

Adobe/SHRSS KT Session – DAM Training & Usage Guide for Admins

Adobe/SHRSS KT Session: DAM Training & Usage Guide for Admins

Jira: AAEMDAM-3736 — DAM Training & Usage Guide for Admins

Primary audience: SHRSS DAM architect

Secondary audience: AEM Sites/Assets authors, AEM developers/TAs

Authoritative references used

- [Organize your digital assets](#)
 - [Folder structure best practices for AEM Assets](#)
 - [Manage metadata of digital assets](#)
 - [Metadata best practices for AEM Assets](#)
 - [Manage digital assets](#)
 - [Dynamic Media best practices](#)
-

0. Session Objective & Outcomes

Objective

Understand how to properly use the DAM (AEM Assets as a Cloud Service) so that **admins** can manage assets, maintain governance, and support property/brand teams without risking misuse or disorganization.

This session focuses on **Assets/DAM**; tagging/taxonomy and Content Fragments are covered in detail in the separate **Tagging, Taxonomy & Metadata Governance** guide.

By the end of the session, the DAM architect and admins should be able to:

- Explain **how the SHRSS DAM is currently structured**, and where each team should store assets.
 - Describe the **target-state DAM architecture** and a phased roadmap to get there.
 - Understand **which metadata schemas/profiles** are in use and how they support search, filters, and governance.
 - Confidently perform **core DAM operations** (upload, move, rename, version, publish) without breaking references.
 - Understand the **current Dynamic Media state** and an opinionated plan for adoption across Sites.
 - Identify **follow-up discovery topics** for a new round of enhancements/fixes.
-

0.1 How to use this document in the session

- Use this document as the **live agenda and whiteboard** during the call.
 - Move around based on dialog, but try to **touch each main section**:
 - 1 – Content Architecture
 - 2 – Metadata
 - 3 – Operations
 - 4 – Dynamic Media
 - For each subsection:
 - First describe the **concept/best practice**.
 - Then discuss **how it works today at SHRSS**.
 - Capture **gaps / issues / ideas** inline (Confluence comments or live notes).
 - Treat all sections labeled "**Discovery prompts**" as seeds for **follow-on requirements sessions**.
-

1. DAM Content Architecture

1.1 Current-state DAM architecture (SHRSS)

Host prep: Fill in this subsection with actual paths and screenshots before the session.

Key questions to answer for the room:

- What are the **primary DAM roots** for SHRSS/Hard Rock?
 - Example placeholders (replace with actual paths):
 - /<insert DAM root for hardrock.com, e.g., /content/dam/hardrock>

- /<insert DAM root for reverb.hardrock.com>
 - /<insert DAM root for careers site, e.g.,
/content/dam/shrss/careers>
- How do these DAM roots map to the **Sites** currently in production or UAT?
 - <https://www.hardrock.com/> → <insert primary DAM path(s)>
 - <https://reverb.hardrock.com/> → <insert primary DAM path(s)>
 - <https://aem.careers.stage.hardrock.com/> → <insert primary DAM path(s)>
 - How is the DAM currently **segmented?**
 - By brand (Hard Rock, Reverb, etc.)
 - By line of business (Hotels, Cafes, Casino, etc.)
 - By region or property
 - By asset type (Photography, Logos, Documents, Video, etc.)

Walkthrough (live demo suggestion)

In Assets → Files:

1. Navigate to <insert main DAM root> and **walk the tree**:
 - Top-level folders (e.g. brand, properties, campaigns, shared, temp).
 - Where **site-specific** vs **shared/global** assets live.
2. Call out **any patterns**:
 - Are property-specific images under a consistent pattern (e.g. /properties/<city>)?
 - Are there “project folders” mixed with “evergreen folders”?
3. Highlight any **pain points** the team already feels:
 - Large “junk drawer” folders with 1,000+ assets.
 - Deep folder nesting.
 - Duplicate folders with similar names.

Anchor back to best practices: AEM supports both folder-based and metadata-based organization; folder structure should be the **stable backbone**, not the only way to find assets
[Organize your digital assets, Folder structure best practices for AEM Assets.](#)

1.2 What “good” looks like – Target-state DAM architecture

Principles (opinionated):

- **Stable, shallow folder hierarchies.**
 - Users should be able to guess where to put an asset in **3–4 clicks**, not 8–10.
 - Avoid putting more than ~1,000 assets per folder for usability/performance [Folder structure best practices for AEM Assets](#).
- **Separate long-lived structure from short-lived projects.**
 - Evergreen folders for: Brand, Properties, Products, Common UI, Iconography.
 - Separate project or campaign folders that can be archived later.
- **One source of truth for an asset.**
 - Each asset should live in **one canonical folder**; reuse is driven by **metadata and tags**, not copies.
- **Align DAM to how content is consumed.**
 - Align high-level folders with **how Sites and marketing teams think**: brand → property → channel or region → asset type.

Example target folder model (to adapt for SHRSS)

| Host: adapt this to the actual SHRSS structure and use actual names.

- /content/dam/shrss/brand/<brand-name>/...
- /content/dam/shrss/properties/<property-name>/...
- /content/dam/shrss/shared/global/ (logos, iconography, shared backgrounds)
- /content/dam/shrss/campaigns/<year>/<campaign-name>/
- /content/dam/shrss/people/ (HR/careers imagery, headshots, etc.)

1.3 Roadmap from current → target state

For this customer, avoid a “big bang” re-org. Propose a phased approach:

1. Baseline & freeze

- Identify a **baseline folder set** (roots that should be stable) and communicate: “New assets for X must go into <folder>.”
- Avoid large restructures until we have a clear map and rollback plan.

2. Design target architecture

- Draft a **DAM architecture diagram** showing:
 - Brand folders
 - Property folders
 - Shared/global folders
- Validate with **marketing + content teams**.

3. Define move rules

- Which existing folders are "correct as-is"?
- Which need to be merged / renamed / archived?
- Where will **new content** go immediately (even before migration)?

4. Iterative clean-up

- Choose one area (e.g., **Careers**) and:
 - Create target folders.
 - Move a **small, high-value subset** of assets.
 - Fix broken references if needed.
- Use this as a template for other areas.

5. Governance

- Document who can:
 - Create new top-level folders.
 - Create property/brand folders.
 - Create project/campaign folders.
 - Tie this to **group permissions**.
-

1.4 Content architecture best practices & anti-patterns

Best practices ([Organize your digital assets](#), [Folder structure best practices for AEM Assets](#)):

- Use folders to **impose a consistent storage structure**, not to encode every dimension of metadata.
- Keep **hierarchies shallow**, prefer metadata/tags for deep categorization.
- Use **descriptive, human-friendly names** for folders; keep system node names stable.
- Design **baseline folders** and reuse them across environments (dev/stage/prod) for consistency.
- Place assets in **leaf folders**, not parent "category" folders that also contain assets.

Anti-patterns to call out:

- "Everything under /content/dam/shrss/misc" or similar junk drawers.
 - Deep nesting like /properties/us/new-york/times-square/hotel/2024/summer/campaign1/social/instagram/....
 - Duplicating assets across many folders instead of using tags/metadata and search.
 - Authors creating top-level folders as needed ("everyone can create everything").
-

1.5 Discovery prompts (capture during session)

Use these to drive follow-on requirements:

- Which **teams** (by role) are using the DAM today? Which will use it for the upcoming careers site?
 - What are the **biggest complaints** today about finding or storing assets?
 - Are there **compliance or legal** requirements (rights, retention) that should shape folder structure?
 - Do we need separate **zones** for:
 - Internal-only assets vs public assets
 - Work-in-progress vs approved/published assets
 - Are there any **external systems** (Workfront, product catalog, HRIS) that drive how assets should be organized?
-

2. Assets Metadata

2.1 What metadata is and why it matters

Concept recap

- In AEM Assets, metadata is primarily stored on each asset under `jcr:content/metadata`.
- It powers:
 - **Search / filters / facets**
 - **Automation and workflows**
 - **Governance** (e.g., rights, expiration)
 - **Personalization and content reuse**

Metadata is the **backbone of structured content** in the DAM [Manage metadata of digital assets](#), [Metadata best practices for AEM Assets](#).

Tie to SHRSS:

- The Tagging & Taxonomy guide already documents tag-based fields like **Brand**, **Property**, **Event categories**, etc.
 - This session should focus on:
 - **Which metadata fields are required** for SHRSS assets.
 - How those fields connect to **search, filters, and downstream usage** (e.g., Sites components, external channels).
-

2.2 Current metadata schemas & profiles (SHRSS)

Host prep: Confirm these paths/definitions and paste in screenshots where helpful.

Typical locations (based on the tagging guide and AEM patterns):

- **Admin UI metadata schema** for SHRSS assets:
 - /<insert path, e.g., /conf/global/settings/dam/adminui-extension/metadataschema/shrssmetadataschema>
- **Additional schemas** (if any):
 - /<insert path for specific folders or asset types>
- **Metadata profiles** (default metadata values applied to folders):
 - /<insert path under Tools → Assets → Metadata Profiles>

During session:

1. Navigate to **Tools → Assets → Metadata Schemas**.
2. Open <insert primary SHRSS metadata schema> and walk the tabs:
 - Standard fields (Title, Description, etc.).
 - SHRSS-specific fields (Brand, Property, LOB, Product, etc.).
3. For each field, explain:
 - What authors should enter.
 - Whether it's **required vs optional**.
 - Where it is used (search, filters, components, reporting, rights).
4. If metadata profiles are used:
 - Navigate to **Tools → Assets → Metadata Profiles** and review any profiles applied to key folders.

- Call out default values (e.g., Brand, Channel) that are automatically set for assets uploaded under those folders [Metadata profiles](#).
-

2.3 Folder-level metadata schemas & application

Why this matters

- You can apply **different metadata schemas** to different folders when that area has special requirements:
 - Careers images needing additional fields (role, department, location).
 - Property-specific assets needing property ID or region.
- This reduces UI clutter for authors while still capturing necessary detail.

Walkthrough

1. In **Assets → Files**, select a folder such as

/<insert example: /content/dam/shrss/careers>

2. Open **Properties → Metadata / Cloud Services** tab and review:

- Which **metadata schema form** is applied.
- Which **metadata profile** is applied.

3. Discuss whether that is **aligned** with how the folder is being used.
-

2.4 Metadata governance – required fields & responsibilities

Opinionated recommendation:

Define and document at least the following:

- **Global required fields** (all assets)
 - Title, Description
 - Brand (tag/field)
 - Asset type / content type
 - Usage rights / license
 - Expiration or review date (if relevant)
- **Per-domain required fields** (examples)
 - Property-specific assets → Property (tag), Region
 - Careers → Location, Department, Role level

- Campaigns → Campaign ID, Campaign name

Roles & responsibilities:

- **DAM architect / admin**
 - Owns the **metadata model** and schema changes.
 - Approves new global fields and ensures backward compatibility.
- **Business owners (Brand, HR, Properties)**
 - Define **which metadata is required** for their domain.
 - Help maintain value lists (e.g., departments, property names).
- **Authors**
 - Are responsible for **populating metadata** correctly when uploading or editing assets.

Tie back to [Metadata best practices for AEM Assets](#):

Metadata should be **intentional** and **governed**, not an ad-hoc collection of fields.

2.5 Best practices & anti-patterns

Best practices

- Start with a **minimal but meaningful set** of required fields; expand gradually.
- Use **consistent naming** and clear help text so authors know what to enter.
- Use **tag fields** for taxonomy-like values (Brand, Property, Region, Category).
- Use **typed fields** (date, number, boolean) where appropriate for validation and automation.
- Leverage **metadata import/export** for bulk updates when cleaning or backfilling metadata

[Manage metadata of digital assets](#).

Anti-patterns

- Dozens of rarely used fields; authors skip them or enter junk data.
- Duplicating the same concept across multiple fields (e.g., Brand as both a free text field and a tag).
- Letting every team add their own fields without central review.
- Using tags for things that should be fixed/enumerated fields (e.g., license ID).

2.6 Discovery prompts

Questions to uncover requirements and gaps:

- Which **fields do authors actually use** when searching for assets?
 - Which fields are **critical** for compliance (rights, retention)?
 - Are there **integrations** (e.g., Workfront, HR systems) that should push metadata into AEM?
 - Where is metadata **incomplete or inconsistent** today?
 - Do we need **additional metadata schemas** for new areas (e.g., careers site, future properties)?
-

3. DAM Operations

3.1 Uploading & organizing assets

Use this section to align on **how admins and authors should actually work in the DAM**.

Key flows to demonstrate ([Upload assets to the repository / Assets view](#), [Manage digital assets](#)):

- Uploading:
 - Drag-and-drop vs **Add Assets** button.
 - Uploading individual assets vs full folder hierarchies.
 - Where to upload for:
 - Hard Rock main site
 - Reverb
 - Careers
- Organizing:
 - Creating folders (and **who is allowed** to create top-level vs nested folders).
 - Using **folder templates or naming conventions**:
 - YYYY-MM_<CampaignName>
 - property-name/content-type
- Bulk operations:
 - Multi-select for move/copy.
 - Using **Extract ZIP** (if available) for hierarchical uploads.

Opinionated guidance:

- Only **DAM admins or designated power users** should create top-level or brand/property root folders.

- Standard authors create folders **only within their assigned area**.
 - Avoid uploading to generic or “root” folders; always place into the **nearest appropriate leaf**.
-

3.2 Versioning & lifecycle

Concept

- AEM Assets automatically maintains **versions** on edits and uploads.
- Versions are helpful for:
 - Rolling back accidental changes.
 - Tracking approval history.

Demonstrate

1. Select an asset and open **Timeline / Versions**.
2. Show:
 - Previous versions.
 - Reverting to an earlier version.

Recommend a simple lifecycle model:

- Upload as **Draft** (optionally controlled through a status field or workflow).
 - Review/approve → **Approved**.
 - Publish to Sites or external channels.
 - On replacement:
 - Upload a **new version**, not a new asset in a different location, unless there's a business reason.
 - Consider capturing **changelog notes** in the Version label/comment.
-

3.3 Moving, renaming, and avoiding broken references

This is one of the highest-risk behavior areas; be explicit and opinionated.

How references work (high level)

- Pages, Content Fragments, and components store **paths** to assets.
- If an asset is **moved or renamed**, AEM can update references in many cases, but:
 - Cross-environment or external consumers may not be updated.
 - There is still risk of confusion, especially if authors have bookmarks or direct URLs.

Guidance for admins/authors:

- **Do not move or rename** assets that are already widely used **without coordination**.
- If you must reorganize:
 - Prefer moving entire **folder segments** during low-traffic windows.
 - Use AEM's "**Move**" operation (not delete + re-upload) so references can be updated.
 - Republish affected assets and, if needed, pages.

Tie this back to the folder-architecture roadmap: if major moves are required, plan them as a **governed project**, not ad-hoc changes.

3.4 Publishing & how updates flow to live pages

Explain & demonstrate

- When an asset is updated and **re-published**, any page that references that asset will **automatically show the new rendition** after cache/CDN refresh.
- For the group:
 - Show a page on author that uses an image.
 - Swap the image with a new **version** (same asset).
 - Publish the asset (and page, if needed).
 - Show that nothing needs to change in the component configuration.

Important behaviors to highlight:

- **Updating the asset binary** (new version) → safe; page references remain valid.
- **Moving the asset** → can update references, but still a risk.
- **Deleting the asset** → breaks references; components will show missing images.

Recommended practice

- **Never delete** assets that may still be referenced, unless:
 - You have verified references (and these references are intentionally being removed).
 - You have an archival strategy (e.g., move to an **Archive** folder and remove publish).

3.5 Workflows & approvals (if present)

Host prep: note any existing SHRSS workflows; if none exist, treat this as a design discussion.

If there are asset workflows in place:

- Review:
 - Who can upload and move assets.
 - Who must **approve** assets before publish (if any).
 - Any **automated watermarking, rendition generation, or metadata extraction**.

If there are **no workflows** yet:

- Suggest lightweight options:
 - Simple **review** step for brand-critical folders (logos, hero imagery).
 - Automated **metadata enrichment** (e.g., Smart Tags, XMP extraction) where appropriate.
-

3.6 Operational runbook anchors

Tie back to your **DAM Ops Runbook** handout. Highlight:

- **Daily/Weekly:**
 - Review failed uploads/rejections.
 - Spot-check new folders/metadata for consistency.
 - **Monthly/Quarterly:**
 - Audit **large folders** and reorganize if needed.
 - Review **orphaned assets** (no references).
 - **Ad-hoc:**
 - Support property/brand teams in planning campaigns that need new assets and metadata fields.
-

3.7 Discovery prompts

Capture answers live:

- Where do admins feel **most nervous**: moving, deleting, or publishing assets?
- Are there known cases of **broken images** or files on live sites due to DAM changes?
- Do they need **reporting** on:
 - Most-used assets
 - Unused/orphaned assets
 - Assets nearing rights expiration?

- Would **formal workflows** for certain folders help, or slow teams down?
-

4. Dynamic Media

4.1 Current state (SHRSS, as of this session)

From the agenda:

- Dynamic Media is:
 - **Provisioned** for the tenant.
 - **Configured in lower environments** (dev/stage).
 - **Not configured in production** yet.
 - Sites pages and experience fragments **do not currently use Dynamic Media components**.

Host prep: Fill in specific details:

- DM company name / configuration ID (if known).
- Which environments have DM set up and tested.
- Any pilot components or pages already using DM in lower tiers.

4.2 What changes with Dynamic Media for authors/admins

Dynamic Media primarily changes:

- How **images and videos are rendered and delivered** (via DM URLs).
- How authors choose **presets** (image profiles, video profiles).
- The ability to:
 - Automatically serve **responsive images** and **web-optimized renditions**.
 - Use **viewers, zoom, spin sets**, etc.

From [Dynamic Media best practices](#):

- Organize assets in clear folders and apply consistent metadata; this is even more critical when scaling DM.
- Use **processing profiles** and **image presets** to standardize how assets are rendered.

Explain in business terms:

- DM is a **delivery engine**, not a separate DAM:

- Admins manage assets as usual in AEM.
 - DM handles **fast image delivery, resizing, and format optimization**.
-

4.3 Integration patterns for SHRSS Sites

Host prep: Identify which components will be switched to DM-enabled versions (core components or custom ones).

Discuss how Dynamic Media will be used on:

- **hardrock.com:**

- Hero images, banners, galleries.
- High-value imagery for properties and brand experiences.

- **reverb.hardrock.com:**

- Product imagery, if applicable.

- **careers site:**

- Candidate-focused imagery, location/office photos, possibly video.

Patterns to consider:

- Updating image components to use **DM-enabled variants** (e.g., AEM Core Image component with DM).
 - Using **DM Smart Imaging** for automatic format/quality tuning.
 - Defining **standard image presets** (e.g., hero, card-thumb, gallery-large) and mapping them to components.
-

4.4 Dynamic Media adoption roadmap

Propose a staged rollout to reduce risk:

1. Enable and validate in Stage

- Confirm DM is fully configured and working in stage.
- Validate key use cases:
 - Responsive hero images
 - Card thumbnails

- Basic video playback (if in scope)

2. Select a pilot area

- For example, **Careers site** or a limited subset of hardrock.com pages.
- Switch those components to DM and monitor performance, image quality, SEO impact.

3. Define governance

- Who is allowed to:
 - Create new image/video presets.
 - Change DM configuration.
- Document **naming conventions** for DM presets.

4. Rollout to additional areas

- After a successful pilot, expand by brand or property.
- Plan communication/training so authors know what changes (if anything) in their workflow.

5. Measure and optimize

- Page performance (LCP, image weights).
- Author satisfaction / ease of use.

4.5 Dynamic Media usage guidelines

Opinionated rules for SHRSS:

- Use DM for:
 - All **hero/above-the-fold** imagery on high-traffic pages.
 - Any imagery that must be **responsive** across devices.
 - Heavy assets like carousels and large galleries.
- Consider not using DM for:
 - Small, rarely used internal images where overhead isn't worth it.
 - Edge cases where assets are downloaded rather than rendered inline (e.g., PDFs).
- Ensure:
 - **Original/master** files are stored in DAM; DM renditions are derived.

- Presets are **centrally governed**; don't let every team create their own presets without review.
-

4.6 Dynamic Media discovery prompts

Use these to set up a dedicated DM design session:

- Which **pages and experiences** are most impacted by performance and image quality today?
 - Does the organization have **image/brand guidelines** that should drive DM presets (ratios, focal points)?
 - Are there **third-party integrations** (e.g., legacy image servers) that DM will replace?
 - Who should own **preset definitions** (brand vs web team vs DAM architect)?
-

5. Next Steps & Follow-on Sessions

Use this section in the last 5–10 minutes to align.

5.1 Immediate actions (post-session)

- **Fill in all placeholders** in this guide with:
 - Actual DAM root paths and folder examples.
 - Exact metadata schema names & key fields.
 - Dynamic Media configuration details and pilot areas.
- Circulate:
 - This guide (Confluence link) to DAM admins and key authors.
 - The **DAM Ops Runbook** and **Author Do/Don't guide** as companion documents.

5.2 Proposed follow-on sessions

Recommend at least:

1. DAM Architecture & Migration Workshop

- Deep dive into current vs target folder structure.
- Plan phased reorganization.

2. Metadata & Governance Workshop

- Finalize required fields per domain (Brand, Properties, Careers).
- Define ownership and processes for metadata quality.

3. Dynamic Media Design Session

- Define image/video presets, pilot scope, and rollout plan.
- Confirm component-level integration approach.

4. Author-focused Hands-on Training

- For content authors across brands/properties:
 - How to find assets.
 - How to apply metadata correctly.
 - How to use DM-enabled components (once in place).

Appendix A: Placeholders to fill in (host checklist)

Before or after the session, update this document with:

- **DAM roots / key folders**
 - <insert path> for:
 - Hard Rock
 - Reverb
 - Careers
- **Example asset paths**
 - A representative hero image, card image, and property image.
- **Metadata schema(s)**
 - Name + path of the main SHRSS metadata schema.
 - Any specialized schemas (e.g., careers, campaigns).
- **Metadata profiles**
 - Names of profiles and which folders they are applied to.
- **Dynamic Media config**
 - DM company & environment details (if shareable).

- Target pilot pages and components.
 - **Screenshots**
 - DAM folder tree for each site.
 - Metadata form for a typical SHRSS asset.
 - Example of a page using DAM assets (for illustration).
-

Appendix B: Quick reference to AEM documentation

For convenience, here are the key docs used in this guide:

- [Organize your digital assets](#)
- [Folder structure best practices for AEM Assets](#)
- [Manage metadata of digital assets](#)
- [Metadata best practices for AEM Assets](#)
- [Manage digital assets](#)
- [Dynamic Media best practices](#)

Use this guide as the primary live artifact in the session, then treat it as a **living governance document** in Confluence as SHRSS iterates on their DAM implementation.