

Asset Deletion 3 - Deletion process

The deletion process of Asset Deletion 3 consists of three steps: mark asset version for deletion, delete the marked asset versions and delete the asset files. This article explains the process and then each step in detail.

Context

- Understanding the deletion process of Asset Deletion 3 is the base for configuring the server actions and the monitoring of the deletion process on the server.
- It helps to understand how asset version policies are executed in the deletion process. This is useful when configuring asset version policies or executing reports to see why certain asset versions cannot be removed.
- Asset deletion 3 is the successor of Asset Deletion 1 and 2.

Prerequisites

- Common knowledge of the technical setup and configuration of the censhare Server and the basic principles of relational databases.

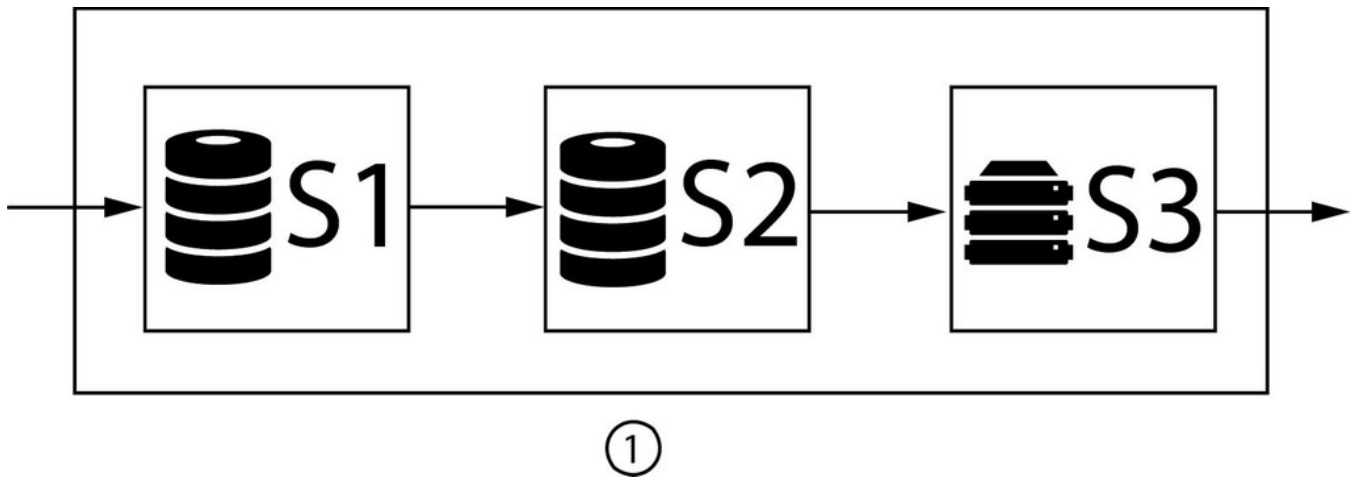
Introduction

Asset Deletion 3 removes asset versions that are no longer needed. Asset files that are related to deleted asset versions are deleted, too. The asset deletion process defines how this is done. The deletion process is a three-stage process:

- Mark an asset version for deletion.
- Delete the marked asset version physically.
- Delete related asset files.

In each stage, constraint checks are carried out.

For performance reasons, the three stages of the deletion process are not done consecutively on single assets. Instead, each stage processes assets during a defined period, then stops and proceeds with the next stage.



The asset deletion process (1) walks through three parts one after another: **Mark asset versions (S1)**, **Delete asset versions (S2)** and **Delete storage items (S3)**.

The asset deletion process was designed for large databases with a great number of assets, asset versions, and asset files:

- Each part of the execution process has a limited execution time: The deletion process does not run until all assets are processed.
- All assets are processed before the deletion process starts from the beginning. This might require several execution times.

This allows running deletion process during a defined maintenance window on large systems.

Overview

The deletion stages

1. Mark asset versions for deletion
2. Delete asset versions
3. Delete storage items

Note: The stages can be executed independently.

Stage 1: Mark asset versions for deletion

In this stage, censhare walks through the assets and marks them for deletion. It only processes asset versions of an asset that have no constraints on asset or asset version level. If there is no asset version policy for an asset, it is also skipped.

censhare then checks the constraints defined by the selected asset version rule. Typically, an asset version policy keeps the first X and the last Y versions. The versions of an asset are marked for deletion that does not apply to any constraint.

Stage 2: Delete asset versions

In this stage, asset versions are processed that are marked for deletion. It checks if there is an asset version policy that still applies. If not, the mark for deletion flag stays.

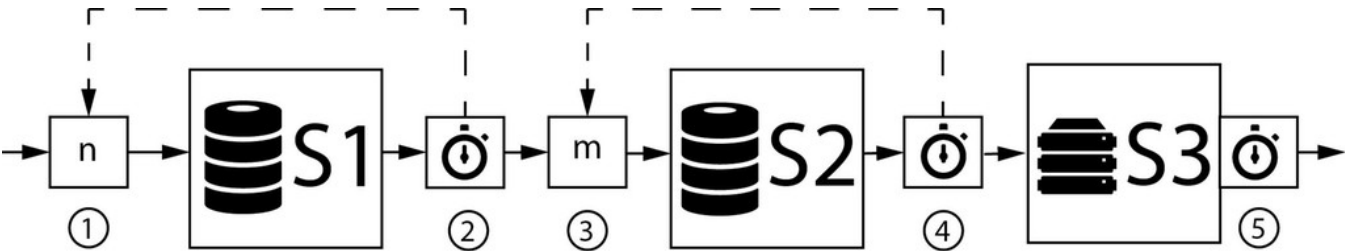
Every asset version policy defines a period that has to expire before the asset version is physically deleted. If this is not true, the asset version is still kept.

The remaining asset versions are deleted from the database table. If a removed asset version has an asset file, censhare adds this file to the deletion queue in the database. This queue is processed in the third stage.

Stage 3: Delete storage items

In this stage, censhare processes the deletion queue and removes the asset files physically from the file system. This stage also includes constraints checks.

Deletion process flow



The deletion process flow has three parts: **Mark for deletion (S1)**, **Delete asset versions (S2)** and **Delete storage items (S3)**.

S1 marks the asset version until the time limit (2) is over. Then, S2 deletes asset versions until its time limit (4) is over. Then, S3 delete asset files until the time limit (5) is over.

When S1 is finished, it stores the last processed Asset ID in the variable n (1). The next time, S1 starts with the asset ID stored in n (1). When S2 is finished, it stores the last processed Asset ID in the variable m (1). The next time, S2 starts with the asset ID stored in m (3).

S1 and S2 are processing batch by batch (default batch is 10.000 versions). If the time limit (2 or 5) is reached, the deletion process continues until it finishes the current batch. This might take some additional minutes.

If all three stages are configured, censhare executes them consecutively. The server action can be configured to execute all stages or single stages only. The manual server action executes the deletion process on selected assets. The automatic server action is triggered by a cron event.

Each deletion stage runs for a defined time. This allows that the deletion process only runs for the length of the maintenance window. If a deletion stage finishes before that time, the next deletion stage starts respective finishes earlier. The execution times are configured in the Admin Client.

If not all assets are processed during the execution time, censhare keeps the last processed asset ROWID and starts from that ROWID at the next execution period. ROWIDs are generated by the database and returned when censhare queries for assets in the asset deletion process. If all asset ROWIDs are processed before the defined execution is reached, the stored ROWID (mark or deletion) is set to 0. So, the run of the mark/deletion stage needed less time than the defined execution time.

The stored ID can be reset by an administrator.

Storing the last execution state

The IDs of the last processed assets (stage 1 and 2) are stored in a resource asset with the name **AD3** and the resource key **censhare:query-deletion-ids**

It stores two features for the last processed asset:

Feature	Resource key	Remark
---------	--------------	--------

AD3 last marked id	censhare:deletion.last-id.marked	Stores the last state for the Mark for deletion part.
AD3 last deleted id	censhare:deletion.last-id.deleted	Stores the last state for the Do deletion part.

Details

Stage 1: Mark for deletion

The Mark for deletion stage consists of five steps:

1. censhare queries a batch of 10.000 versions from the database, checks for constraints on asset and asset version level, and creates the first candidate list.
2. censhare queries for each asset ID all versions of the asset.
3. censhare applies the corresponding asset deletion policy to the asset versions of each asset.
4. censhare cross-checks if the filtered asset versions are in the first candidate list. If not, the asset version is skipped. If yes, the asset version is written on the final candidate list.
5. censhare processes final candidate list and marks the versions for deletion by setting the deletion attribute of the asset version to "1". For performance reasons, this step is done in batches of 120 asset versions. Only if this fails for one item in a batch, the batch is processed one by one.

Then, censhare starts to query the next batch of assets and walks through all five steps again.

Stage 2: Delete asset versions

The Delete asset versions stage consists of three steps:

1. In an SQL query, censhare fetches a batch of 10.000 asset versions (number cannot be changed) that are marked for deletion from the database. censhare checks if a valid asset version policy still exists and creates a first candidate list with asset versions to be deleted. (1)
2. censhare applies the **Keep x hours for deletion** setting of the policy. If the period is over, the asset version is deleted physically. Otherwise, a version is skipped, but remains as "marked for deletion".
3. censhare deletes batches of 120 asset versions physically in one transaction from the database. If the transaction for a batch fails, the versions are deleted one by one

(1) Note: If an asset version policy changed in the meantime, the new policy does not apply to versions that are already marked for deletion.

Stage 3: Delete storage items

In this stage, the asset files from the **storage_deletion_queue** table in the database are processed and deleted if possible. This stage is not done in batches, the table entries are processed one by one..

The part consists of two steps:

1. Check the **storage_deletion_queue** table.
2. Select the entries from the table and delete the files.

censhare checks in the **storage_item** table if an entry references an asset file. This asset file is skipped.

The asset file is also not deleted If the **state** field of the table entry has the following values:

- 0 (available online)
- 1 (offline / archived)

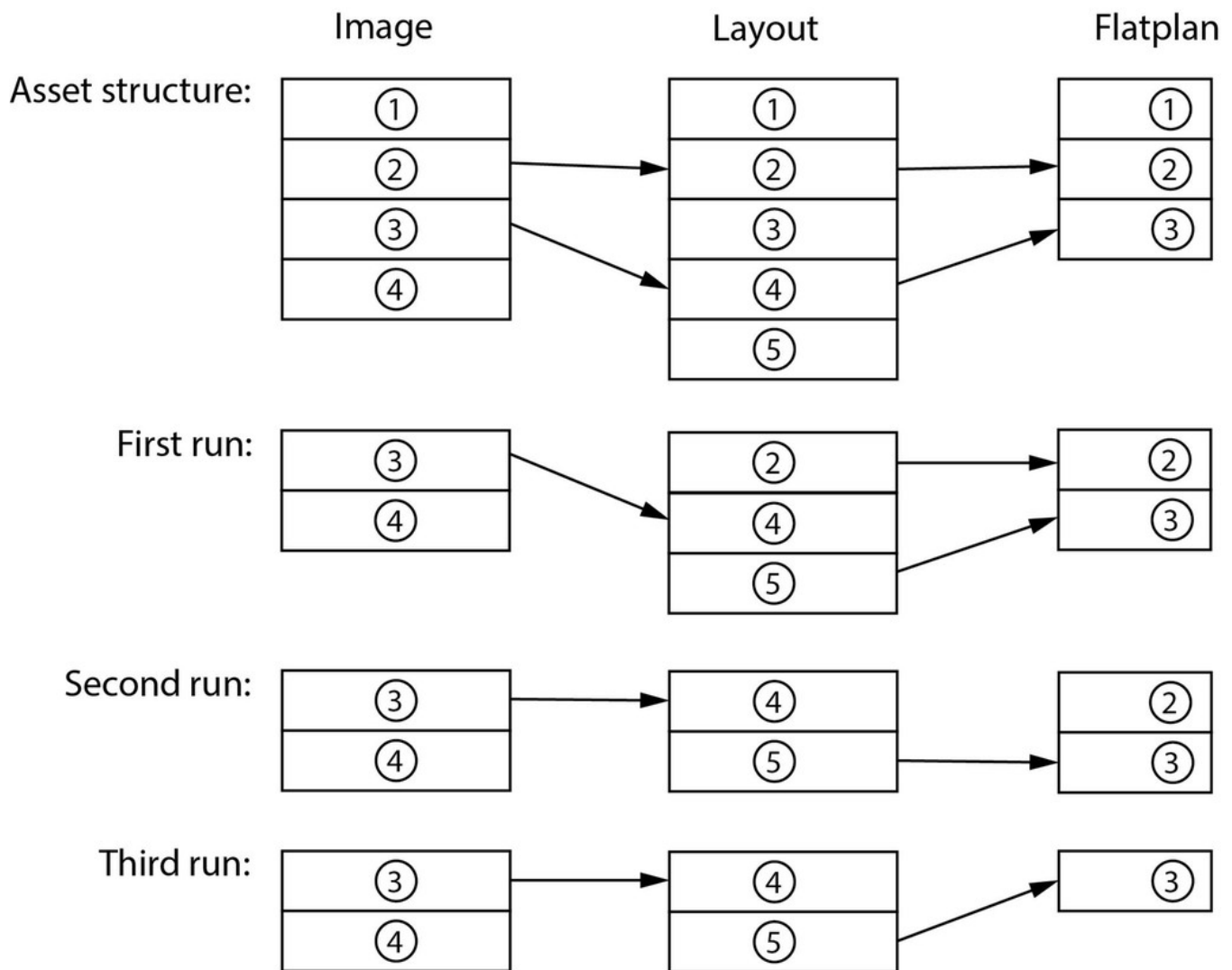
Deleting asset structures

Deleting versions in assets that are part of an asset structure cannot be done in one run. For example, an asset structure contains Flatplan, Layouts and Images with relations: between flatplan asset versions and layout asset versions and between layout asset versions and image asset versions.

A flatplan version can only be deleted if there is no relation to a Layout version. Layout versions can only be deleted if there are no relations to image versions. This has to be checked before marking a flatplan asset version for deletion. This requires to follow all these relations. For large asset structures, the checks require a huge effort.

Therefore, Asset Deletion 3 processes asset structures level by level. It only checks asset versions of direct parents and children. The next level (grand-parents and grand-children of the original asset) are processed in the next run of the deletion process and so on.

The following diagram illustrates this:



The asset structure with an image asset, a layout asset and a flatplan is processed in three runs.

There is an image asset with 4 versions, a layout asset with 5 versions and a flatplan with 3 version and several relations:

1. In the first run, all possible versions of the image are removed. This includes version 2 of the image because there is a younger version 3 that also relates to the layout. From layout, version 1 and 3 can be deleted because there is no policy against it. The same is for version 1 of the flatplan.
2. In the second run also version 2 of the layout is deleted because there is a younger version 5. Version 2 of the flatplan has no longer a relation that prevents it from being deleted.
3. In the third round, version 2 of the flatplan is finally deleted.

Change of policies while the deletion process

Case 1: Change selection conditions after "Mark for deletion"

Situation: An asset version has been marked for deletion. Now, the respective asset version rule changes: For example, there is a new domain or asset type policy. The rule does not apply anymore to the asset version in question.

Question: Is the asset version still deleted when **Do deletion** is executed?

Answer: When **Do deletion** is executed, it skips all asset versions that are marked for deletion but do not have a matching asset version policy. If there is no matching policy, that asset version is not deleted. It stays in the system with a marked for deletion flag until there is a matching asset version policy.

Case 2: Change "Keep versions" parameters after "Mark for deletion"

Situation: An asset version has been marked for deletion. Now, the respective asset version rule is edited: The **Keep version** parameters are changed in a way that the asset version in question shall be kept.

Question: Is the asset version still deleted when **Do deletion** executes?

Answer: Yes, the asset version in question is still deleted when **Do deletion** is executed. **Do deletion** only checks if there is any asset version policy that applies. It does not check if the **Keep version** parameters still apply.

Steps

Reset the last execution state

If you need to reset the last execution state for the Mark for deletion step, do the following:

- 1. Open the censhare Client.
- 2. Change to the Admin mode.
- 3. Open the search window and search for the resource key **censhare:query-deletion-ids**.
- 4. Go to the **Features** tab.
- 5. Enter "0" in the **AD3 last marked id** feature.

For the Do deletion step do the same steps besides the last step:

- 1. Enter "0" in the **AD3 last deleted id** feature.

Note: You can reset only one of the features or both together.

Note: At the moment this cannot be done in censhare Web.

Reference: deletion state

In terms of deletion, an asset can have several states. The actual state of an asset is set in the attribute **deletion**:

State	Attribute deletion
not deleted	0
marked for deletion	1
cannot delete	2
physical deletion	3
planned for deletion	4
undefined	-1

Result

You understand the setup of the deletion process in Asset Deletion 3. You know the three parts of the deletion process and the steps that each part includes. You understand which parts constraints are applied. You understand the deletion process flow and how the defined time for each part influences the execution.

Normally, the parts **Mark for deletion** and **Delete asset versions** continue from the point, where they stopped the last time. You know how to reset these that one or both parts start from the beginning instead.