

Archiving Datasets

- [Manual archiving](#)
 - [openBIS core UI](#)
 - [ELN-LIMS](#)
- [Automatic archiving](#)
 - [Archiving Policies](#)
 - [ch.systemsx.cisd.etlserver.plugins.GroupingPolicy](#)

Manual archiving

openBIS core UI

Archiving can be triggered by doing the following steps:

- go to an experiment/collection or an object.
 - switch to the tab "Data Sets". There will be in their lower right corner the button 'Archiving'.
 - click on the button and choose either 'Copy to Archive' or 'Move to Archive'.
 - if you did not select any data set all data sets will be archived.
- If you have selected some data sets you can choose if you want to archive only them or all the data sets accessible in the table.

Because archiving does not happens immediately the status (called 'Archiving Status' in data set tables) of the data sets will be changed to BACKUP_PENDING or ARCHIVE_PENDING.

To make archived data sets available again repeat the steps, but choose 'Unarchive'.

If you want to disallow archiving, choose 'Lock'. Remember that you can do this only for available data sets. The 'Archiving Status' will change to 'AVAILABLE (LOCKED)'. To make archiving possible again, choose 'Unlock'.

ELN-LIMS

Instead of triggering archiving only requesting archiving is possible. The maintenance task [ArchivingByRequestTask](#) is required. It triggers the actual archiving.

Automatic archiving

Archiving can be automated by the Auto Archiver. This is a [maintenance task](#) which triggers archiving of data sets fullfulling some conditions (e.g. not accessed since a while). Note that the auto archiver doesn't archives itself. It just automates the selection of data sets to be archived. For all configuration parameters see [AutoArchiverTask](#).

Archiving Policies

An archiving policy selects from the unarchived data sets candidates (which are either data sets not accessed since some days or data sets marked by a tag) the data sets to be archived. If not specified all candidates will be archived.

The policy can be specified by `policy.class` property. It has to be the fully-qualified name of a Java class implementing `ch.systemsx.cisd.etlserver.IAutoArchiverPolicy`. All properties starting with `policy.` specifying the policy further.

ch.systemsx.cisd.etlserver.plugins.GroupingPolicy

Description: Policy which tries to find a group of data sets with a total size from a specified interval. This is important in case of [Multi Data Set Archiving](#). Grouping can be defined by space, project, experiment, sample, data set type or a combination of those. Groups can be merged if they are too small. Several grouping keys can be specified.

Searching for an appropriate group of data sets for auto archiving is logged. If no group could be found an admin is notified via email (email address specified in `log.xml`). The email contains the searching log.

Configuration:

Property Key	Description
minimal-archive-size	The total size (in bytes) of the selected data sets has to be equal or the larger than this value. Default: 0
maximal-archive-size	The total size (in bytes) of the selected data sets has to be equal or the less than this value. Default: Unlimited

grouping-keys	<p>Comma separated list of grouping keys. A grouping key has the following form:</p> <pre><basic key 1>#<basic key 2>#...#<basic key n>[:merge]</pre> <p>A basic key is from the following vocabulary: All, Space, Project, Experiment, Sample, DataSetType, DataSet.</p> <ul style="list-style-type: none"> • All basic keys of a group key define a grouping of all data set candidates. In each group all data sets have the all attributes defined by the basic keys in common. Note, that basic key All means no grouping. For example: Experiment#DataSetType means that the candidates are grouped according to experiment and data set type. • The optional :merge is used when no group fulfills the total size condition and there are at least two groups with total size below minimal-archive-size. In this case groups which are too small will be merged until the total size condition is fulfilled. • If a grouping key doesn't lead to a group of data set fulfilling the total size condition the next grouping key is used until a matching group is found. • If for a grouping key more than one matching group is found the oldest one will be chosen. • If merging applies for more than two groups the oldest groups will be merged first. • The age of a group is defined by the most recent access time stamp. <p>Examples:</p> <ol style="list-style-type: none"> 1. Grouping policy by experiment: DataSetType#Experiment, DataSetType#Project, DataSetType#Experiment#Sample 2. Grouping policy by space: DataSetType#Space, DataSetType#Project:merge, DataSetType#Experiment:merge, DataSetType#Experiment#Sample:merge, DataSet:merge
---------------	---

Example:

plugin.properties	<pre>class = ch.systemsx.cisd.etlserver.plugins.AutoArchiverTask interval = 10 days archive-candidate-discoverer.class = ch.systemsx.cisd.etlserver.plugins.TagArchiveCandidateDiscoverer archive-candidate-discoverer.tags = /admin-user/archive policy.class = ch.systemsx.cisd.etlserver.plugins.GroupingPolicy policy.minimal-archive-size = 30000000000 policy.maximal-archive-size = 150000000000 policy.grouping-keys = Space#DataSetType, Experiment#Sample:merge</pre>
--------------------------	---

In this example the candidates are unarchived data sets which have been tag by the user `admin-user` with the tag `archive`. The policy tries to find a group of data set with total size between 30 Gb and 150 Gb. It first looks for groups where all data sets are of the same type and from the same space. If no group is found it tries to find groups where all data sets are from the same experiment and sample (data set with no samples are assigned to `no_sample`). If no matching groups are found and at least two groups are below the minimum the policy tries to merge groups to a bigger group until the bigger group match the size condition. If no group can be found an email will be sent describing in detail the several steps of finding a matching group.