mgnl:comment (STRING)

Deleted nodes

old property	new property (defined by)	remark
mgnl:deletedOn	mgnl:deleted	consistency - see other date properties
mgnl:deletedBy	mgnl:deletedBy	just added explicitly
mgnl:comment	mgnl:comment	just added explicitly

[mgnl:deleted]

MIXIN

- mgnl:deleted (DATE)
- mgnl:deletedBy (STRING)
- mgnl:comment (STRING)

Modules to check and update

<TO BE VERIFIED>

- Observation: be sure to consider Jan's comment: we need a listener excluding mgnl:* (and maybe jcr:lastModified*)
- STK
- Form
- Mail
- InplaceTemplating
- Data
- Scheduler

API Changes

- info.magnolia.jcr.util.NodeUtil#ALL_NODES_EXCEPT_JCR_FILTER deprecated since there are node node types with jcr prefix
- info.magnolia.jcr.util.NodeUtil#EXCLUDE_META_DATA_FILTER deprecated since the meta data nodes wont be there and there are no node types with jcr prefix
- info.magnolia.jcr.MgnlNodeTypeNames introduced containing constants for node types
- info.magnolia.jcr.MgnlPropertyNames introduced containing constants for property names
- info.magnolia.cms.core.MgnlNodeType deprecated replaced with MgnlNodeTypeNames and MgnlPropertyNames
- info.magnolia.cms.core.MetaData deprecated and replaced by NodeUtil and ActivationUtil

Q: What do we do with the possibility of setting arbitrary properties on the MetaData node? The API supports this. Is it enough to allow the supported parameters and fail on unknown names?

A: Will fail if someone passes in "unknown" property name.

Q: What is the state of audit logging? We log modification as we update modification on nodes

A: Needs to be kept

Note, there will places where we have to exclude jcr properties. Places where this has been done includes MgnlUserManager and MgnlGroupManager.

Note, the MetaData class will remain as long as we keep the content API.

Won't do's

Note: This section only contains "decisions" that are not yet integrated in their proper section are cannot be integrated because they were dropped.

- Wouldn't that be a good occasion to switch to CND notation (see

 Unable to locate Jira server for this macro. It may be due to Application Link configuration.

, [JR Website](#) for additional

information)?

- will not be considered right now
- Should we also introduce a more enhanced/flexible activation status using jcr calculated values (would finally enable things like sort per activationStatus)?
 - judging from a post on the [jr mailing list](#) it's not supported out-of-the-box
 - we won't solve it with the current step

Links

- General information about [jcr nodetypes](#)
- List of [nodetypes defined by jcr 2.0](#)
- [Minutes](#) from meeting on Nov 7th, 2012