

Concept outdated - DAM 1.0 (Review)

Purpose

Overview of the current DAM implementation and open Issues/Questions.

Existing documentation

- [Current Implementation](#)
- [Tracking DAM copies and uses](#)
- [DAM Workspace Structure](#)
- [Concept - DAM 1.0 future improvements](#)

Concepts to clarify

Metadata

AssetProvider assembles the **DamAsset** based on the asset node from the workspace and the related DAM asset.

- Sets the Meta-Data source which allows for flexible changing the ways of how meta-data is fetched and from where (typically - from the asset node internals, but could be from wherever).

Stored informations on an Asset:

- `mgnl:asset language` [Asset.getLanguage\(\)](#) --> returned type: String or Local ?
- `mgnl:asset name` [Asset.getName\(\)](#)
- `mgnl:asset title` [Asset.getTitle\(\)](#)
- `mgnl:asset type` ???
- `mgnl:asset subject` [Asset.getSubject\(\)](#)
- `mgnl:asset description` [Asset.getDescription\(\)](#)
- `mgnl:asset caption` [Asset.getCaption\(\)](#)
- `mgnl:asset copyright` [Asset.getCopyright\(\)](#)
 - `jcr:content extension` [Asset.getFileExtension\(\)](#)
 - `jcr:content fileName` [Asset.getFileName\(\)](#)
 - `jcr:content height` [Asset.getHeight\(\)](#)
 - `jcr:content size` [Asset.getSize\(\)](#)
 - `jcr:content width` [Asset.getWidth\(\)](#)
 - `jcr:content jcr:data` [Asset.getBinaryData\(\)](#)
 - `jcr:content jcr:mimeType` [Asset.getMimeType\(\)](#)

Information stored in the Meta-Data content node (these are the optional properties that a content node can have)

- `mgnl:content caption` [Asset.getCaption\(\)](#)
- `mgnl:content title` [Asset.getTitle\(\)](#)
- `mgnl:content description` [Asset.getDescription\(\)](#)
- `mgnl:content copyright` [Asset.getCopyright\(\)](#)

Questions

1. `mgnl:asset` attributes are not exposed by the Asset Implementation (language/title/...). Is this correct?
2. Meta-Data: Should we keep the concept of a MetaData object?
 - a. copyright is this not a Asset attribute ?
 - b. how to handle duplicate (title/description) defined on Asset and Meta-Data ?
3. Should we introduce a concept of CopyRight ?

Proposal

Get meta-Data from DAM asset, and override it if existing on the content node.

Decision

Remove the Meta-Data concept.

Add caption and copyright properties under asset Node.

It will be to responsibility of modules using DAM to override these properties if they come from another place (STK, textImage model class)

Asset Node Type

Current DAM workspace hierarchy structure:

- `mgnl:folder` (`nt:folder`)
 - `mgnl:asset` (`nt:file`)
 - description
 - title
 - ...
 - `mgnl:resource` (`nt:resource`)
 - `jcr:data` (type="Binary")
 - size
 - ...
 - `mgnl:asset` (`nt:file`)
 - ...

Open Issues/Questions:

1. Version: How to handle versions?
2. For external provider we said that we should at least have one DAM Node pointing to the external Content (binary) but the current `mgnl:asset` required a `mgnl:resource` with a `jcr:data` (type="Binary")
Should we introduce a new type for external Content ?

In addition from **previous concept page**:

- a. 'Although assets could come from different sources, the asset node will have standard fields that will enable it to be displayed in lists etc'
 - b. 'AssetNode has a field specifying `assetProviderType`' **Not Implemented**
3. For multivariant assets (i18n-ed, multiple sizes etc.) it might support `multiasset`.

Proposal:

- `mgnl:folder` (`nt:folder`)
 - `mgnl:asset` (`nt:file`)
 - `assetProviderType`*
 - title
 - ...
 - `mgnl:resource` (`nt:resource`)
 - `jcr:data` (type="Binary")
 - size
 - ...
 - `mgnl:asset` (`nt:file`)
 - ...
 - `mgnl:externalAsset` (`mgnl:content`)
 - `assetProviderType`*
 - title
 - ...
 - `mgnl:multiasset`(`nt:folder`)
 - `mgnl:asset` (`nt:file`)
 - `mgnl:asset` (`nt:file`)
 - ...

(`mgnl:multiasset`) this way we could potentially group assets that should be presented as one within Magnolia, but still allow each variant to keep own set of properties and be indexed separately.

Still How to handle version?

Decision

Keep the current JCR structure.

MultiAsset, ExternalAsset will be better defined in another SPRINT.

Asset Provider functionalities

Extend the Asset provider functionalities:

- getAssets(Node, String) List<DamAsset> : If the Identifier refer to a DAM folder, return the list of child assets
- getAssetVersion(...)

General questions

- In previous STK/DMS collaboration - link to a document can be a path or a Identifier. Do we support path?
- Currently in info.magnolia.dam.asset, there are Asset, DamAsset, and DamAssetWrapper (And considering there is no more metadata) (Im not talking about stk AssetWrapper here)
 - Do we really need Asset and DamAsset - why, is this just an artifact from DMS?
 - What is the value of DamAssetWrapper

Decision

Only support Identifier link to DAM.

Review the current DAM API

- Remove DamAssetWrapper
- Rename DamAsset to InternalAsset

Open Questions/Tasks

Links (related to MGNLSTK-1048 issue)

Create a DAMURI2RepositoryMapping extends URI2RepositoryMapping (Equivalent of DMSURI2RepositoryMapping)

Create a DAMDownloadServlet (Equivalent of DMSDownloadServlet)

These two class should use the DAM class to get the correct AssetProvider. The AssetProvider should expose a method getLink(damNode) --> String that return the correct Link based on the source storage (dam workspace, webdav...)

API review

DamAsset ? should we keep this class?

Should we create an equivalent of DMS document?

Should we remove or deprecated all STK dam package?

STKImagingSupport should use only STKTemplatingFunction to create Link for Binary content and not TemplatingFunction.

In STK for model class that use previously DMS, we expose a method that return the target as ContentMap. This content map is the representation of a JCR node.

How to handle external source ?

Should we return an Asset object instead? --> review and migrate FTL's

- Taken approach : Deprecate old methods. Expose a new method returning an Asset.

STK Model still expose method like getDMSDocument. Should we rename these methods --> review and migrate FTL's.

Actions

type key summary assignee reporter priority status resolution created updated due

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