Features of the SAWSConnector version 1.13.0

Project: SAWSConnector

Document type: Feature showcase

Version: 1.13.0

Date: 16.07.2021

Author: The SAWS team

saws
software
engineering

Hauptstraße 29
83533 Edling
fon: 08071/510312
fax: 08071/510311
mail: info@saws.de
internet: www.saws.de

Edling, 2021.

Hauptstraße 29
83533 Edling
Tel: 08071/510312
Fax: 08071/510311
Internet: www.saws.de
eMail: info@saws.de

SAWS GmbH & Co. KG
Sitz der Gesellschaft: Edling
Amtsgericht Traunstein, HRA 8211
UST-ID: DE230638335

Persönlich haftende Gesellschafterin:
SAWS Verwaltungsgesellschaft
Sitz der Gesellschaft: Edling
Amtsgericht Traunstein, HRB 15276
Geschäftsführer: Hagen Schneider

- Table of contents -

1. General release information

The Version 1.13.0 of the SAWSConnector family has been released on July 16th, 2021!

All features mentioned in this document can be traced using their ticket number from our ticket system "Redmine" (https://redmine.saws.de).

SVN links and the list of features included in this version can be found on our Wiki pages:

https://sawsconnector.saws.de/index.php/release-notes/version-1-13-x/

If you need access to our SVN or Redmine, please contact support@saws.de!

2. Release Spotlight Video

Click here to see the release spotlight!

For every one of our half-annually releases, we are creating showcase and tutorial videos. For this release, we created a short release spotlight video, in which one of our developers is presenting a slice of all the new features in a compact manner.

The video is hosted by Paul, one member of our development team, who speaks about comments and suggestions about these features, so we recommend watching it with sound on.

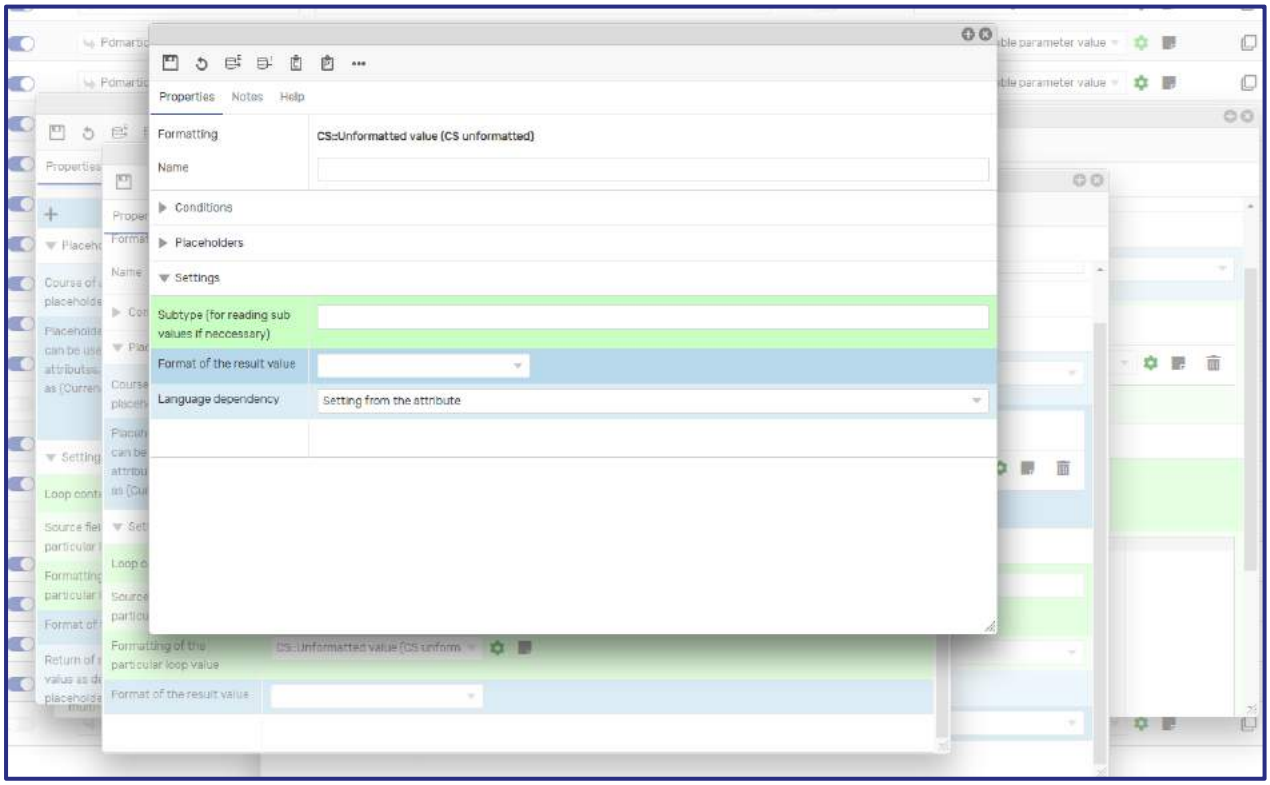
3. Major features

3.1. Datamap features

The configuration of the value transformation was greatly enhanced in terms of its possibilities but also the accessibility and user experience.

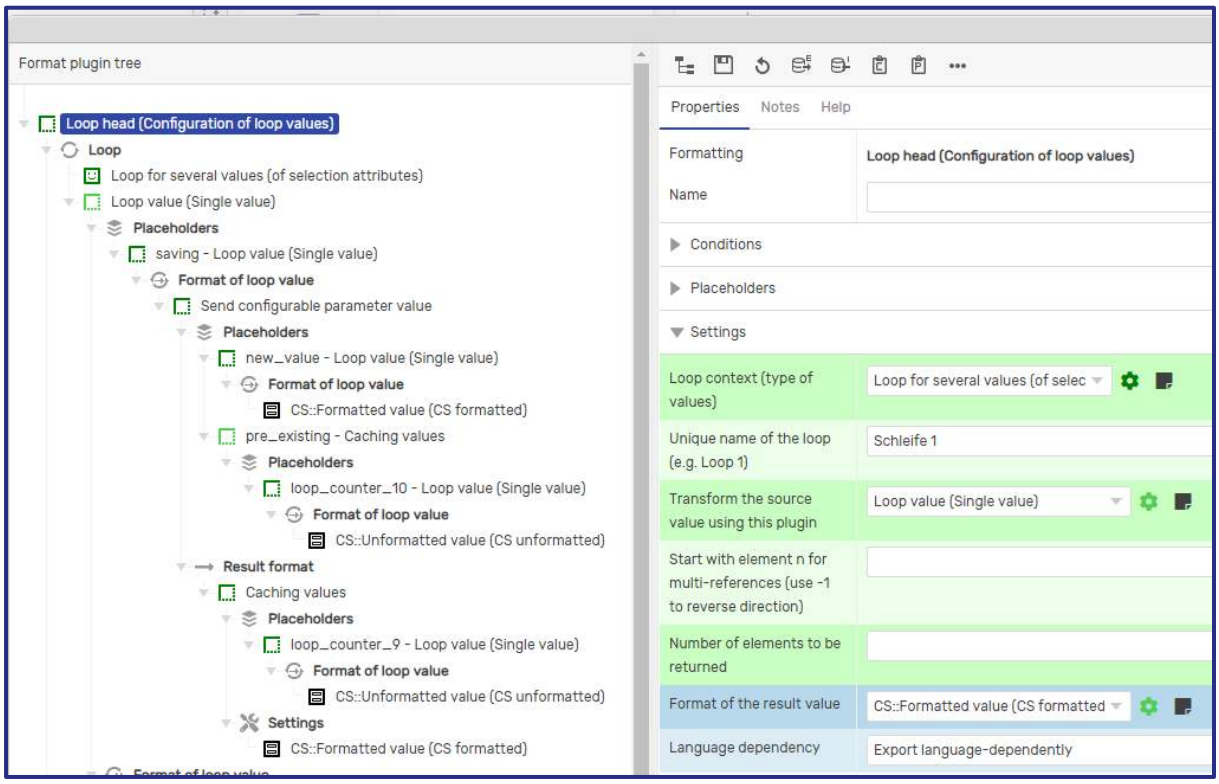
3.1.1. The format plugin tree

We noticed that format configurations, especially those with a lot of placeholders and conditions for alternative formats, could have up to 50 sub-configurations, leaving you with a cascade of windows and no oversight.



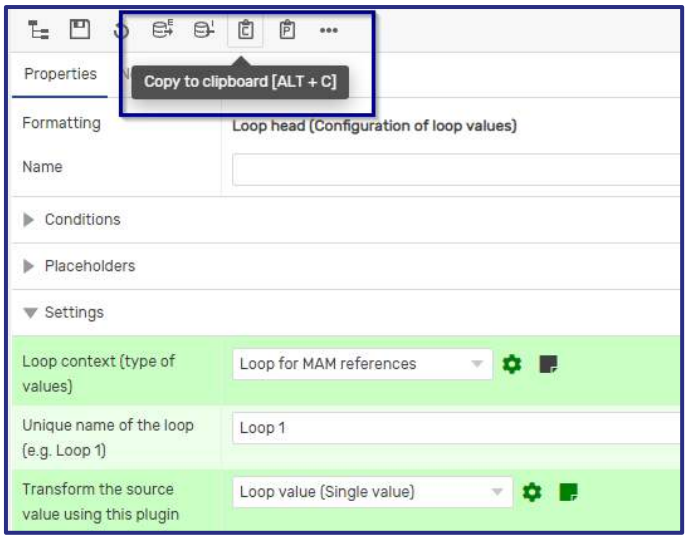
This is an extreme example

We addressed this issue by introducing the format plugin tree, allowing you to navigate these sub-configurations with ease. Any configuration can now be navigated with only one open window.



3.1.2. Copy&Paste for format settings

Format settings can now be copied and pasted analogue to *Ctrl + C* and *Ctrl + V*. Of course, we implemented the shortcut like all other Contentserv shortcuts with *Alt*, so it is *Alt + C* and *Alt + V*. You may also click the buttons as shown here.

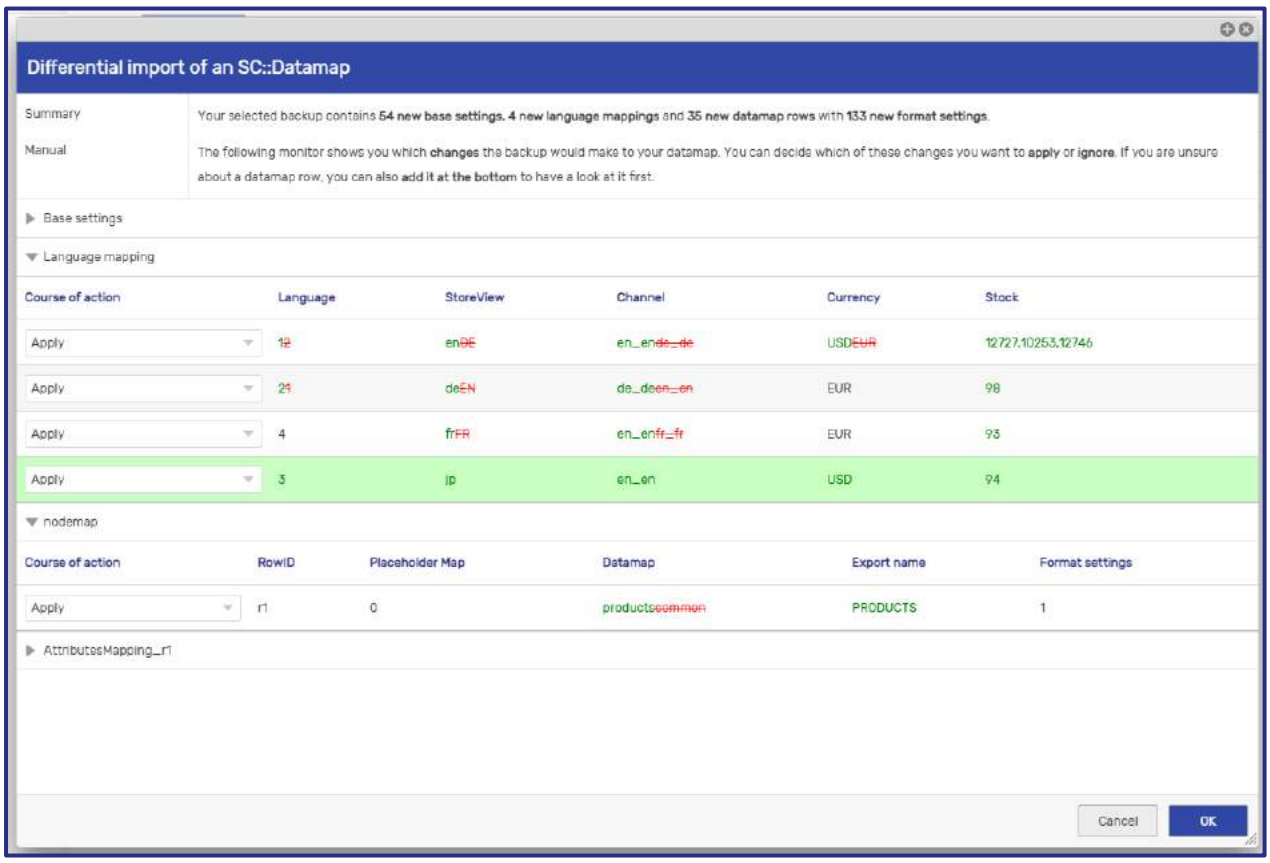


Note: If you want to import the copied format setting, you need to make sure to have the same format plugin selected. For example: A copy of a *Format strings* configuration can only be imported in another *Format strings* configuration. Aside from that you can import it in any layer and without having to save the configuration first.

3.1.3. Importing datamap backups differentially

The backup process in the connector can be a bit of a blackbox. A datamap is imported, the new settings are there but which settings are exactly new? We addressed this issue by introducing the differential datamap import.

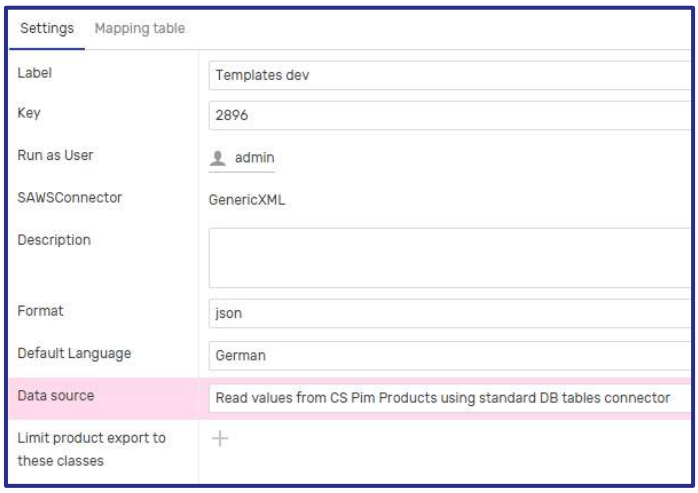
When importing a datamap configuration into your datamap, you can now choose between the direct overwrite (known behaviour) and the differential import. The differential import will open another dialog, showing you the incoming changes that will be performed to your current datamap – a *diff*, so to speak.



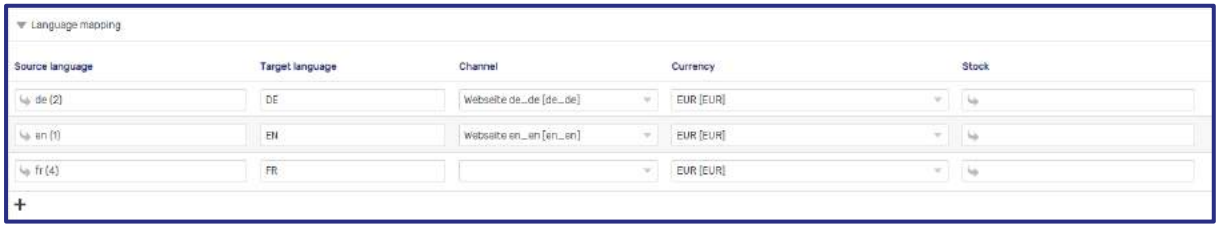
You can then decide which of these changes you want to apply or ignore. If you are unsure about a datamap row, you can also add it at the bottom to have a look at it first.

The settings are divided into up to four parts – the **Base settings**, **Language mapping**, **Nodemaps** and **Mappings** itself.

The base settings are straight forward. These are the settings at start of your datamap.



The language mapping is the collection of languages in which your values will be transformed. You find it at the bottom of the settings.



The nodemaps are containers for a datamap mapping. Adding a nodemap is often required to build a new submapping. They are most commonly known in the GenericXML connector but also appear in various other occasions.

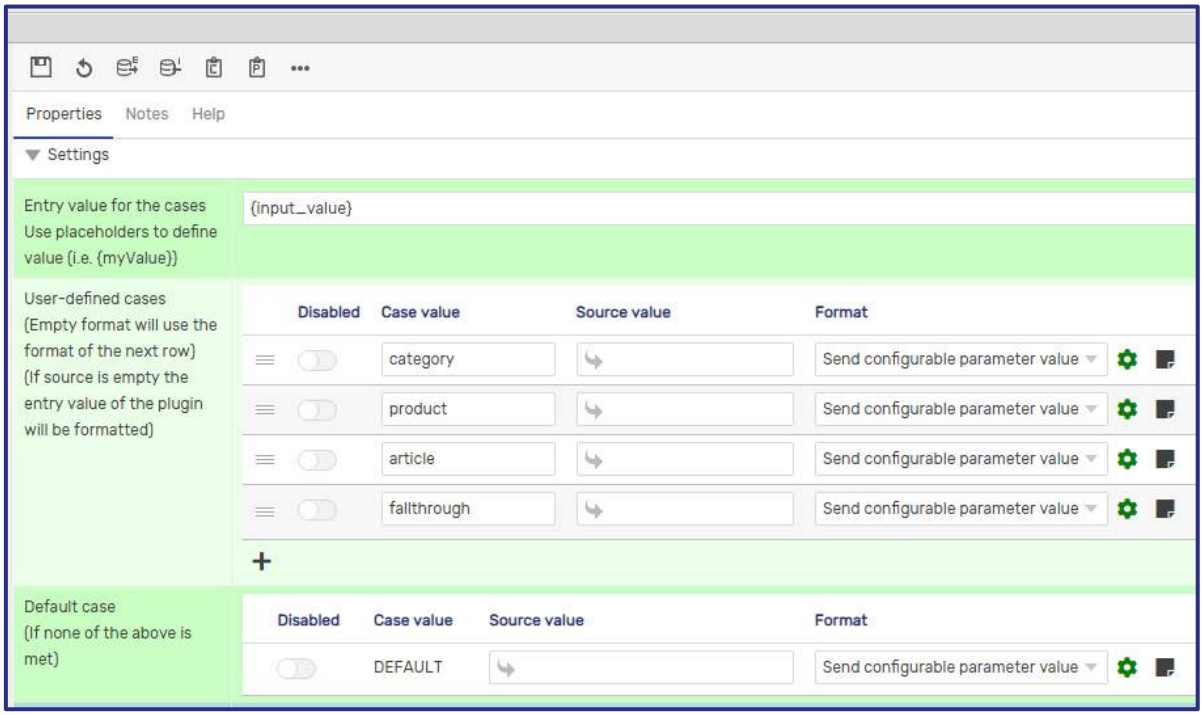
The mappings themselves are the core part of the export. They are the classic Attribute-Target-Format-Combo that we know and love.

We plan to improve the accessibility of this feature in the future with many descriptions and convenience implementations, so stay tuned!

3.1.4. Conditional format plugin (switch-case)

We introduced conditions and loops, so it was only a matter of time until we implemented a switch-case. This new format plugin allows you to combine multiple conditions in one configuration.

At the top, a placeholder is entered that shall be checked for its value. If you want to use the incoming value of the previous format, use *Source* in the placeholder mapping.



If that entered value equals then for example “category”, then the transformation of the category case in the case mapping below is executed.

If the format plugin of a case is left empty, the format of the next case will be executed. On top of that, a default case can be given that will be executed if none of the above apply.

We **highly** recommend you to read the help page of this plugin because an even more detailed example is explained there.

In a nutshell this plugin represents an SAWSConnector configuration with infinite conditions that have “Use alternative format” selected. It therefore saves a lot of time for having many different cases.

```
switch (expression) {
  case value1:
    //Statements executed when the
    //result of expression matches value1
    [break;]
  case value2:
    //Statements executed when the
    //result of expression matches value2
    [break;]
  ...
  case valueN:
    //Statements executed when the
    //result of expression matches valueN
    [break;]
  [default:
    //Statements executed when none of
    //the values match the value of the expression
    [break;]]
}
```

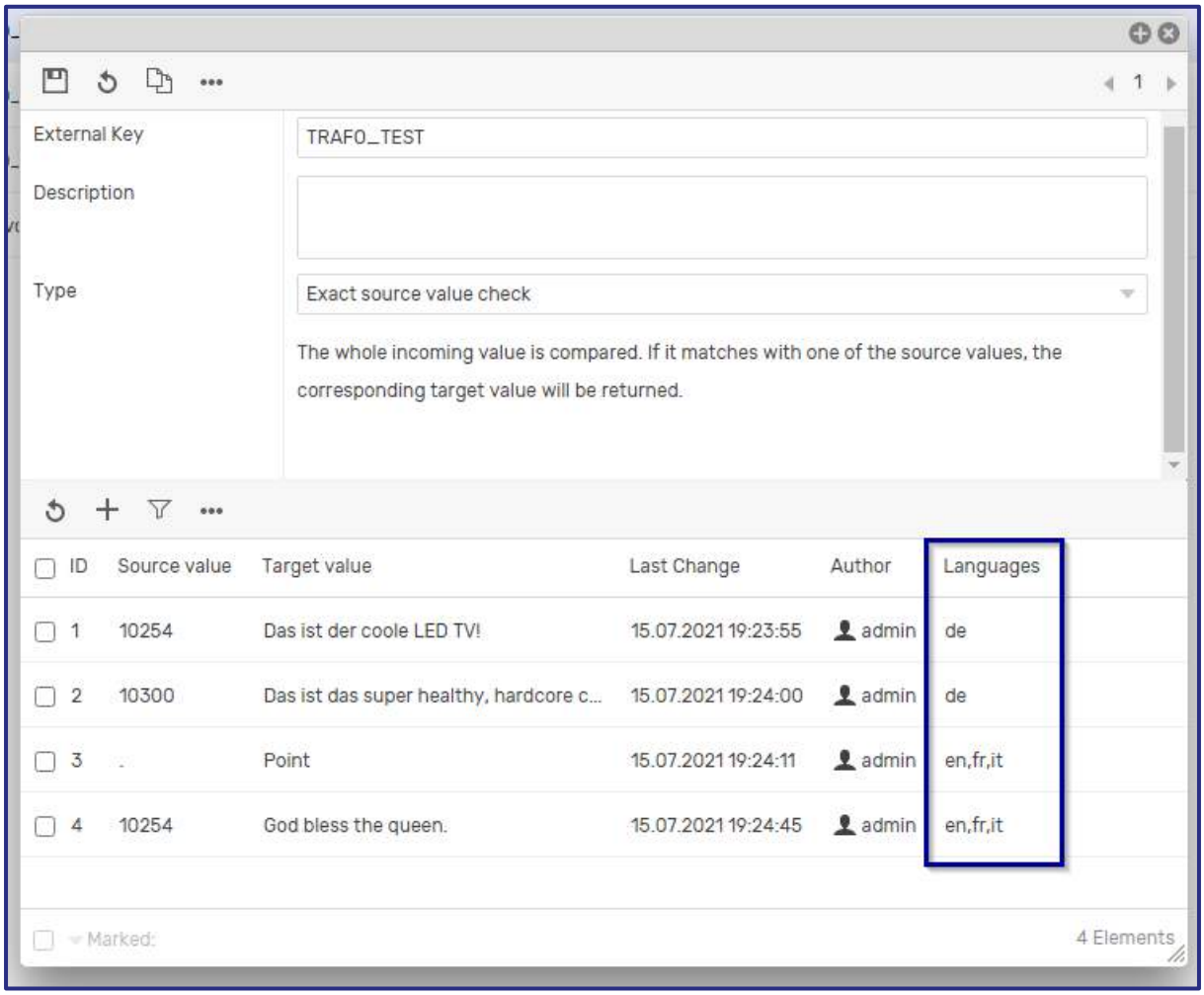
Programmers will instantly recognize this

3.1.5. Search bar for datamap columns

It is now possible to search the columns of a datamap. This is very useful for navigating a huge mapping or looking up all usages of a certain attribute.

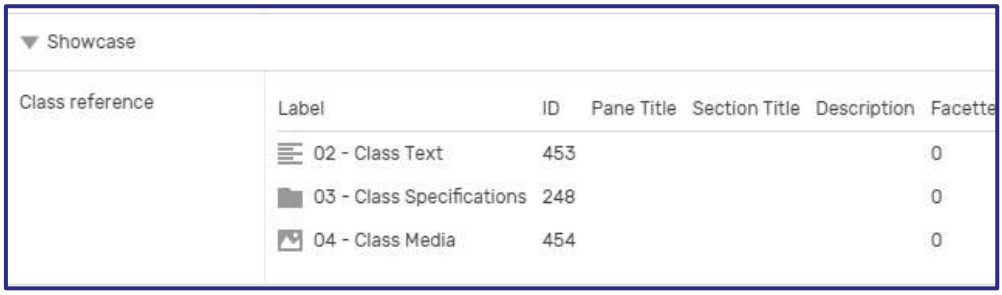
3.1.6. Language-dependent transformation lists

It is now possible to limit transformation of an SC::Transformation list to languages. It is even a multi-select which means that one transformation can apply to for more than one language. Of course, leaving the languages empty will apply the filter for all languages.



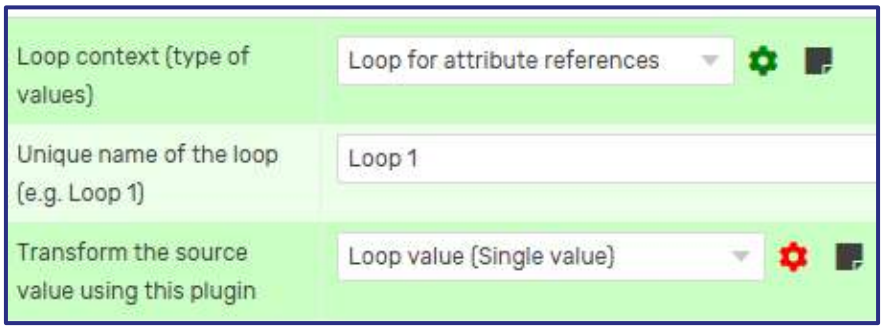
3.1.7. Export of attributes of class references in the class context

It is now possible to export attributes of class references instead of just the classes itself.

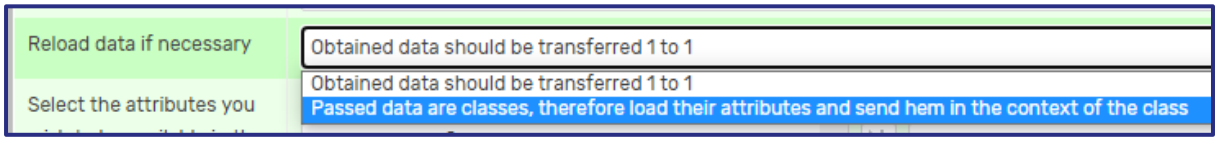


The functionality was added in the attribute reference loop plugin, so we need to configure a loop context first.

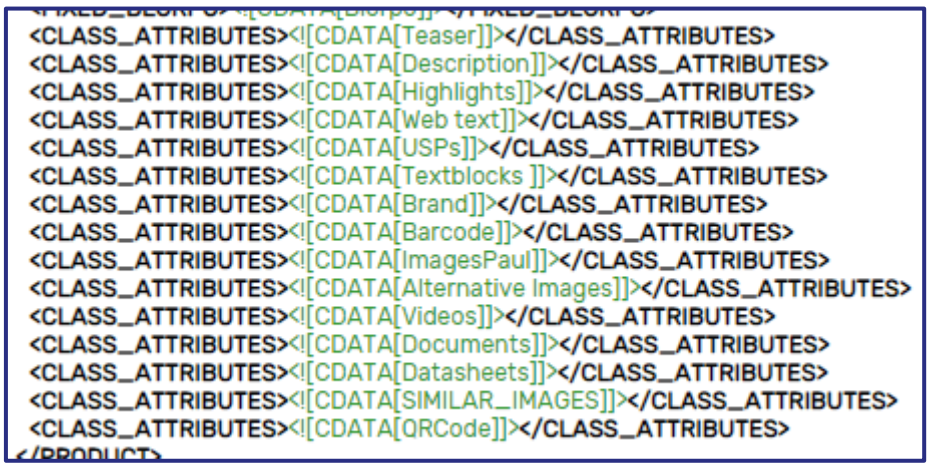
Then select *Loop for attribute references*.



Select the new option to load the attributes in the class context. If the attribute differs in the class context, this will be regarded by the connector.



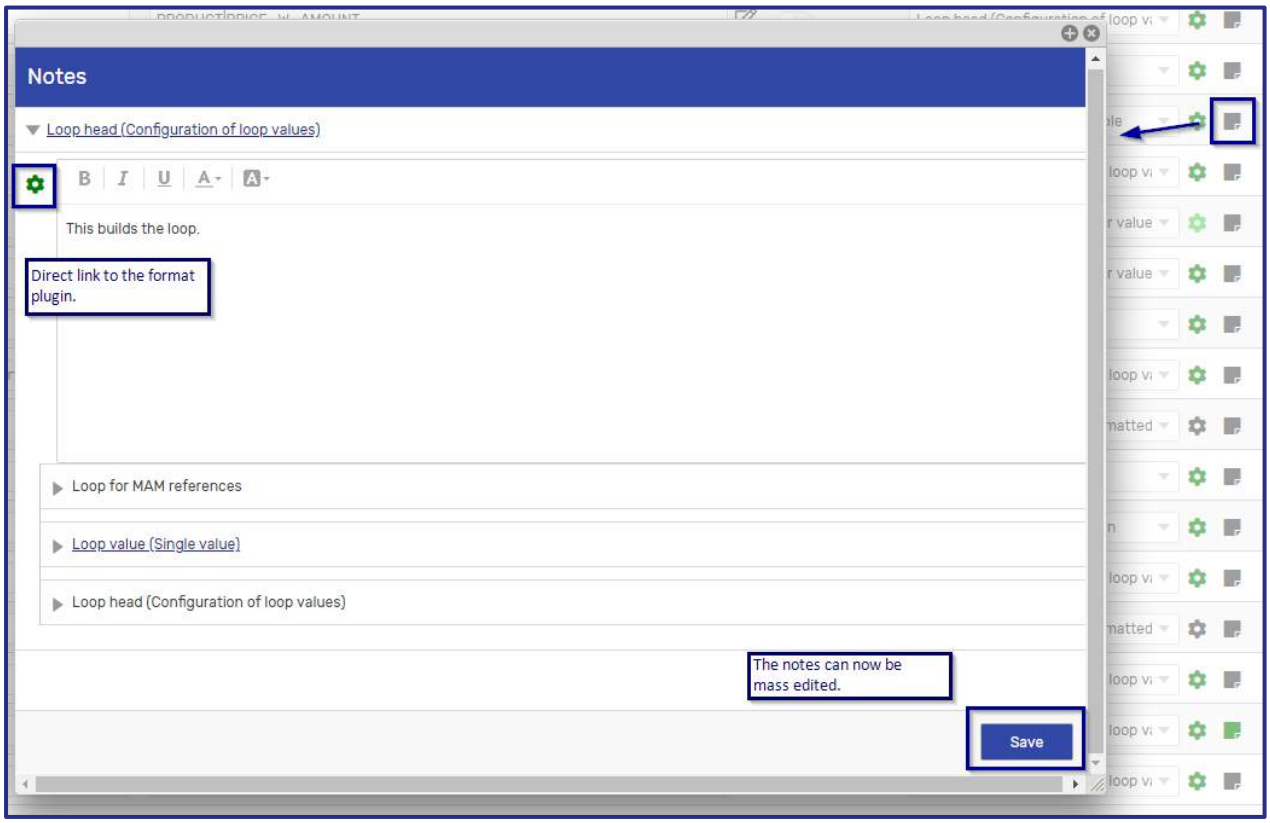
Then you can export the attribute properties in any way, shape or form.



3.1.8. Mass edit for format plugin notes

The plugin note overview was improved with two new features. It is now possible to open up the format setting directly from the note and the notes can now be mass edited.

The notes are intended to be a documentation that can be left for your colleagues or your future self. Complex configurations can oven be described in a single sentence, so this can be very useful when returning to an old configuration.



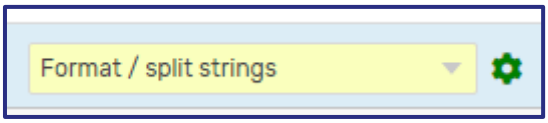
3.1.9. Combining references from multiple attributes with a new format plugin

A new format plugin was added that allows you to merge references before formatting them. This is useful if you have multiple image references that shall be exported in the same collected manner.

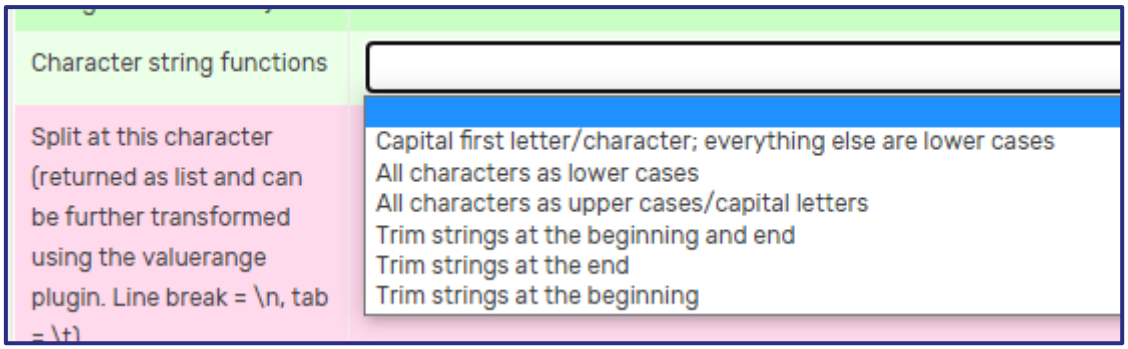
Simply select the reference attributes in the plugin. You can then pass them on to the loop head for references like you would normally do with a single reference.

3.1.10. Possibility to trim text in the string format plugin

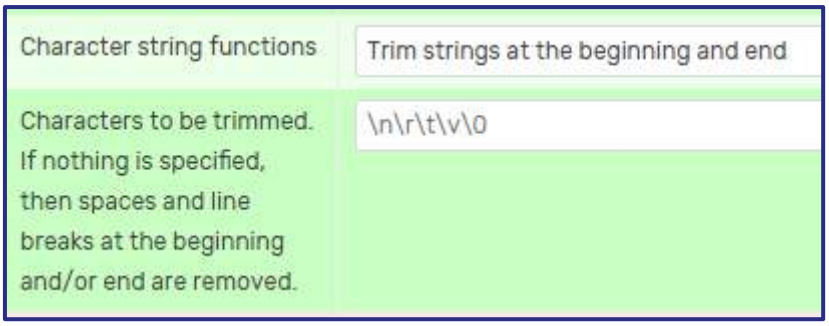
The *Format / split strings* plugin was extended by the possibility to trim strings. Trimming means cutting of certain unwanted characters at the beginning and/or end of a string.



The trim function can be selected here.



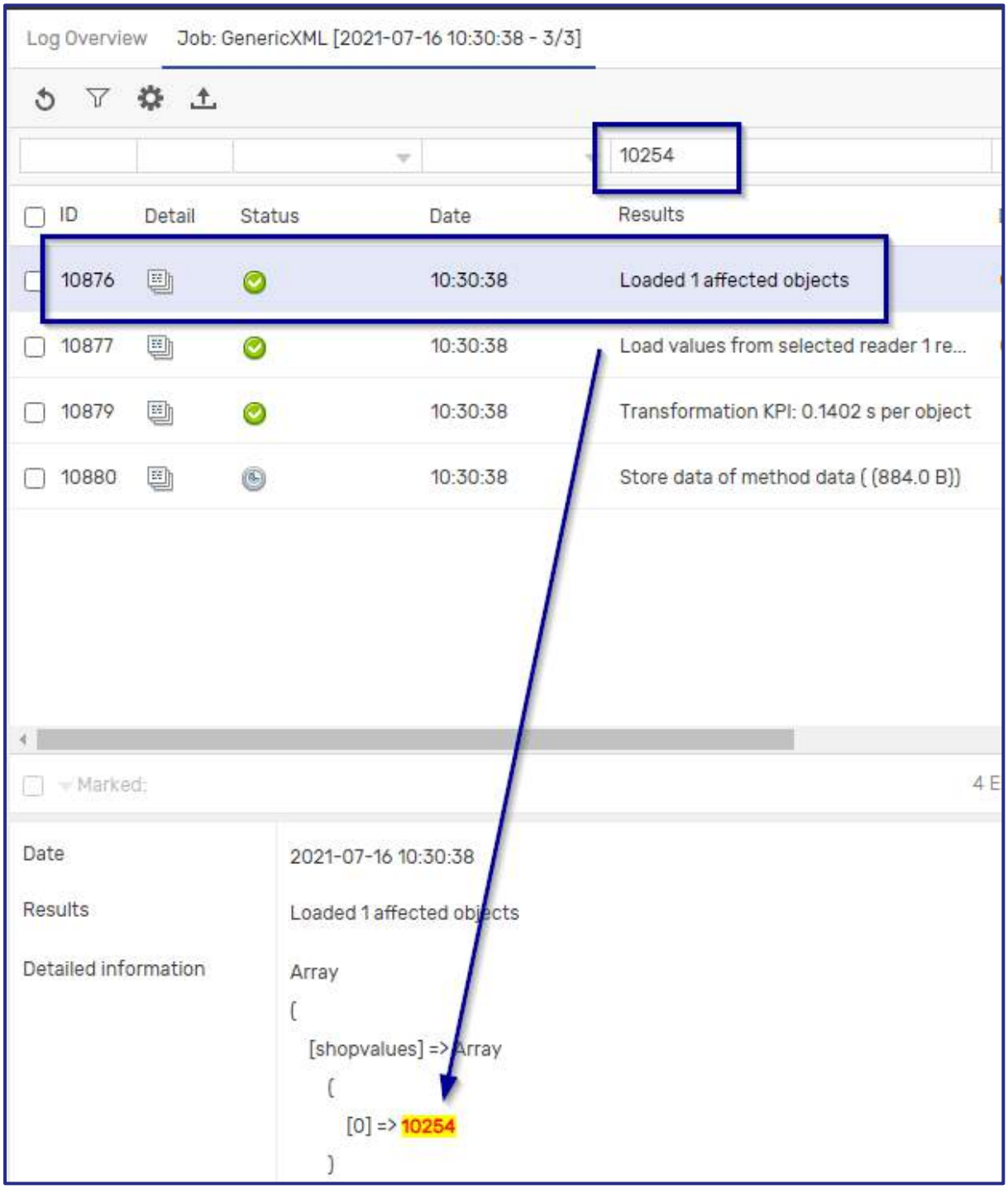
Then the characters can be provided that shall be removed. For example *\n* is the control sign for a line-break.



3.2. Export features

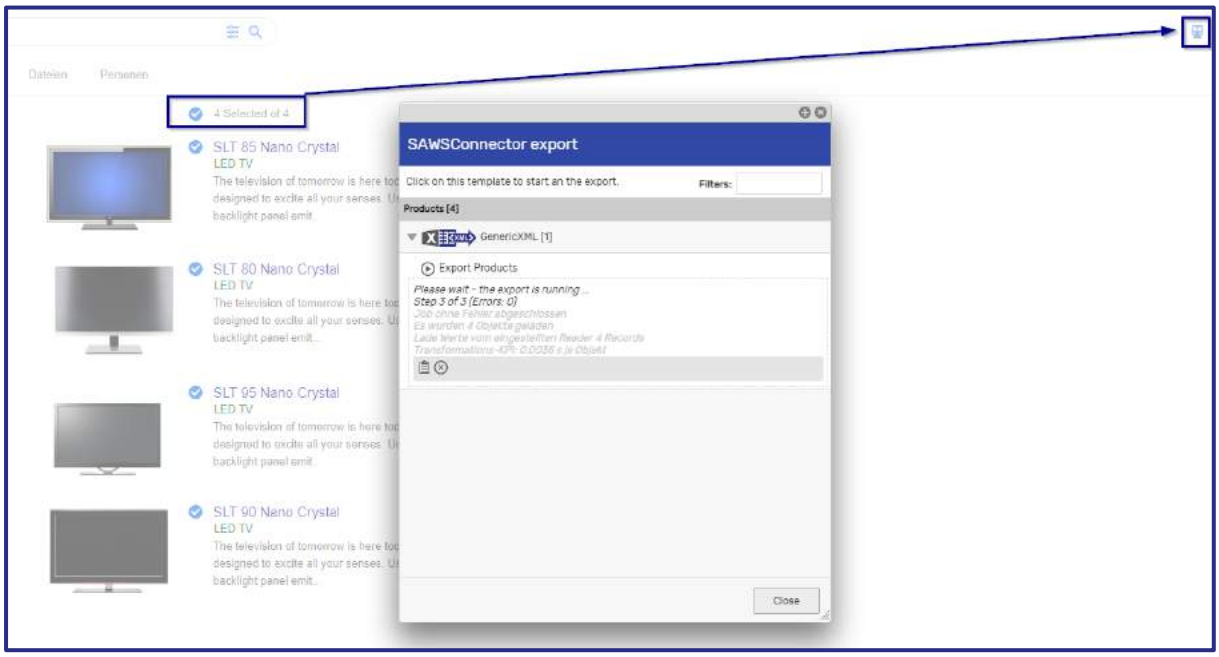
3.2.1. Full content search in the log entries

To this point the search in the log entries of connector jobs only searched in the main message of the entry. It now searches the detailed content as well and highlights the results, leaving you with far more results when searching for a specific step in the export.



3.2.2. Integration of the SAWSConnector exports into the DeepSearch

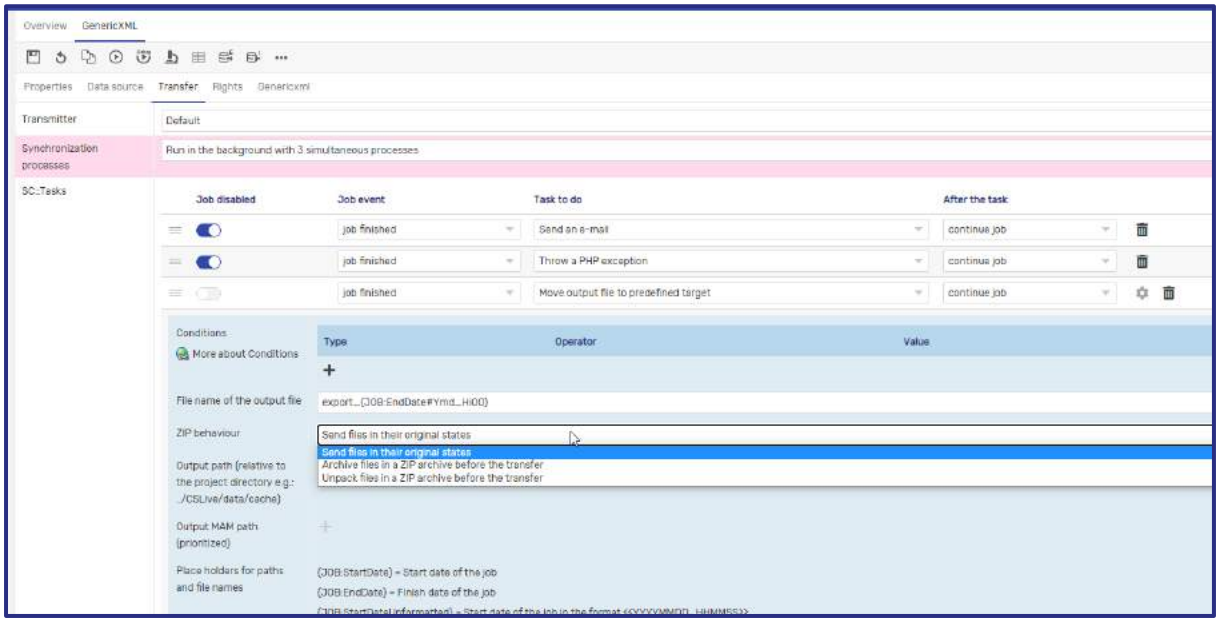
We established the compatibility of the SAWSConnector with the Contentserv DeepSearch module. You can now perform exports directly from your search results by clicking the connector train icon in the toolbar.



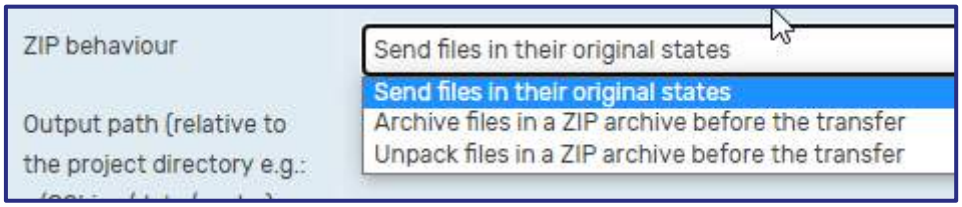
Additionally, we implemented the search for SC::Channels into the facetted search as well.

3.2.3. File export with zipping or unpacking

The SC::Task plugin for controlling the output file now also supports to zip or unzip the file.



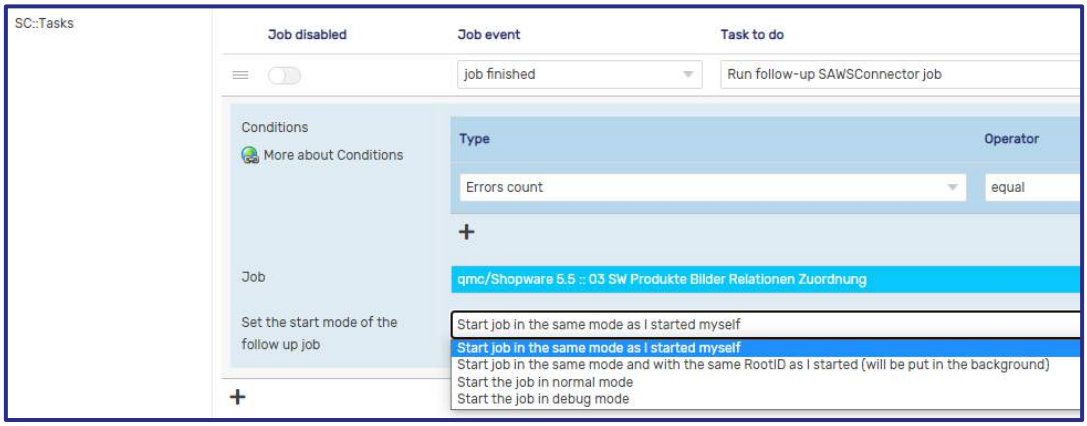
Of course, if you for example select to archive the files, they will not be archived if they already are.



3.2.4. Chaining SC::Jobs together with the same job mode

Connector jobs can be started in the normal or debug mode. The debug mode creates are more detailed logging. When chaining connector jobs together, the next job was always started in the normal mode.

You can now select in which mode the next job is started. It can even use the mode of the previous job and the same root ID.



3.2.5. Better control in which context the SC::Tasks are executed

With the conditions in an SC::Task plugin setting you can decide in which context it shall be executed. This is best explained in examples.

If for example an FTP upload shall only be exported if the user starts a job in the right-click context of the PIM product tree, this can now be configured.

The values for these new conditions are still a bit technical.

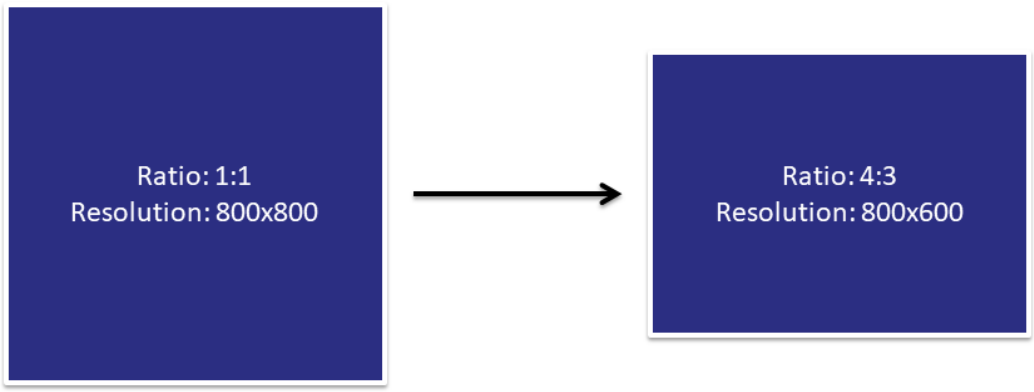
The option *Job start mode* describes in which mode the job has to be started, so the task plugin can be executed. The normal mode equals 0, the debug mode equals 1 and the development mode equals 2. This means you can control that for example plugins are not executed in development job mode.

The option *Job context* describes whether the job was started by clicking the play button (equals 1) or via the right-click context (equals -1).

3.3. PIM maintenance features

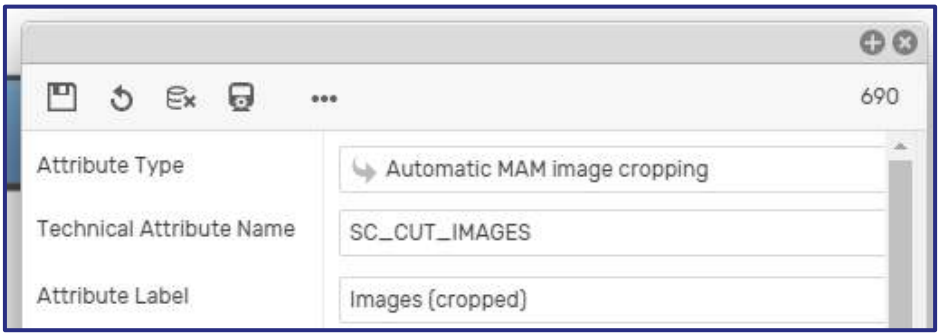
3.3.1. CSType for cropping images via an SC::Preset

It is now possible to crop images and adjust their ratio using a newly implemented MAM preset and CSType. Let's say on your website all images were previously a square and they shall be changed to the 4:3 ratio.

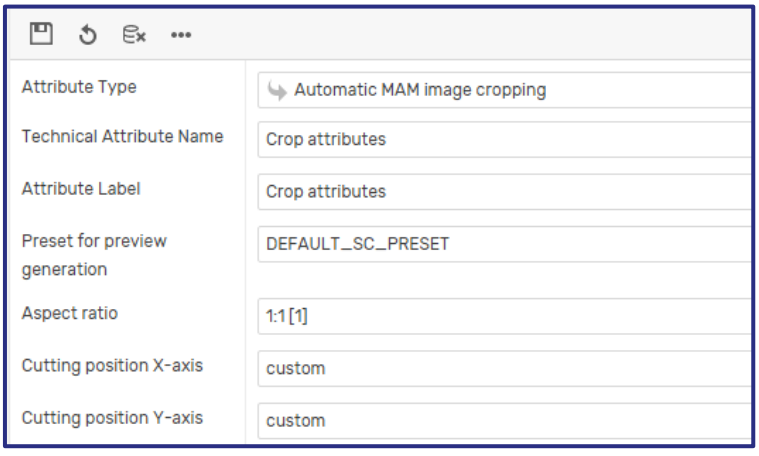


Normally you would have to maintain a whole new derivate of images, all changed to the new format. We wanted to automate this process and spare you the work of having to crop all 20.000 images or more by hand.

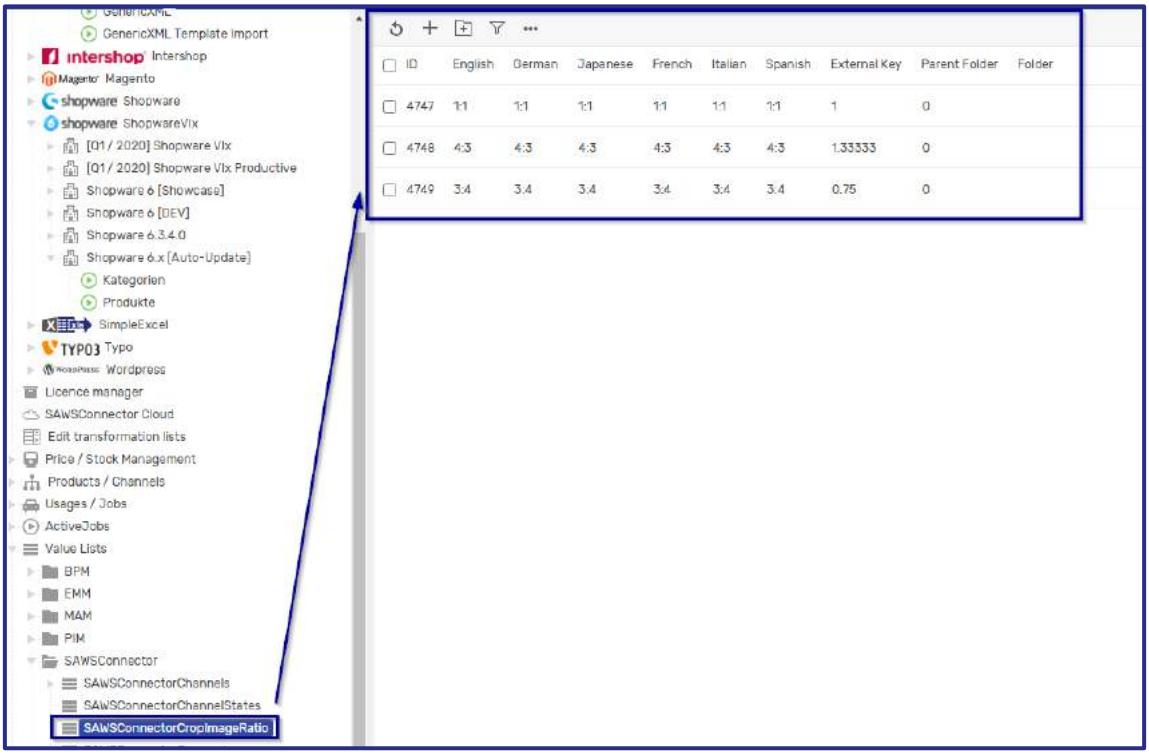
First you need to create a MAM attribute with our new CSType.



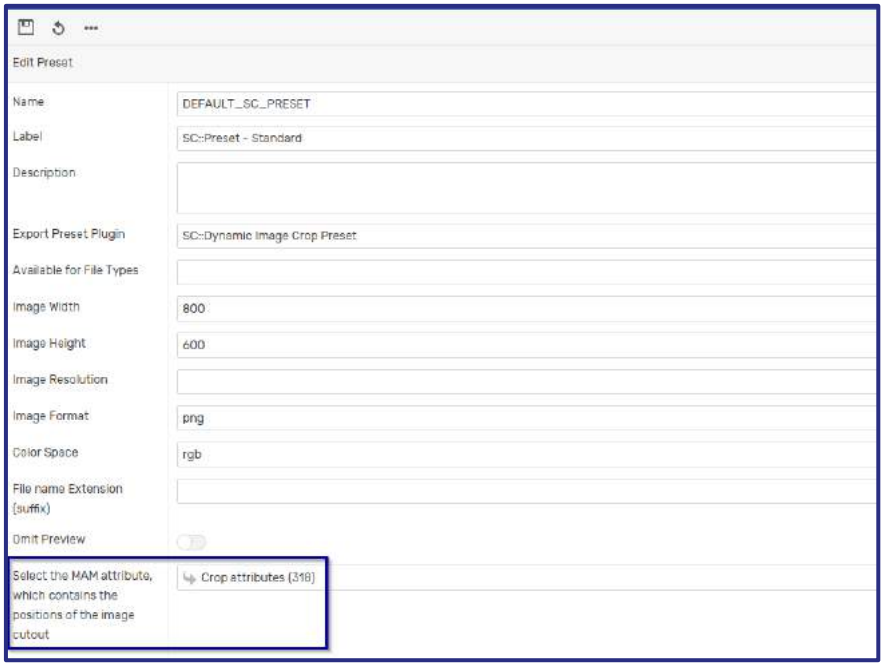
Select the requested aspect ratio and configure the X- and Y-axis for the cropping. You can also set that to *custom*, so that you can then crop it yourself. Of course, you can use inheritance.



The possible aspect ratios are maintained in the CSValueRange called *SAWSConnector-CropImageRatio*. You can add more here and if the External Key is set correctly, the connector should be able to dynamically interpret it in the image calculation.

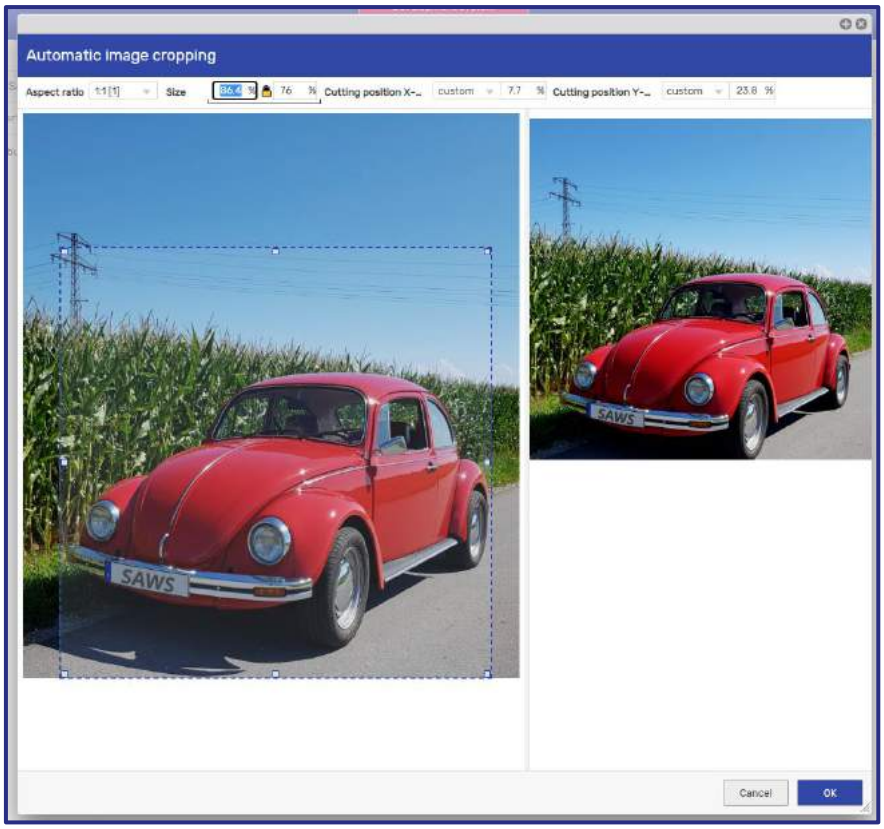


Next up, add the new preset and select the newly created MAM attribute in it.

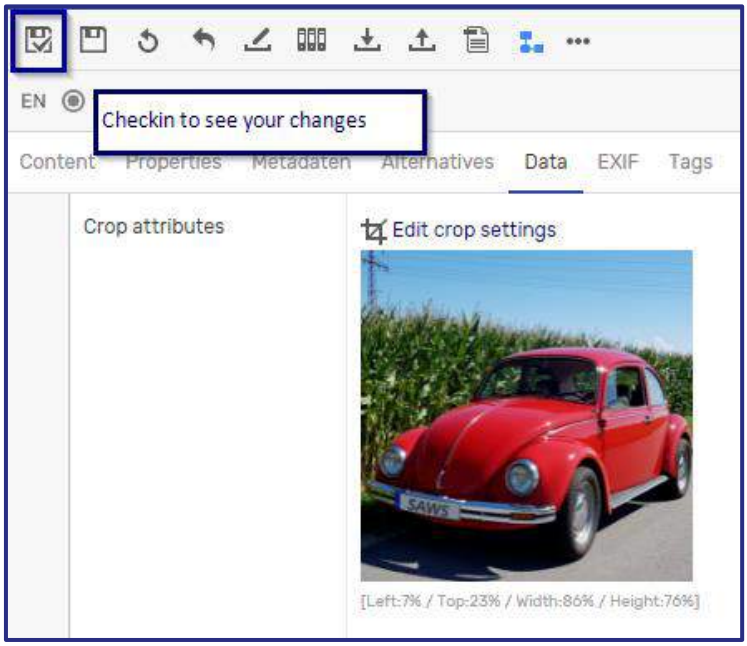


Add the attribute to the right class in your MAM as you would add any other attribute and you will see a preview for the crop settings.

Click on *Edit crop settings* and a new dialog will pop up, allowing you to fine-tune the crop position.



Then checkin or save the product to see your changes in the preview. If an image is now exported via the new preset for example in a connector export, it will read your crop settings from the image itself and then export it in this manner.

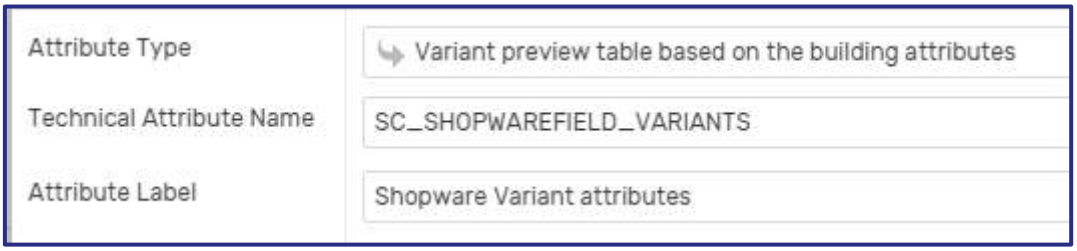


We plan to implement more improvements to this feature, including dynamic updates and previews in the coming months, so stay tuned!

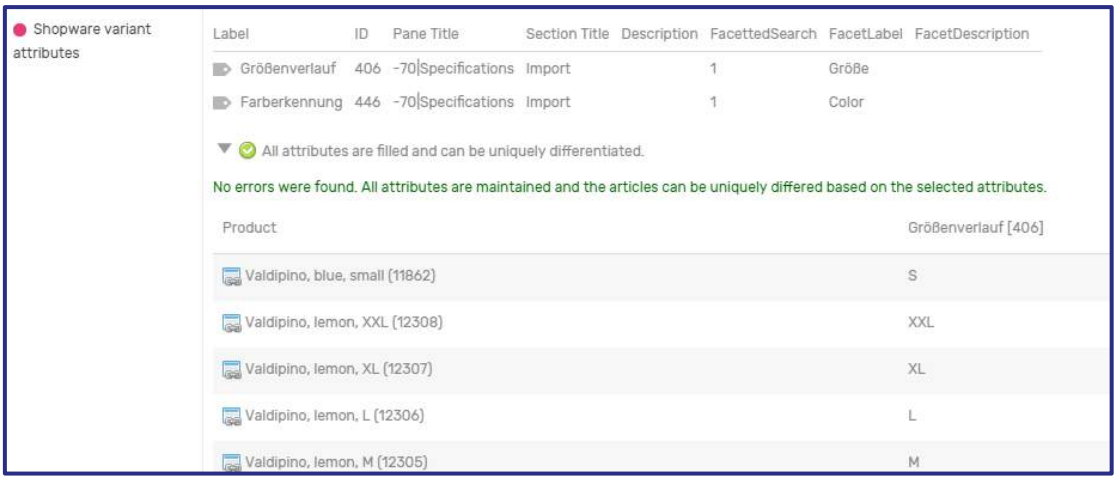
3.3.2. CSType for a variant attribute selection with preview

Nearly all target systems support a kind of variant structure for example for size variants. So variants are regularly built in the PIM tree with an attribute reference often called “Variant building attributes”. These are maintained in an attribute reference.

We implemented a new type of attribute reference that not only lets you pick and export the attributes as before but also shows if the variants, built with these attributes, are all maintained correctly. You can easily switch your selected attribute reference to the variant preview table without changing any values or destroying any previously existing maintenances.



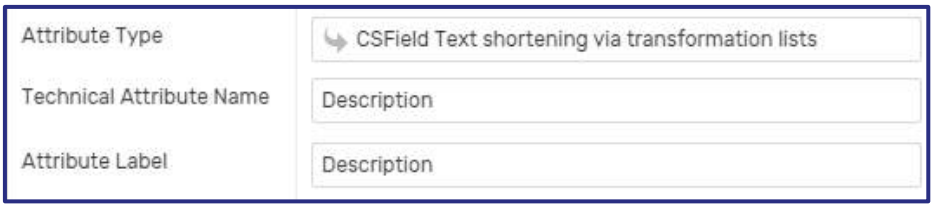
Once the plugin is maintained in your product, it checks all variants if they are correctly built with the given variant building attributes. It will then display a message about the result and print a list of all built variants with the according attributes.



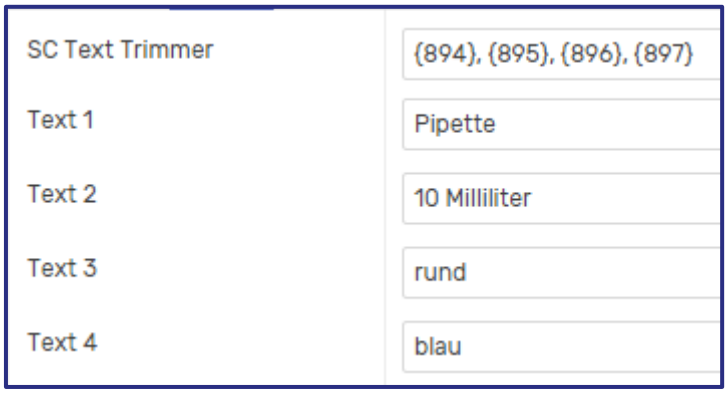
In case not all values are maintained correctly, meaning that they cannot be uniquely differed, the feature marks the missing values red.

3.3.3. CSType for text editing via SC::Transformation lists

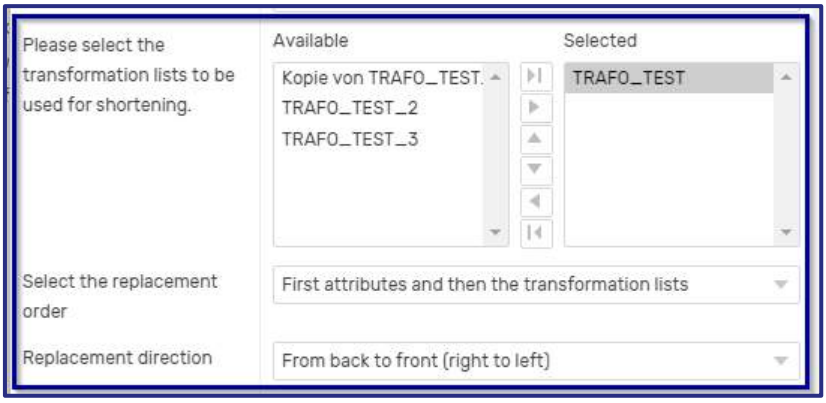
We introduced a new CSType allowing you to build dynamic text out of attributes while simultaneously using SC::Transformation lists to format the text to fit a certain length.



In this example we want to build and shorten the text 1-4 to a short description, so we use the known placeholders with *{ID of the attribute}*.



Then select the transformation lists with which you want to trim the text.



Provide the targeted text length.

Then checking the product in builds the requested text in the right length using the given transformation lists.

4. List of all features

#	Projekt	Release Notes
3666	SAWSConnector - Allgemein	Implemented loading notifications for nearly all connector windows
4541	SAWSConnector - Allgemein	Implemented a better oversight for bigger exports when using the development editor
4417	SAWSConnector - Allgemein	Implemented the support of the new Contentserv object types and templates
4391	SAWSConnector - Allgemein	Implemented a new CSType allowing you to shorten and replace values via SC::Transformation lists
4347	SAWSConnector - Allgemein	Implemented the possibility to copy & paste format settings with Alt+C and Alt+V
4065	SAWSConnector - Allgemein	Implemented a field plugin tree, allowing you to navigate complex field plugin trees with ease
4356	SAWSConnector - Allgemein	Added configuration state color to the field plugin tree icons
4490	SAWSConnector - Allgemein	Implemented the possibility to search and filter in columns of the datamap
4488	SAWSConnector - Allgemein	Implemented a new column for the expected delivery time to the SC::Stock subamounts
4037	SAWSConnector - Allgemein	Implemented the possibility to trim texts with the string format plugin
4472	SAWSConnector - Allgemein	Added the possibility to filter log entries by their content
3995	SAWSConnector - Allgemein	Added multiple configurations to the "Export multiple attributes" field plugin that enables the user to increase its performance by fine-tuning the exported values
4102	SAWSConnector - Allgemein	Added new conditions to the SC::Task plugins allowing you determine in which start contexts of a job the task plugin should be executed
4103	SAWSConnector - Allgemein	Added the possibility to archive or unpack ZIP files when using the Move-File-SC::Task plugin
4038	SAWSConnector - Allgemein	Added the possibility to determine the amount of decimal places (e.g. 0.1 with 4 places => 0.1000)
3994	SAWSConnector - Allgemein	Changed the field notes overview dialog, so that the notes can be edited directly in the overview
3784	SAWSConnector - Allgemein	Changed the design of the notes to allow the user to open a field setting directly from the notes
3639	SAWSConnector - Allgemein	Added manuals to the edit dialog of the transformation lists
4361	SAWSConnector - Allgemein	Implemented many visual and quality of life improvements for the field plugin tree
4144	SAWSConnector - Allgemein	Added a description field to all datamaps
4131	SAWSConnector - Allgemein	Added a new field plugin allowing to combine references of multiple attributes into one flat data structure
4052	SAWSConnector - Allgemein	Added a new CSType allowing you to see built variant by the variant attributes (e.g. size) and whether they are maintained correctly

3841	SAWSConnector - Allgemein	Adjusted log overview to show the name of the executed job as well
3673	SAWSConnector - Allgemein	Implemented search for SC::Channels in the DeepSearch and enabled exports directly from the search results
4321	SAWSConnector - Allgemein	Implemented different logos for transmitters
4307	SAWSConnector - Allgemein	Implemented a new switch case field plugin allowing you to execute different transformations for different configurable cases
3824	SAWSConnector - Allgemein	Added a possibility to decide in which mode a following job shall be started when using the Job Starter SC::Task plugin
4425	SAWSConnector - Allgemein	Implemented a new virtual export field "HasLanguageDependentValues" which sends a flag that is more precise than just "IsLanguageDependent"
4390	SAWSConnector - Allgemein	Implemented the possibility to use transformation lists language-dependently
4118	SAWSConnector - Allgemein	Added the possibility to export attributes of class references via the according loop field plugin
4187	SAWSConnector - Allgemein	Implemented internal improvements for all connector mapping tables
3515	SAWSConnector - Allgemein	Greatly improved the debugging API of the SAWSConnector and adjusted the exports to print data in a slightly improved manner
4376	SAWSConnector - Allgemein	Added a new functionality that allows you to import datamaps differentially, selecting which datamap rows shall be imported and which not
3959	SAWSConnector - GenericJSON	Implemented the possibility to group all values of each language in a record each
4117	SAWSConnector - GenericJSON	Backward incompatible change: Removed forced language dependency from the target column "name"
4172	SAWSConnector - GenericJSON	Implemented a transmitter for Apache Kafka
4099	SAWSConnector - Magento	Implemented the possibility to remove variant options if they would only have one option
4276	SAWSConnector - Magento	Implemented the possibility to map categories only to variants or only to the variant head
4365	SAWSConnector - Magento	Implemented support of the format plugin tree for the extended attribute configurations
4379	SAWSConnector - Magento	Implemented the possibility to disable the meta description for categories
2765	SAWSConnector - Oxid	Implemented compatibility of the OXID module with non-standard language dependent fields
4526	SAWSConnector - SimpleExcel	Implemented new internal systems for the advanced support of language dependent fields
4175	SAWSConnector - SimpleExcel	Implemented a new format plugin allowing you to export arrays with key-value pairs directly as Excel columns
4332	SAWSConnector - SimpleExcel	Improved the design and presentation of the cell format options in the field plugin tree
3537	SAWSConnector - SimpleExcel	Implemented the possibility to limit tabs to certain languages
3909	SAWSConnector - SimpleExcel	Implemented the possibility to set the Excel cell format for the header row as well