Technical Notes

Asset Manager

Impact of AM ID Defragment

Document Release Date: November 2015

Table of contents

Summary

The ID defragment tool is aimed to defragment the IDs of Asset Manager (AM) records when the ID number is close to reach the limit. After running the tool, the IDs in all AM tables will be regenerated and the old IDs will not be reserved in the AM database. The defragmentation process shrinks the amount of primary row IDs in all AM database tables and therefore each record ID will be updated.

This documentation lists the things that you need to pay attention to in the Asset Manager database before and after an ID defragmentation. This documentation also lists the impact on integrations with external systems including UCMDB AM push adapter and AM generic adapter.

Asset Manager Database

Mapping tables

During the ID defragmentation, Asset Manager iddefrag process creates mapping tables, either by adding a column to existing tables or by creating _M mapping tables.

For example, amAsset will have a new mapping table named amAsset_M that has the following two columns.

- lAstId: ID of the primary key in Asset Manager (lastid for amAsset). Note that this field will be named lEmplDeptId for the mapping table amEmplDept_M).
- NewIDGen: New ID of the primary key. After the ID defragmentation, for records that had amAsset.lAstId= amAsset_M.lAstId before the ID defragmentation, the value of amAsset.lAstId will be replaced with this value.

Scripts, actions, and access rights restrictions

After the ID defragmentation, Asset Manager scripts that use non-ID keys to identify records (fields like Barcode, AssetTag, SQLName, UD identifiers, Serial Numbers, and so on) then obtain the IDs from the matching records. They can still work without any adjustments.

Scripts, queries, actions, and all code that is based on hard-coded IDs will require certain adjustments after the ID defragmentation. This potentially impacts all queries, scripts, and other actions (emails, wizards, and so on) depending on how they are tailored.

Queries

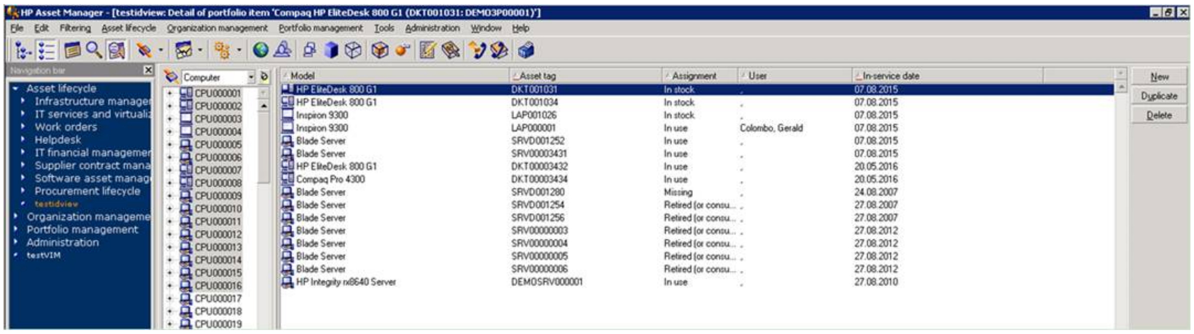
To prevent queries from being invalidated by the ID defragmentation, you need to modify the hard-coded IDs in your queries (amQuery).

All those queries will have to be altered to query either other unique identifier keys or use the new IDs. In particular, for SLO Counters, queries to select items (scope or counts models) for a SLO counter will have to be checked and modified if they use IDs to determine the license or inventory scope.

Views

To prevent views from being invalidated by the ID defragmentation, you need to make sure that:

- Initially, views can be created with filters set on IDs.
- View definitions are saved in a specific table that is not updated during the ID defragment process.
- Validation on views must be done manually after the ID defragment.
- The visible way to filter records based on ID can be:
    - In the ‘Simple filter’, select the ID field as the condition.
    - In the ‘Query filter’, write the script by filter ID fields directly.
- Views can also implicitly use ID filter. In the ‘Simple filter’, select the Link as the condition, then in the displayed records, select the ones you want to make a filter, as shown in the following figure.



HP Asset Manager - [testidview: Detail of portfolio item 'Compaq HP EliteDesk 800 G1 (DKT001031: DEMO3P00001)']

Model	Asset tag	Assignment	User	In-service date
HP EliteDesk 800 G1	DKT001031	In stock	.	07.08.2015
HP EliteDesk 800 G1	DKT001034	In stock	.	07.08.2015
Inspiron 9300	LAP001026	In stock	.	07.08.2015
Inspiron 9300	LAP000001	In use	Colombo, Gerald	07.08.2015
Blade Server	SRVD001252	In use	.	07.08.2015
Blade Server	SRV00003431	In use	.	07.08.2015
HP EliteDesk 800 G1	DKT00003432	In use	.	20.05.2016
Compaq Pro 4300	DKT00003434	In use	.	20.05.2016
Blade Server	SRVD001280	Missing	.	24.08.2007
Blade Server	SRVD001254	Retired (or consu...	.	27.08.2007
Blade Server	SRVD001256	Retired (or consu...	.	27.08.2007
Blade Server	SRV00000003	Retired (or consu...	.	27.08.2012
Blade Server	SRV00000004	Retired (or consu...	.	27.08.2012
Blade Server	SRV00000005	Retired (or consu...	.	27.08.2012
Blade Server	SRV00000006	Retired (or consu...	.	27.08.2012
HP Integrity rx8640 Server	DEMOSRV000001	In use	.	27.08.2010

UCMDB AM push adapter

As the AM Push Adapter caches AM IDs in the UCMDB database, after the ID defragment tool is run, the cached AM IDs become invalid. The cache must be cleaned up and re-built with the new AM IDs, otherwise, the invalid IDs will cause failure of data push and unexpected errors.

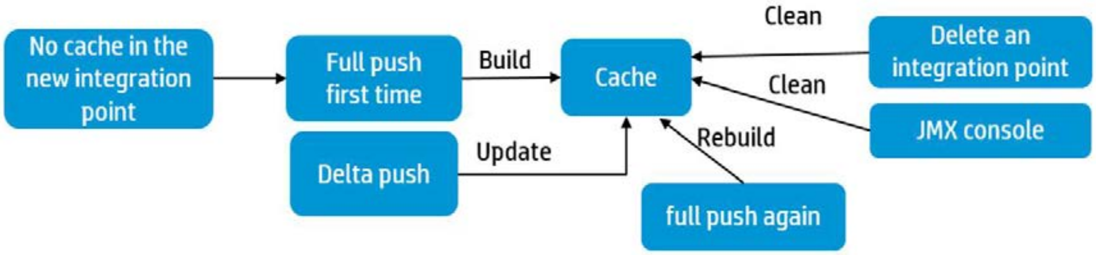
How Asset Manager IDs are cached in the push adapter

Asset Manager IDs are cached in the push adapter with the following procedure.

- UCMDB queries the CIs by using TQL statements.
- The AM push adapter transforms and pushes CIs to AM. The ID of the record that matches the pushed CI in the AM database is pushed back to the adapter and stored in an ExternalId object. The ExternalId object contains the CI information and the ID of the corresponding record in AM.
- After the push job is completely executed, the ExternalId objects of all pushed CIs are cached in the UCMDB database, those Asset Manager record IDs are stored in the UCMDB table sync_ids_map.

Lifecycle of cached data

The cached data is generated when running a full push. It is kept throughout the push job execution and not cleaned after the job completes. The following diagram shows the lifecycle of the cached data from generation to disposition.



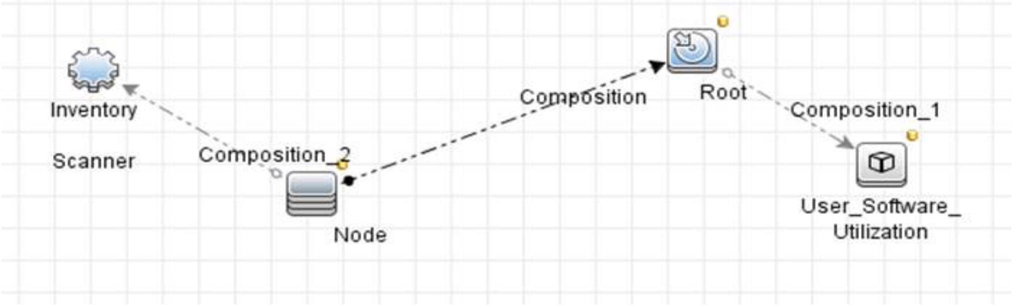
- The cache associated to an integration point is empty when the integration point is created.
- The cache is built when running a full data push for the first time.
- The cached data is updated when running delta pushes.
- The cache is re-built when running a full data push again.
- After an integration point is deleted, the cache is permanently deleted from the UCMDB database.

ID defragmentation impact on push jobs

The following two types of CIs are affected by the ID defragmentation.

- Relation CIs. The end1 CI and end2 node CI of a relation CI use the cached AM IDs to find the corresponding records in AM. Because the cached IDs have changed during ID defragmentation, a Relation CI push will fail if it is run right after the ID defragmentation. A full push has to be run on all Node CIs referenced as end1 or end2 in relation CIs first.

- Node root CIs that depend on other CIs. When pushing the root CIs, it will use the cached AM ID of the dependent CIs to locate the AM record of the dependent CIs in the AM database. Because the cached AM ID of the dependent CIs are not valid after running ID defragmentation, the root CI push will fail.



For instance, in the AM Software Non-Norm TQL, the root node is software installation and it depends on the Computer node. After running the ID defragment tool, the cached AM IDs for computers will become invalid. If software installations are pushed directly after an ID defragmentation, the push will fail (until computers and other dependent Cis are pushed with a full push).

Actions to perform after an ID defragmentation

For the two CI push failure types described in the *ID defragmentation impact on push jobs* section, the dependent CI must be pushed again.

- Relation CIs. The end1 and end2 node CIs must have a full push to update the cached AM IDs, but the relation CIs do not need to be re-pushed to refresh their cached AM IDs.
- Node root CIs depending on other CIs. The dependent CIs must have a full push to update the cached AM IDs, but the root CIs do not need to be re-pushed to refresh their cached AM IDs.

In the previous example of the software installation push, after the ID defragmentation, you must run a full push for the computer CIs, while it is not necessary to do the same for software installation CIs. It is possible to push software installations in either a full or delta push (both will work),

Other integrations and programs

After the ID defragmentation, all scripts or code in external applications that use non-ID keys to identify records (fields like Barcode, AssetTag, SQLName, UD identifiers, Serial Numbers, and so on) then obtain the IDs from the matching records. They can still work without any adjustments.

Scripts, queries, actions, and all code that is based on hard-coded IDs will require some adjustments after the ID defragmentation.

The adjustments are also required for integrations with other HP applications, as well as third-party applications (including Crystal Reports) interacting through the Asset Manager C++/Web Service APIs or through other means like database views, Asset Manager ODBC driver, and so on.

More information

For more information, please visit the HP Management Software support website:

https://softwaresupport.hp.com

This website provides contact information and details about the products, services, and support that HP Management Software offers.

HP Management Software online software support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued customer, you can benefit by being able to:

- Search for knowledge documents of interest
- Submit and track progress on support cases
- Submit enhancement requests online
- Download software patches
- Manage a support contract
- Look up HP support contacts
- Review information about available services
- Enter discussions with other software customers
- Research and register for software training

Note:
Most of the support areas require that you register as an HP Passport user and sign in. Many also require an active support contract.

To find more information about support access levels, go to the following website:

http://www.hp.com/managementsoftware/access_level

To register for an HP Passport ID, go to the following website:

http://www.managementsoftware.hp.com/passport-registration.html