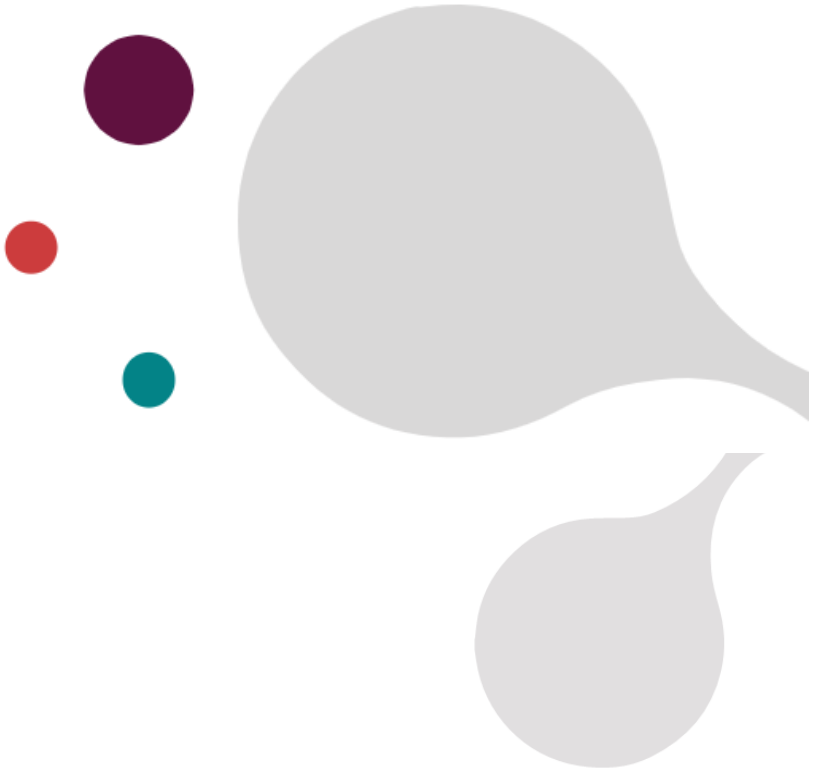


axiell

Axiell Collections
model applications 5.0
release notes





Axiell ALM Netherlands BV

Copyright © 2020 Axiell ALM Netherlands BV® All rights reserved.
Adlib® is a product of Axiell ALM Netherlands BV®

The information in this document is subject to change without notice and should not be construed as a commitment by Axiell ALM Netherlands BV. Axiell assumes no responsibility for any errors that may appear in this document. The software described in this document is furnished under a licence and may be used or copied only in accordance with the terms of such a licence. While making every effort to ensure the accuracy of this document, products are continually being improved.

As a result of continuous improvements, later versions of the products may vary from those described here. Under no circumstances may this document be regarded as a part of any contractual obligation to supply software, or as a definitive product description.

CONTENTS

1 A brief list of all changes	1
2 Selected topics	15
2.1 GIS functionality and fields added	15
2.2 Maps records and fields	16
2.3 Linked Open Data and Persistent ID's	21
<i>New data type: URI</i>	22
<i>New PID fields in Axiell Collections</i>	23
<i>Management of PIDs via Handle/SurfsARA</i>	25
<i>Workflow</i>	26
<i>Notes</i>	27
2.4 The "group" term status (neutral value 6)	28
2.5 A location <i>Gap report</i>	30



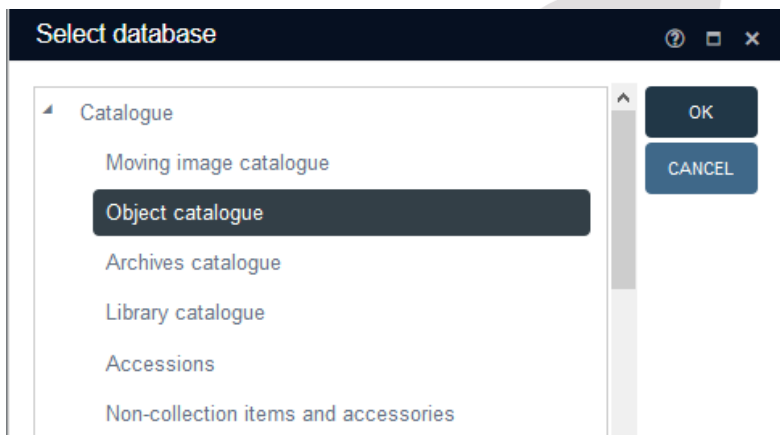
1 A brief list of all changes

Version 5.0 of the Axiell Collections model applications (previously known as the Adlib model applications) is the first to have been developed for Axiell Collections only (making the previous version, 4.5.2, the last version to be fully Axiell Collections-ready as well as completely Adlib for Windows compatible).

We recommend always using the latest Axiell Collections core software release available (at least 1.7.1), regardless of the Collections model application version you have.

Besides many small improvements, model applications 5.0 contain quite a few major changes too:

- To start with, these applications have been made [Spectrum 5.0](#) compliant. To this end, new fields, screens and some database tables and data sources have been added because of changes in the following Spectrum procedures: *Collections review*, *Use of collections*, *Insurance and indemnity*, *Collections care and conservation*, *Deaccessioning and disposal*, *Loans out*, *Location and movement control*, *Object entry*, *Object exit*, *Valuation* and *Reproduction*.
- Library records now reside in the same database table as museum objects and archival items, allowing for an encompassing **Catalogue** data source in which you can search for (and edit) all these materials simultaneously. The *Catalogue* has six sub data sources: *Moving image catalogue*, *Object catalogue*, *Archives catalogue*, *Library catalogue*, *Accessions (bulk)* and *Non-collection items and accessories*.



- A **Moving image catalogue** data source, implementing the [FIAF cataloguing rules for film archives](#), has been added to register film materials. These records will be stored in the newly combined museum/archive/library table as well. For this data source, five so-called [record types](#) have been implemented to build any allowable internal hierarchy: WORK (e.g. a story or script for example), MANIFESTATION (e.g. a movie or a theatre play), ITEM (e.g. a film reel, dvd box set), ITEM_PART (e.g. an individual dvd from a box set) and VARIANT (an expression, maybe a black and white version of the movie). So-called [relationship dependency rules](#) have been set up to define the allowable hierarchies:

WORK

- WORK
- MANIFESTATION
- - ITEM
- ITEM
- - ITEM_PART
- VARIANT
- - MANIFESTATION
- - ITEM

In effect, this model hierarchy defines which direct parent and which direct children any record of the listed type is allowed to have. So a WORK record in our example can parent other WORK records, MANIFESTATION records, ITEM records and VARIANT records, but not ITEM_PART records, while it can be a child of another WORK record (e.g. if a WORK is part of a series of WORKs). A MANIFESTATION record can only be parented by a VARIANT or WORK record and can only have ITEM children if it is parented by a WORK. And so on. Note that the model in itself only demands that *if* you create a relation between two record types, it must comply to this model: it doesn't require you to actually create relations at all.

The record type of an individual record also determines which screens and thus which fields you get to edit.

The first thing you now have to do when creating a new record is to set the record type for this record.



New Catalogue record

Select a record type

Record type

- Work
- Variant
- Manifestation
- Item
- Item part

- The **Object catalogue** distinction between internal and external objects is no longer made through datasets underneath a catalogue, but by the *Management status* drop-down list. By default, for new records it is set to internal (to represent an internal object) but you can set it to any other available status. The *semi-current record* status should be used for archival records management purposes, records which are in the archive's care but are not ready to be fully archived yet: they are "semi-current" in the sense that they might still be in use by the provider of the records. These records may also be excluded from the archive's own collection and be transferred or destroyed after a retention period.

Identification

Identification

Management st... internal

Institution name

Institution code

Administration n...

Collection

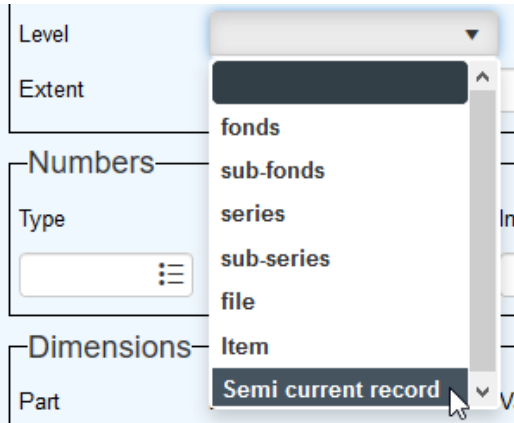
Barcode

Object number

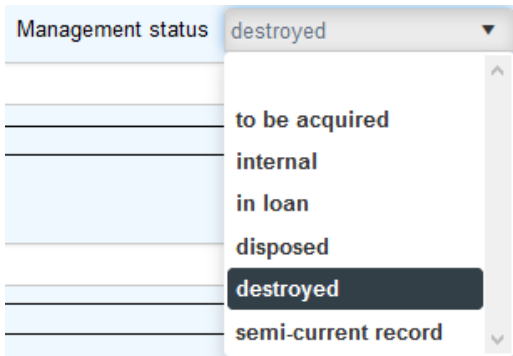
- to be acquired
- internal
- in loan
- disposed
- destroyed
- semi-current record



- In the **Archives catalogue**, the *Level* screen field now refers to the record type field, offering the following levels: *fonds*, *sub-fonds*, *series*, *sub-series*, *file*, and *item*.



Destroyed records of any level can be assigned a *destroyed* status in the new *Management status* field (next to the *Level* field) which offers six statuses in total.

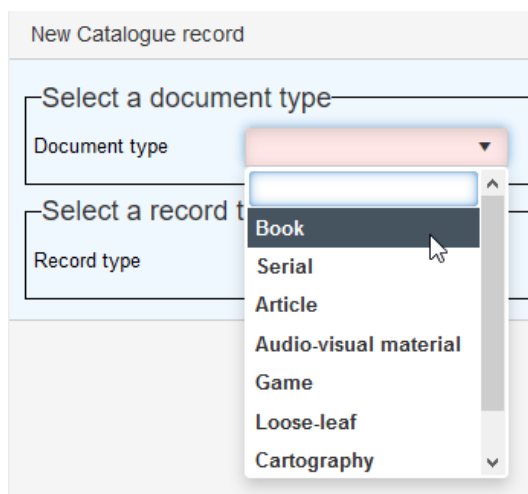


To the *Archives catalogue*, also a *Change retention reference* task was added. This task applies a selected retention schedule to all marked records at once and moves any old retention schedule reference to the retention schedule history.

- The **Library catalogue** distinction between books, audio-visual materials, games, articles, serials, loose-leaves, cartography materials and resources is no longer made through datasets underneath a catalogue but by a new, mandatory *Document type* field.

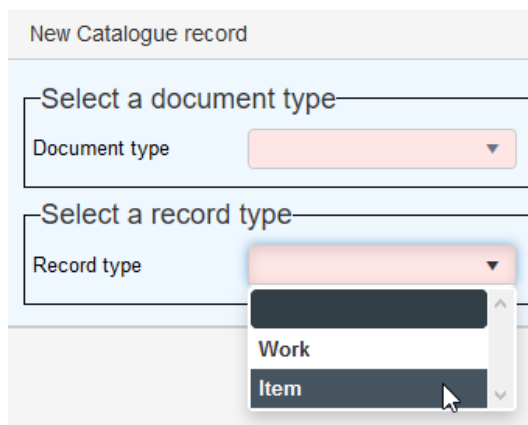


The available screens for a record will depend on the value selected in this field and on the *Record type* being a *Work* (see next paragraph about copies for more info about this new field).



The screenshot shows a form titled "New Catalogue record". It has two main sections: "Select a document type" and "Select a record type". The "Document type" dropdown menu is open, showing a list of options: Book, Serial, Article, Audio-visual material, Game, Loose-leaf, and Cartography. A mouse cursor is pointing at the "Book" option.

- **Copies** of library materials are no longer registered in their own database table, they are now registered in the combined museum/archive/library table as well, with a hierarchical relation to the associated title record. When creating a new **Library catalogue** record, you'll not only have to specify a *Document type* (as indicated above), but also a *Record type*: *Work* or *Item*, where a normal title record is considered a *Work* and a copy an *Item*.



The screenshot shows the same "New Catalogue record" form. The "Record type" dropdown menu is open, showing a list of options: Work and Item. A mouse cursor is pointing at the "Item" option.



The available screens and fields for the new record will depend on this choice. A copy record can and should be linked to an associated title via the *Catalogue number* field (tag bt) on the *Copy details* screen. The link can of course also be made from within the title record, in the *Consists of* field (tag nt) on the *Hierarchical cataloguing* screen.

- The *Archives (accessions)* data source has been renamed to **Accessions**. This data source is meant to register bulk accessions, to be processed in the **Archives catalogue** at a later time.
- A new data source is **Non-collection items and accessories**. Any object that is not suited for one of the other catalogue data sources, but still needs to be registered, can be registered in here. These items can be linked to other catalogue objects or items. Think of objects that often go together with actual catalogue objects, for display purposes for example, but are clearly not catalogue objects themselves, like a pedestal, a showcase or a dress display for a particular object.
- An **Events** data source and matching database table have been added, for the purpose of registering events related to records in the **Moving image catalogue**, like the receiving of awards, publications, decisions, censoring etc.
- In *people.inf* (the **Persons and institutions** data source), the CURATOR domain has been replaced by STAFF and the following new domains were added: ADMINISTRATION (*administration name*) (link domain for collect.inf (tag AG), AGENCY (link domain for event.inf (tag Ir), APPLICANT (link domain for event.inf (tag Ia), AWARDNOMINEE (*award nominee*) (link domain for collect.inf (tag AP) and event.inf (tag Ap), CAST (link domain for collect.inf (tag cy), CREDITS (link domain for collect.inf (tag c1), CREDITS_INST (*credits (organisations)*) (link domain for collect.inf (tag B8), EVENTAGENT (*event agent*) (link domain for event.inf (tag eo).
Relation fields were added for broader, narrower, related and pseudonym names, plus an equivalent_name.category field. A checkbox called *Create the authorized name from the name elements* has been added on the *Identification* tab, to allow you to immediately put the complete name together from the details entered in the field below it. (Previously this procedure was executed automatically, just before storage of the record.)



Identification ▲

Authorised form of name

Name

Name type

Reference number

Name note

Other forms of name

Use

Used for

Pseudonym for

Pseudonym

Equivalent

Person's name elements

Create the authorized name from the name elements

Title Gender

Surname Forename(s)

Prefix(es) to sur... Addition(s) to n...

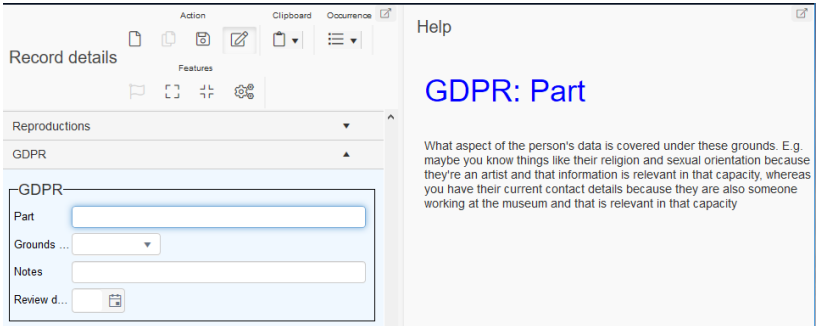
Initials

Salutation

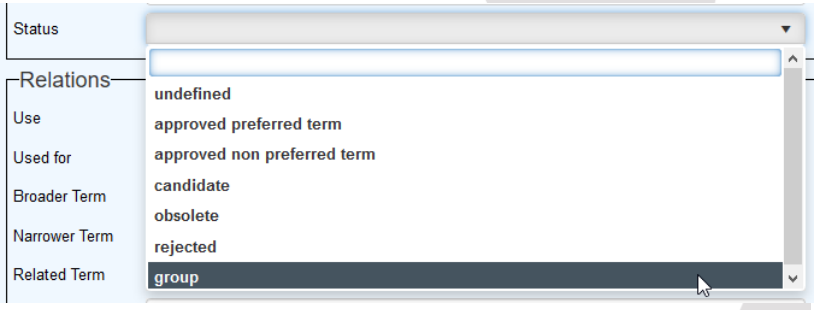
Further, the *Gender* field has been changed from an enumerative field with the limited choice between FEMALE, MALE and UNKNOWN to a linked field, allowing customers to register any value they need.

Privacy related fields have been added on the new *GDPR* screen tab.





- To the **Thesaurus** term.status field a seventh status has been added: *group* (see chapter 2.4 for more about this).



Further, a lot of new domain names have been added, namely: SOUNDTYPE, CARRIERTYPE_SPECIFIC, GENRE, CREDITTYPE, REGION, TITLETYP, BITDEPTH, COLOURSYSTEM, ASPECTRATIO, APERTURE, SOURCEDEVICE, ASSOCPI (*association (Persons & Institutions)*), CARRIERTYPE_GENERAL, CASTTYPE, STOCK, ALTERNATIVE_NUMBER_TYPE, VIDEO_CODEC, SOUNDSYSTEM, PRODUCTION_REASON, AWARDACHIEVEMENT, AWARDNAME, EVENT_SERIES, AUDIENCE_RATING, COLLECTION_STATUS, AUDIENCE_TYPE, COLOURTYPE, LOGICAL_EXTENT_UNIT, SOUNDFIXATION, FRAME_RATE, LANGUAGE_USAGE, AUDIO_CODEC, RESOLUTION, LINE_STANDARD, DESCRIPTION_TYPE, DATING_TYPE, REVIEW_METHOD, REVIEW_REASON, REVIEW_STATUS, REVIEW_TYPE, GEOCODE_TYPE, MONUMENT_TYPE, FIELDCOLLECTION_EVENT_TYPE, FIELDCOLLECTION_SITE_TYPE, FIELD_COLL_DESCR_SUBJECT (*field collection description subject*) and FIND_CATEGORY.

Added multiple domain values for event.inf (tag Et) variable link domain EVENT_TYPE_%event_type%), namely:



EVENT_TYPE_PUBLICATION (*publication event type*),
 EVENT_TYPE_PRODUCTION (*production event type*),
 EVENT_TYPE_DECISION (*decision event type*) and
 EVENT_TYPE_MANUFACTURE (*manufacture event type*).
 Further values that were added as link domains are:
 ENVIRONM_MEASURE_QUALIFIER (*environment measurement
 value qualifier*) (location.inf, tag EQ), ENVIRONM_MEASURE_TYPE
 (location.inf, tag ET), ENVIRONM_MEASURE_UNIT (*environment
 measurement value unit*) (location.inf, tag EU), GENDER (peo-
 ple.inf, tag ge), USE_RESTRICTION (collect.inf, tag UW),
 MEASURE (*measurement*) (research.inf, tag MT), MEASURE_UNIT
 (*measurement unit*) (research.inf, tag MU), REQUESTER_STATUS
 (*loan requester status*) (loans.inf, tag BB), VALUE_REASON (*valu-
 ation reason*) (collect.inf, tag tr), FILM_FORM (*form (Film)*) (col-
 lect.inf, tag OB in dataset film, record_type (Df) WORK/VARIANT),
 FILM_MANIFESTATION_TYPE (*manifestation type (Film)*) (col-
 lect.inf, tag OB in dataset film, record_type (Df) MANIFESTATION)
 and FILM_ITEM_ELEMENT_TYPE (*item element type (Film)*) (col-
 lect.inf, tag OB in dataset film, record_type (Df) ITEM/ITEM_PART.

- A **Geographical thesaurus** data source and matching database table have been added, specifically for registering geographical names, terms and map locations (see chapter 2.1 for more information about map locations). The bulk of geographical names could be so large and their typical hierarchy deserved a separate database table. So a number of place, country and geographical keyword fields in other database tables now link to the geographical thesaurus instead of to the normal thesaurus.
- In *media.inf* (the **Multimedia documentation** data source) reverse linked fields to fieldcollection.inf, thesaugeo.inf and research.inf were added on the *Linked records – other* and *Use of Collections* screens
- A **Field collection** data source and matching database table have been added, while a *Field collection* screen tab (now with a linked *Field collection reference* field), is still available for museum objects too. For individual object finds, you may still choose to use the fields on the *Field collection* screen if that is most convenient, while for field collection sites where multiple objects were found it is probably more efficient to create a *Field collection* record and link to it from all relevant object records.
 A *Field collection* record must have a *Record type*: you must choose either *field collection event* or *site*. The choice will determine which fields you get to see. There's much overlap though. The differences are that for a *site* you get a *Site type* field, a *Site dating* field group and a *Site positioning details* field group, while



for a *field collection event* you get a *Field collection type* field, a *Find summary* field group and a *Topographical coordinates* field group. If both a field collection event record and any underlying site records can be specified, then their hierarchical relation should be defined on the *Hierarchical relations* screen tab. On the *Related accessions* tab you can even link to an *Accession* record in which the excavated objects have been registered as an accession.

- In ***Transport***, fields were added for an exit authorisation date, an entry manager, an exit authorisation authoriser and an expected return method.
- In ***Location and containers***, location environment measures fields were added on the *Location details* screen. Also, a *Gap report* output format was added. This output format lists gaps in the repository so that you may know where there is sufficient space in the repository to place new items or containers. This report is especially helpful for archives as these organisations often store new accessions that consist of many containers and/or items that they would like to store in uninterrupted shelf space. See chapter 2.5 for more information about this.
- The *Research/use* data source has been renamed to ***Use of collections*** and has been expanded significantly to implement the Spectrum 5 [Use of collections procedure](#). It can be used to register how, when and where the objects or reproductions from your collection are used, the requests for any use and the results. Available types of use comprise exhibitions, enquiries, examinations, demonstrations, research projects, performances, filmings, online image publications and printed image publications. Fields were also added for reproduction references, digital references, user's (researcher's) contribution information, measurement fields (information from use), internal co-ordination fields like organiser and authoriser fields, request fields for venues and exhibitions and general fields for the use title, use status, use date, user role and user contact.
- An acquisition.inf and matching ***Acquisitions & acquisition proposals*** data source have been added to register the whole process regarding a planned or proposed acquisition, without having to register object records yet (although later object records can still be linked to an acquisition record).
- A new ***Collections review*** data source and matching database table have been added, to register a formal assessment of your objects collection. It implements the Spectrum 5 [Collections review procedure](#).



- A **Maps** data source and matching database table have been added. This is required by the GIS functionality (also see chapter 2.1). In here, you need to create a record for each base map and overlay you would like to make available to users. See chapter 2.2 for more information.

Map layer configuration	
Layer properties	
Type	base map
Name	OSM
Display name	Open Street Map
Format	default
Open layer map source	Open Street Maps
Style	blue
Opacity	100
Attribution	Copyright Open Street Map
Source identifier	
Source URL	
Projection properties	
Code	ESPG:3857

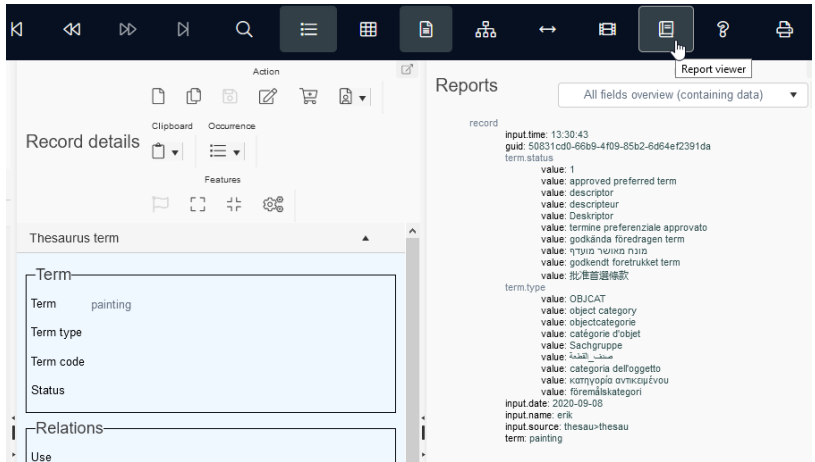


- The following fields do now automatically receive a unique number before storage, if no value has been entered manually:

Data source	Tag	Field label on screen
Assessments and treatments	RA	Reference number
Exhibitions	RN	Number
Events	EN	Number
Field collection	FR	Reference number
Orders	OL	Item ID
Use of collections	rN	Reference number
Collections Review	RN	Reference number
Transport	En	Transport number

- The *Catalogue*, *Object catalogue*, *Archives catalogue*, *Library catalogue* and *Accessions (bulk)* data sources now have a *Publish on the web* checkbox (tag `wp`) on the *Management details* tab. This checkbox is relevant for SAAS customers as a way to mark records that can be published by Axiell Internet Server 6.
- A *temporaryprinttags.inf* was added (without matching data source), containing only temporary tags Q1 up to and including QZ, to be used as (future) temporary tags by adapls, so that no longer temporary tags will have to be created for this use in other databases, which helps keeping them clutter-free. Currently, the tags from this database table are only used in *reminder.ada* and *CheckForDuplicateLoans.ada* though.
- All data sources now have an inline report, which is available as the *All fields overview (containing data)* output format as well, to show the contents of all fields with data in a single overview. One can see field group names, English field names as they have been specified in the database definition, link reference (*.lref*) fields pointing to a record number in the linked database and of enumerative fields one can see the neutral *value* (usually in capitals) and any of its interface language translations: only the neutral value is really stored in the record. Open or close the inline report via the *Report viewer* icon in the Collections top toolbar.





In most browsers you can use the **Ctrl+F** key combination to search for words in all currently displayed content in the browser. You can use that here too to find values or field names in the inline report (and in the *Records details* view if all screens tabs have been opened): a quick way to find a field or certain data in a large record.

- For the *Object catalogue*, *Persons and institutions*, the *Thesaurus*, *Exhibitions*, *Transport* and *Incoming loans* and *Outgoing loans*, English XSLT stylesheets are available which can be used as *Format* for [Saved search schedules](#), to lay out the result of a search executed by an Axiell SDI server.

SDIexhibitions.xslt

SDIloans.xslt

SDIobjects.xslt

SDIpeople.xslt

SDIthesaurus.xslt

SDItransports.xslt

You may use these as they are or build upon them and add or remove fields or translate the fixed *headertexts* in there to your own language. And in XSLT templates like the following, the number determines which translation is retrieved from an enumerative (aka drop-down list) field (where 0 means English, Dutch = 1, French = 2, German = 3, Arabic = 4, Italian = 5,



Greek = 6, Portuguese = 7, Russian = 8, Swedish = 9, Hebrew = 10, Danish = 11, Norwegian = 12, Finnish = 13 and Chinese = 14).

```
<xsl:template match="term.status">  
  <xsl:value-of select="./text[@language='0']"/>  
</xsl:template>
```

See the [Programming XSLT stylesheets for Adlib and Axiell](#) guide for more information about adjusting these stylesheets.

The same stylesheets (but then in a multilingual variety for the English, Dutch, French and German interface languages in Collections) are also available as output formats in the same data sources, to output a marked search result to a web page from which you can still print: then use the in-browser printer settings to have the printed pages “Shrink to fit” or print to landscape A4 format, otherwise the wide tables won’t fit on the printed pages. It concerns the following output formats: *Objects list (html)*, *Persons and institutions list (html)*, *Thesaurus terms list (html)*, *Exhibitions list (html)*, *Transports list (html)*, *Incoming loans list (html)*, *Outgoing loans list (html)* in their respective data sources.



2 Selected topics

2.1 GIS functionality and fields added

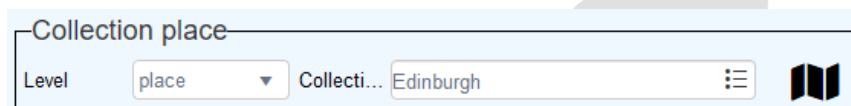
GIS (Geographic Information System) functionality in Collections 1.7 allows for an application to be customized to allow users to register locations and areas by means of their location on a geographical map and also to search for records by a variety of map manipulations to select places, precise spots or areas of different shapes.

In model applications 5.0, you can find such a GIS field (of datatype GeoJSON, with field tag jl) firstly in the new *Geographical thesaurus* data source, at the bottom of the *Geographical Thesaurus term* screen, labeled *Geolocation*.



You can also search on this field via the *Geolocation* access point in this data source. Because of redundancy, the grid type and coordinates fields were removed from the *Coordinates* box on the *Geographical Thesaurus term* screen.

Secondly, this geographical thesaurus GIS field has been made a read-only, merged-in field with the linked field *Collection place* (tag NF, linking to the geographical thesaurus) on the *Field collection* screen in the *Field collection* data source.



You can also search this merged-in field by means of a *Geolocation* access point in this data source.

So the idea is that if you want to register a map location, you would do that in the *Geographical thesaurus*, where you also name the location. Any link to such a named location from a *Field collection* record, will then retrieve the graphically represented map location as well (for viewing only).

Because the new GIS functionality deprecates the older geographical maps functionality, all fields which were set to data type *Geo location*



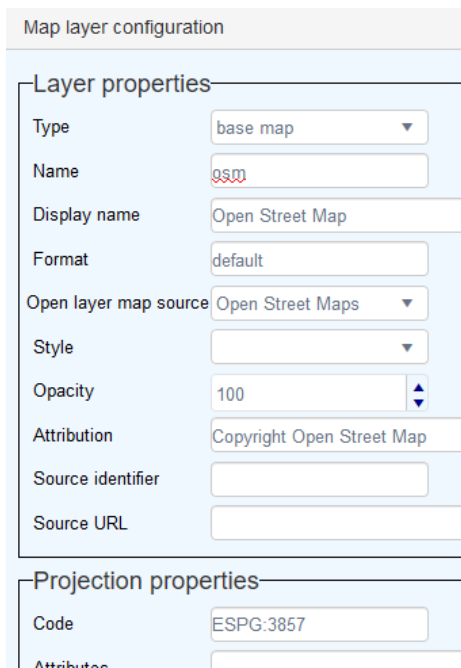
in model applications 4.5.2 have now been reset to data type *Text*, as they were before.

See the next chapter for more information about the required *Maps* records. And see the Collections [release notes 1.7 online help topic](#) for more information about how to use this functionality.

2.2 Maps records and fields

A *Maps* data source has been added. This is required by the GIS functionality (also see previous chapter). In here, you need to create a record for each base map and overlay you would like to make available to users. (Your base maps need to have been registered in the Collections *settings.xml* file already and the overlay files need to be present in the *json* folder in your Collections installation: see the [Collections installation guide](#) for more information about that.)

If not present by default, create at least a record for *Open Street Map*, with the following data. (Note that for this particular base map a *Source URL* is not required because the OpenLayers library used for the GIS functionality already knows where to find the relevant web service.)



Map layer configuration

Layer properties

Type: base map

Name: OSM

Display name: Open Street Map

Format: default

Open layer map source: Open Street Maps

Style:

Opacity: 100

Attribution: Copyright Open Street Map

Source identifier:

Source URL:

Projection properties

Code: ESPG:3857

Attributes:



See the following table for an explanation of all the fields you can find in a *Maps* record.

Field name in user-interface	Field details
<i>Associated base map</i>	This field is only visible in a <i>Maps</i> record (all the way at the bottom) if the <i>Type</i> field has been set to <i>overlay</i> . The <i>Associated base map</i> field can (optionally) be used to specify which base map the current overlay is associated with, if the overlay requires it.
<i>Style</i>	This field applies to overlays only and specifies the colour in which the overlay will be displayed.
<i>Format</i>	Use this field to specify the layer format for base maps of the WMTS type. For Open Street Map, enter <code>default</code> . For other maps, the map producer should have this information. Typical values are <code>image/jpeg</code> or <code>image/png</code> .
<i>Matrix ID</i>	These values must be provided by the producer of the WMTS base map or overlay. They are used in conjunction with <i>Grid resolutions</i> and specify the possible zoom levels and how much of the map is shown at each zoom level. They are not required for other types of maps or overlays. An example for a particular map producer would be the following list:



	<table border="1"> <thead> <tr> <th colspan="2">Tile properties</th> </tr> <tr> <th>Grid resolution</th> <th>Matrix ID</th> </tr> </thead> <tbody> <tr><td>1638.4</td><td>L00</td></tr> <tr><td>819.2</td><td>L01</td></tr> <tr><td>409.6</td><td>L02</td></tr> <tr><td>204.8</td><td>L03</td></tr> <tr><td>102.4</td><td>L04</td></tr> <tr><td>51.2</td><td>L05</td></tr> <tr><td>25.6</td><td>L06</td></tr> <tr><td>12.8</td><td>L07</td></tr> <tr><td>6.4</td><td>L08</td></tr> <tr><td>3.2</td><td>L09</td></tr> <tr><td>1.6</td><td>L10</td></tr> <tr><td>0.8</td><td>L11</td></tr> <tr><td>0.4</td><td>L12</td></tr> <tr><td>0.2</td><td>L13</td></tr> </tbody> </table>	Tile properties		Grid resolution	Matrix ID	1638.4	L00	819.2	L01	409.6	L02	204.8	L03	102.4	L04	51.2	L05	25.6	L06	12.8	L07	6.4	L08	3.2	L09	1.6	L10	0.8	L11	0.4	L12	0.2	L13
Tile properties																																	
Grid resolution	Matrix ID																																
1638.4	L00																																
819.2	L01																																
409.6	L02																																
204.8	L03																																
102.4	L04																																
51.2	L05																																
25.6	L06																																
12.8	L07																																
6.4	L08																																
3.2	L09																																
1.6	L10																																
0.8	L11																																
0.4	L12																																
0.2	L13																																
<i>Grid resolution</i>	<p>These values must be provided by the producer of the WMTS base map or overlay. They are used in conjunction with <i>Matrix IDs</i> and specify the possible zoom levels and how much of the map is shown at each zoom level. They are not required for other types of maps or overlays.</p>																																
<i>Source identifier</i>	<p>This is really an internal identifier for the layer - not strictly necessary at the moment and currently not used in Collections in any way.</p>																																
<i>Display name</i>	<p>This is the <i>Display name</i> for the layer as it is shown in the user interface of the map menu in Collections.</p>																																



<i>Open layer map source</i>	This is used in Collections to build the layer to display it in the user interface. Currently the only supported sources are <i>WMTS</i> , <i>Open Street Maps</i> and (GeoJSON) <i>Vector maps</i> for overlays.
<i>Name</i>	This is an internal name for the overlay or base map - not displayed in the user interface. Enter any descriptive name without spaces.
<i>Opacity</i>	Set the default opacity percentage for an overlay: this can still be changed in the map user interface though. A typical value is 50. Do not enter a percentage character.
<i>Extent</i>	Required where the open layer map source is <i>WMTS</i> and is concerned with how much of the world will or can be shown in the user interface. This information would be provided by the producer of the map. For non- <i>WMTS</i> layers that use longitude and latitude values, enter four occurrences of this field with the value 0: this works if the base map uses longitude and latitude values as well, like <i>Open Street Map</i> , so it can figure out the scope of what is visible itself. For non- <i>WMTS</i> layers that use UTM coordinates (values without decimals), enter four occurrences representing the extent (bounding box) expressed as UTM units – so, which coordinates your overlay fits within. You need this in conjunction with the UTM zone (in the <i>Attributes</i> field) in order to overlay the map in the correct location.
<i>Code</i>	This code represents a projection coordinate system and is used in Collections in the user interface: it would be provided by the producer of the map. The value for <i>Open Street Map</i> is <i>ESPG:3857</i> (note the



	<p>spelling error, but do not correct it to EPSG). If no code is available for a map or overlay, try <code>EPSG:29999</code>.</p>
<i>(Projection properties) Attributes</i>	<p>Required where the open layer map source is WMTS and is concerned with how much of the world will or can be shown in the user interface. This information would be provided by the producer of the map.</p> <p>For non-WMTS layers that use longitude and latitude values, enter <code>+proj=longlat +datum=WGS84 +no_defs</code> in the field.</p> <p>For non-WMTS layers that use UTM coordinates (values without decimals), enter <code>+proj=utm +zone=<the UTM zone> +ellps=GRS80 +unit=m +no_defs</code> in the field and replace <code><the UTM zone></code> by the actual UTM zone covered by the map.</p>
<i>Attribution</i>	<p>This is used to identify and credit the creators of the map - this could be any text or html - in some cases this must be a specific text, like for Open Street Map you should always enter <code>Copyright Open Street Map</code></p>
<i>Source url</i>	<p>Used in Collections in the case of WMTS or GeoJSON Vector layers to provide information on how to build the map, typically a URL (possibly including a license key) provided by the producer of the map.</p> <p>For downloaded overlays, fill in a relative (relative to the main Collections folder) path to the <code>json</code> folder in your Collections installation. Typically, this is something like: <code>./json/townships.json</code></p>
<i>Type</i>	<p>This mandatory field specifies whether the layer is a <i>base map</i> or an <i>overlay</i>.</p>



2.3 Linked Open Data and Persistent ID's

Axiell have implemented a solution for the management of Persistent Identifiers (PIDs) in Axiell Collections. The support for PIDs forms the basis of making the data of an organisation "Linked Open Data" (LOD). Each record that is made "Linked Open" will have its own unique PID that can be used to refer to it. This implementation covers three components that we'll describe in this document:

1. New data type: *URI*
2. New PID fields to store URI's and associated URL's.
3. Management of registration of PIDs with an external broker: Handle/SurfsARA - <https://www.surf.nl/en/services-and-support/purchasing-services-from-surfsara>

The application changes for LOD have been implemented in Collections application version 5.0. The core functionality in Axiell Collections has been implemented in version 1.6. The application changes could potentially also be ported back into customer specific and older application versions if needed.

While this implementation focusses around creating and managing PIDs, to make data "Linked Open" also requires exposing records via their PIDs in a format which other (usually) systems can understand. Exposing records is usually done in an RDF format (Resource Description Framework) for which there are various standards. An RDF implementation has not yet been done as part of this implementation but is likely to become available in the course of 2020 for collection items.

Making data "Linked Open" is not only relevant for objects in a collection, but also for descriptors and standardised terminology that is used to describe these objects. When open vocabulary resources are used to describe an object, such as the Getty AAT, the PIDs of the used terms/concepts can be stored in the Axiell Collections database. When publishing object records as Linked Open Data, these records will contain references to Getty AAT concepts. This not only means that it's clear what the meaning and context of each term/concept is, but it also allows for the creation of links with data from other datasets that are using the same identifiable concepts/terms as descriptors. E.g. if a keyword that is used to indicate the creation location is London, that would be an ambiguous descriptor as there are multiple places in the world that are called London. But if besides (or even instead of) the term London it's source ID would be listed (e.g. <http://vocab.getty.edu/tgn/7011781>) then it's clear which London is meant here. Therefore it's relevant to be able to manage



PIDs in all entities (record types) in the system. In some cases, the PIDs are the ID's of the organisation, in other cases PIDs from external parties are stored in the system.

New data type: URI

A new data type, *URI*, has been implemented. This data type is available in Axiell Designer from version 7.6. This URI data type is covering the following:

1. The content of a URI type field will be validated according to the .NET URI object rules as described in https://docs.microsoft.com/en-us/dotnet/api/system.uri-ctor?view=netframework-4.8#System.Uri_ctor_System.String. URI's usually have the form of a URL so if the URL is valid, it's also a valid URI.
2. The URI data type has the option to link to another URI type field. One will be defined as the external URI and one will be the internal URI. This is especially useful if PID/URI's are managed through an external broker like Handle/SurfSARA. The external URI will link to the external broker that will refer to the internal URI/URL. The idea behind this is to really keep the PID/URI persistent as it does not include an internal domain name of the organisation. So if the internal URL changes because of, for instance a change in the name of the organisation, the new internal URL can be linked to the external PID/URI. The internal URL could e.g. link to the Axiell WebAPI to present the record in a technical format or even in an HTML presentation.
3. The URI data type also allows for the definition of an initial value format which will be entered in the field automatically upon the creation of a record.

In Axiell Designer, a field definition of the URI data type gets a *URI field properties* tab where you can enter the relevant properties. See below screenshot for an example:

The screenshot shows the 'URI field properties' configuration window. It has several tabs: 'Field properties', 'Default values', 'URI field properties' (selected), 'Multi language fields', 'User interface texts', and 'Z'. The 'URI properties' section contains an 'Initial value format' field with the text 'https://hdl.handle.net/testcode/collect.%prirref%'. The 'LOD properties' section contains a 'URI type' dropdown menu set to 'External' and a 'Related URI field' text box containing 'PID-work-URL' with a search icon to its right.



The configured suffix that is used to make the URI unique in this example is *collect*. plus the record number (the contents of the *preref* (tag %0) field). In model application 5.0 we've chosen to use the contents of the new *guid* field (tag ZZ) as the unique suffix for the URI. The *guid* field is filled by a storage adapl (or actually by code in *force.inc*) with a GUID (a globally unique number).

New PID fields in Axiell Collections

In the standard Axiell Collections application version 5.0, PID fields have been implemented in all database tables, so not only for the objects database table but also for e.g. the Thesaurus and Persons & Institutions database tables. For the objects database table we have implemented three groups of PID fields: Work PID, Data PID and Other PIDs. See the screenshot further down in this paragraph. The PID fields have been configured on a new screen tab called *PID*, so it's easy to switch off for users that don't need to see these fields. The URL and URI fields are of the new URI data type and are inter-linked as internal (URL) and external (URI).

For each newly created record, a URI and URL are automatically generated. While we have configured in the field properties, it can (optionally) also be implemented via an adapl that reads a text file that contains default URI/URL settings. (Such an adapl is executed after storage and only kicks in if the relevant fields are still empty.) This way, this functionality is also available for customers that do not have access to Axiell Designer. In the screenshot below the base-URI for the Work-PID is:

<https://epic5.storage.surfsara.nl:8003/api/handles/21.T12995/>

The configured suffix (in the text file) is *collect* and the field that is used to make the URI unique is the record number/*preref* (tag %0). This is what a sample text file looks like (rows that start with * are comments):

```
* PIDtext.txt
* settings for PID creation
* use no space between a line number and the text behind it
*1 base uri, including institution specific code
1https://epic5.storage.surfsara.nl:8003/api/handles/21.T12995/
*
* table specific settings
*
* collect:
*
*2 suffix for uri for records in collect table, e.g. collect.
2collect.
*3 field, unique part of uri of collect table
```



3%0

* settings for URL creation in collect table

4https://ais.axiell.com/collect.

*5 field for unique part of url

5%0

*

The base-URI on row 1 has been provided by the Handle service of SurfSARA. For this example, the organisation code/prefix provided by Handle.net is: 21.T12995. This prefix was provided by Handle.net as part of a one-off registration process.

The settings of this text file must be done by the customer (or by Axiell) before the PID-implementation is taken into production.

The new *PID* screen tab:

The screenshot shows a web interface for configuring PIDs. It is divided into three main sections:

- PID work:** Contains fields for 'PID work URL' (https://ais.axiell.com/collect.1), 'PID work URI' (https://epic5.storage.surfsara.nl/8003/api/handles/21.T12995/collect.1), and 'PID work notes'. Below these is a table with columns: Action ID, Action, Status, Status date, Time, and Notes. The table contains one row with Action ID '1', Action 'CREATE', Status 'Synchronisation completed', Status date '2019-12-23', Time '16:27:17', and Notes 'Success'.
- PID data:** Contains fields for 'PID data URL' (https://www.njksmuseum.nl/collection/SK-C-5), 'PID data URI' (https://collection.njksmuseum.nl/doc/object/1), and 'PID data notes'.
- PID other:** A repeatable group containing two rows. Each row has fields for 'PID other URI', 'Qualifier', and 'Notes'. The first row has URI 'http://hdl.handle.net/11259/collection.38543', Qualifier 'alternative', and Notes 'Amsterdam Museum'. The second row has URI 'http://hdl.handle.net/10934/RM0001.COLLECT.5216', Qualifier 'obsolete', and Notes.

On the *PID* screen as shown above, we have configured fields for *Work* and *Data PIDs*, as well as a repeatable *Other PIDs* field group. The difference and use of the Work and Data PIDs can be a bit abstract and hard to grasp. The Work PID identifies real-world objects, e.g. works of art. The Data PID identifies metadata about a real-world object. It's possible to have more than one metadata record describing an object e.g. if another organisation also has described the object in their database that they want to refer to. In Other PIDs one can record e.g. Work PID's that have become obsolete (e.g. PIDs from an old or previous system) or other PIDs (from a different organisation) describing the same object (such as those from WikiData if they describe the same object). PIDs from a previous system that are still current should be entered into either the work



PID or data PID URI or URL field. To clarify the URL/URI you can enter a short comment in the Qualifier field, e.g. "previous URL".

Management of PIDs via Handle/SurfSARA

To ensure PIDs are truly PIDs, there are service providers that offer unique URL's that can be used as URI's. As these URI's don't contain the domain name of the organisation, they don't need to change if the organisation changes its name. It's these external URI's that are exposed/published. These external URI's need to be associated with 'real' URL's to make sure that when the URI is called, there is a valid response. This is where service providers like SurfSARA come in. They can host the external URI and its association with the real URL. This association can be managed via the SurfSARA API. It's this API implementation that we have done in Axiell Collections. When the URI is called, SurfSARA automatically refers to the internal URL.

To manage PIDs via SurfSARA, the organisation first needs to get a so called 'prefix'. This is a unique code for the organisation that is added to the URI and is the identifier of the organisation for SurfSARA. The one-off registration process at SurfSARA is quite complex and involves multiple steps. Most Axiell customers will not be able to do this by themselves. After this one-off process, the SurfSARA API can be used.

The organisation prefix can be obtained through Handle.net for a one-off fee. The services of SurfSARA are charged annually.

The PID implementation in Axiell Collections includes features that enable the collections manager to register the URI/URL combination at SurfSARA. For this, a *PID Registration Task* has been introduced in Collections model application 5.0. This task is run by the user on a selection of records that is ready for publication. The task creates registration records in a new data source called "Registration/Synchronisation Queue"; for each URI that needs to be registered a record is created or updated in this table. Possible actions: *Create, Update, Delete*. These registration queue records get an initial status 'to be synchronised'. They are linked to the object records so that the status is visible in those records.

A so-called trigger service has been implemented to run directly on the SQL table. The plug-in that Axiell has developed for this trigger service, detects that records are present in the "Registration/Synchronisation Queue" data source that need to be registered at SurfSARA. This plug-in communicates with the SurfSARA API for each record and performs the action as defined (*Create, Update, Delete*). When the registration is successful, the status in the registration record is updated accordingly. If the action fails, this is also recorded in the registration record.



The most common action would be to *Create* a URI/URL combination. If URL's change, e.g. because of a domain name change as a result of a change in the name of the organisation, the existing URI/URL mapping can be *Updated* to replace the old URL with the new URL.

The *Delete* action would normally not be used as this is rather contradictory to the idea of persistency. In a test environment, this action may have a use case though.

An example of a PID synchronisation record:

The screenshot shows a web interface for a 'Registration queue'. The form contains the following fields:

- Database/table: collect
- Record number: 1
- Action: CREATE (dropdown)
- URI: https://epic5.storage.surfsara.nl:8003/api/handles/21.T12995/collect.1
- URL: https://ais.axiell.com/collect.1
- Number of attempts: 1
- Registration status: Synchronisation completed (dropdown)

Below the form is a 'Registration status history' table:

Status	Date	Time	Notes
Synchronisation completed	2019-12-23	16:27:17	Success

Workflow

From the user perspective and considering that the initial configuration has been done, the workflow to register new PIDs would be as follows:

1. The user creates (object) records. The URI and URL fields (and the GUID field) are automatically filled when you save the record.
2. When records are ready for the PIDs to be published, the user selects these records and marks them in the result set.



- In the result set, the user chooses the *PID Registration Task*.

Object catalogue

Search Items Features

Change locations
PID Registration Task

	preref	medi	object_nu...	creator	title	dating d...	object_n...	
▶	✓	1	100001	Rembr...	Schutters van ...	1642	schilderij	Synchroni
▶	✓	2	100002					Synchroni
▶	✓	3	SK-C-6		De Staalmeest...		schilderij	Synchroni
▶	✓	4	112233		Portret van een...		schilderij	Synchroni
▶	✓	5	223344		Voorbeeldtitel		schilderij	Synchroni
▶	✓	6	334455		Landschap met...			Synchroni

- This task opens a new window enabling the user to choose the action that needs to take place for the selected records: *Create*, *Update* or *Delete*. The default and most used action would be *Create*.

PID Registration Task

PID registration action

Action CREATE

OK CANCEL

- After confirmation with *OK*, a message will appear confirming that the selected records have been put in the registration queue.
- The trigger service immediately starts processing the registrations with SurfSARA and updates the registration status. By refreshing the record list, the changed statuses will become visible.
- The PID-URI's are now ready to be used and they will refer to the associated URL's.

Notes

- It's no use entering a URI or URL in a new record manually, as they will be overwritten when you save the record. (In an already existing record you can edit these fields though.)
The assigning of a GUID doesn't depend on whether the record is new or existing but on whether the field is empty: only if it is



empty will it be assigned upon saving the record. The automatic URI and URL do depend on whether the record is new.

- When you copy a record which already contains PIDs and possibly a GUID, none of these fields and other fields on the *Permanent ID's* screen tab will be copied along because the GUID and PIDs need to be unique. (These fields are or should be non-exchangeable when it comes to copying records.) Upon saving the copied record will these fields be filled with new values.
- When you derive (move) a record from one data source to another, initially all data from the *Permanent ID's* screen tab will be moved along (because it's a move, not a copy), but since Collections considers it to be a new record, the record will get new URIs and URLs. Any GUID will remain the same.

2.4 The “group” term status (neutral value 6)

The *group* status is intended for terms which are useful or important in your thesaurus management to distinguish term collections which themselves can be part of the term hierarchy but are not allowed to be used as a term in catalogue records.

An example might illustrate this concept. Suppose that in your *Thesaurus* you'd like to construct a term hierarchy of body armour:

armour

armour (by construction)

mail armour

plate armour

scale armour

armour (by form)

cuirass

gambeson

gauntlet

helmet

leg guard

shoulder plate

armour (by function)

battle armour

horse armour

parade armour

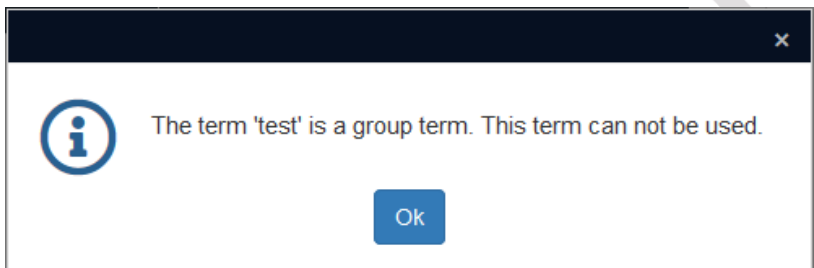
Although the terms *armour (by construction)*, *armour (by form)* and *armour (by function)* come in handy here in the term hierarchy, you would probably not want anyone to link to such a term from within the catalogue (contrary to the other terms). Precisely for this purpose you



could assign the *group* status to the mentioned three terms in this example.

Terms which have been assigned the *group* status, adhere to the following specific rules:

- Within the Thesaurus you can link to such a term as a broader or narrower term so you can include it in a term hierarchy.
- Terms with this status will be indexed in the thesaurus and you can search on them normally.
- In catalogue records which link to the thesaurus you cannot store terms with the *group* status in linked fields.
- In linked fields in catalogue records which link to the *Thesaurus* you can use the *Find data for the field* window to look for terms with the *group* status. In the list of found terms, terms with the *group* status are not marked in any special way but when you try to *Select* such a term you'll be notified that that is not possible: only non-*group* terms can actually be copied to the linked field.



- Avoid using terms with this status in preferred term relations: you won't be using them in the catalogue anyway.

2.5 A location *Gap report*

In the *Location and containers* data source, a new *Gap report* output format was added.

This output format lists gaps in the repository so that you may know where there is sufficient space in the repository to place new items or containers.

GAP REPORT FOR: CCMC | 30002 2020-09-08

Start of gap	End of gap	Available space
CCMC-C1-A-B1-U1-S1	CCMC-C1-A-B1-U1-S1	0.8
CCMC-C1-A-B1-U1-S2	CCMC-C1-A-B1-U1-S3	1.25
CCMC-C1-A-B1-U2-S1	CCMC-C1-A-B1-U2-S1	1
CCMC-C1-A-B1-U2-S2	CCMC-C1-A-B1-U2-S2	0.8
CCMC-C1-A-B1-U2-S3	CCMC-C1-A-B1-U2-S3	0.5
CCMC-C1-A-B1-U2-S4	CCMC-C1-A-B1-U3-S1	1.7
CCMC-C1-A-B1-U3-S2	CCMC-C1-A-B1-U3-S3	1.2
CCMC-C1-A-B1-U3-S4	CCMC-C1-A-B2-U1-S1	1.05
CCMC-C1-A-B2-U1-S2	CCMC-C1-A-B2-U1-S3	1.5
CCMC-C1-A-B2-U1-S4	CCMC-C1-A-B2-U1-S4	0.5
CCMC-C1-A-B2-U2-S1	CCMC-C1-A-B2-U2-S1	0.5
CCMC-C1-A-B2-U2-S2	CCMC-C1-A-B3-U1-S2	8.5
CCMC-C1-A-B3-U1-S3	CCMC-C1-A-B3-U1-S4	1.9
CCMC-C1-A-B3-U2-S1	CCMC-C1-A-B3-U3-S1	4.5
CCMC-C1-A-B3-U3-S2	CCMC-C1-A-B3-U3-S3	1.9
CCMC-C1-B-B1-U1-S2	CCMC-C1-B-B1-U3-S2	9
CCMC-C1-B-B2-U1-S2	CCMC-C1-B-B2-U1-S2	1
CCMC-C1-B-B2-U1-S3	CCMC-C1-B-B2-U2-S4	5.5
CCMC-C1-B-B2-U3-S2	CCMC-C1-B-B3-U1-S1	4
CCMC-C1-B-B3-U1-S2	CCMC-C1-B-B3-U2-S1	3.5
CCMC-C1-B-B3-U2-S2	CCMC-C1-B-B3-U2-S3	1.5
CCMC-C1-B-B3-U2-S4	CCMC-C1-B-B3-U3-S3	3.5
CCMC-C2-A-B1-U1-S1	CCMC-C3-A-B1-U1-S1	2



This report is especially helpful for archives as these organisations often store new accessions that consist of many containers and/or items that they would like to store in uninterrupted shelf space.

The *Gap report* identifies shelf space that is not in use and reports on consecutive linear gaps. It does that by checking all the shelf locations underneath the main location that the report must process. The report lists ranges of shelf locations accompanied by the linear size of the gap for each range. A range is basically a start location and an end location. For each shelf, the report knows its linear capacity and for each container or unpacked object it knows the linear space it occupies on the shelf, **if** these details have been registered for all relevant objects, containers and locations.

Location details	
Location details	
Location type	
Level	Shelf
Linear capacity 1	

So it is required that for shelf location records the *Linear capacity* field on the *Location details* tab has been filled in (the *Level* name is irrelevant for this report though), that all container records (in the same database) have a *Container type* filled in and that the *Container shelf space* on the *Container type details* tab in the *Container types* data source has been filled in and that in the *Catalogue* for all objects and archival items likely to be stored in the repository the *Linear space on shelf* field on the *Location | Future movements* tab has been filled in. There is no dedicated unit (metre, inch, foot) assigned as this may be different per organisation. The only thing that is vital here, is that you use the same implied unit across these fields in the relevant databases. If these details are missing from certain records, the gap calculation will be wrong for the relevant locations but you will be notified of the problematic records (by their record number and database) in the gap report itself, to make it easy to find and correct them.

Always run the *Gap report* for a **single** selected location. Typically you would choose a location higher up in the hierarchy, like a building or a bay, but do realize that running the report takes (much) longer for larger hierarchies. For the calculations, the report only looks at the



hierarchical endpoint locations in the hierarchy underneath the selected main location, usually these are shelf records but they can be named differently too: the locations which are used to assign to containers and unpacked objects.

Note that available space is calculated for the *Default/Normal/Home* location of containers and objects, **not** for their current locations which could be different and temporary. So object and container records which do not have a normal location won't be included in the *Gap report* calculations.

The way an uninterrupted gap is calculated is by looking if a location is used by either a container or an object from the catalogue. The report checks both. A mix of containers and unpacked objects could be on a shelf and their linear spaces would be added up and deducted from the shelf capacity. If a shelf is not filled up to capacity, it will be the start of a gap range. The gap ends when a location is found that is used. That location will not be included in the gap range as it will be assumed that the items on that location are placed at the start of the location. An uninterrupted space may continue on the next shelf, even in the next upright, bay or room etc.

Further considerations:

- Some archives may have shelves that are designated for double stacking or double backing. This doubles the capacity of the shelf. This can simply be met by doubling the linear shelf capacity in the location record.
- An endpoint of a location hierarchy could also be a drawer. A drawer may be designated to hold e.g. maps and may have a specific capacity for a number of maps. The *Gap report* could be 'tricked' in handling this by using the linear capacity of the drawer as the number of maps it can hold. E.g. the linear capacity of a drawer location could be 10 and each unpacked map record (unpacked object) could have a 'linear' capacity of 1. If the drawer only has 4 maps assigned to it, the *Gap report* will then show 6 as available gap. Reporting on drawer gaps should not include shelves in the hierarchy of the selected top node as the reported uninterrupted gap might then contain a mix of shelves and drawers which would not be useful.