

---

# Adlib model applications 4.5.2

## release notes



## Axiell ALM Netherlands BV

---

Copyright © 2018 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 Introduction	1
2 Improvements	3
2.1 Integrity check compliance	3
2.2 A customizable <i>Change locations</i> procedure	3
2.3 A customizable <i>Move records</i> procedure	4
2.4 Geographical maps functionality	5
2.5 Reformatted print templates and adapls	7
2.6 Simple search options	8
2.7 Bulk creating records	9
2.8 Bulk linking records	10
2.9 A location context column in the link window	11
2.10 Original linked file names	11
2.11 Pseudonyms	12
2.12 The <i>Related records</i> view	14
2.13 Inherited fields in Archive data sources	15
2.14 EAD export format changes	16
2.15 Broken reverse location/collect link	17



# 1 Introduction

Version 4.5.2 of the Adlib model applications is the first to be fully Axiell Collections-ready as well as completely Adlib for Windows compatible. These applications can be run in either or both environments equally well. However, with Axiell Collections offering more and more core software functionality that is not available in Adlib for Windows, most application development efforts for version 4.5.2 have been put into applying this extra functionality in the model applications in the best possible way.

Note that even though a lot of application changes in 4.5.2 only have effect in Axiell Collections, you need Adlib executables version 7.5 or higher to run model applications 4.5.2 in Adlib for Windows. Of Axiell Collections you should always use the latest version available.



## 2 Improvements

### 2.1 Integrity check compliance

Version 4.5.2 of the model applications is the first to pass the stringent application and database structure requirements of Axiell Collections, ensuring more consistently structured, and therefore safer and less error-prone, applications.

### 2.2 A customizable *Change locations* procedure

Contrary to the hard-coded *Change locations* procedure in Adlib for Windows, Axiell Collections uses a customizable setup in the form of a task. The [Change locations task](#) as it is set up for the four sub data sources of the *Objects* catalogue in model 4.5.2 specifically for Axiell Collections, consists of the *changelocationtask.fmt* screen file and *changelocation.ada/.bin* adapl files which by default offer exactly the same functionality as the hard-coded *Change locations* procedure in Adlib for Windows (still) does, namely to allow the user to change the current location of objects registered in one or more marked records in the result set all at once and update the location history of those object records at the same time. However, this screen and adapl can in principle be adjusted to your liking, for instance if you'd like to be able to process future movements as well: a nice option to have.

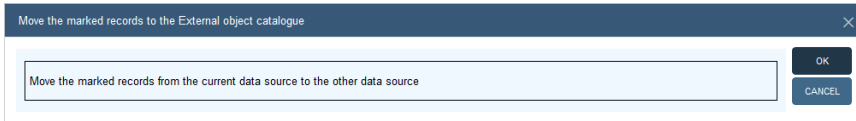
The screenshot shows a 'Change locations' dialog box with the following fields:

- New location section:**
  - To normal location
  - Location: [Text field]
  - Date: [Text field with calendar icon] Time: [Text field with clock icon]
  - Executor: [Text field]
  - Suitability: [Text field]
  - Authoriser: [Text field]
  - Notes: [Text field]
- Movement section:**
  - Method: [Text field]
  - Reference no.: [Text field]
  - Contact: [Text field]
  - Notes: [Text field]

Buttons: OK, CANCEL

## 2.3 A customizable *Move records* procedure

As Axiell Collections doesn't have a hard-coded *Derive* procedure like Adlib for Windows does, Collections uses a customizable setup in the form of a task which actually does the opposite and moves or copies marked records from the current data source to another data source. The task as it is set up for all sub data sources, except for the *Archives (catalogue)*, of the *Objects* catalogue in model 4.5.2 specifically for Axiell Collections, consists of the *confirmmovingofrecords.fmt* screen file (which is just a confirmation message and has no fields) and *moverecordstoootherdataset.adapl.bin* adapl files. For the *Internal object catalogue*, the task is called *Move the marked records to the External object catalogue*, for the *External object catalogue* the task is called *Move the marked records to the Internal object catalogue*, while for *Archives (accessions)* the task is called *Copy the marked records to the Archives (catalogue)*.



So the one adapl implements different functionality for the three data sources: when a record is moved, the original will be deleted, while the copying of an accession record leaves the original intact. A resulting copy in *Archives (catalogue)* will not be an exact copy though:

- The accession number will be saved in the related accession field of the created catalogue record.
- The reference code of the created catalogue record will be saved empty, so the user must fill in this field manually afterwards.
- Management details of the accession won't be copied, just like the accession date, transfer method, purchase price, purchase price currency, transfer reason, transfer source type, transfer source, parent level accession, parent level accession notes, accrual, accrual notes, cataloguing priority, accession status, geographical importance, likely demand, accession category, total processing time, process job, target date, completion date and duration.

This screen and adapl can in principle be adjusted to your liking, for instance if you'd like to copy a record instead of moving it or if you would like to change which fields are copied and which are not.

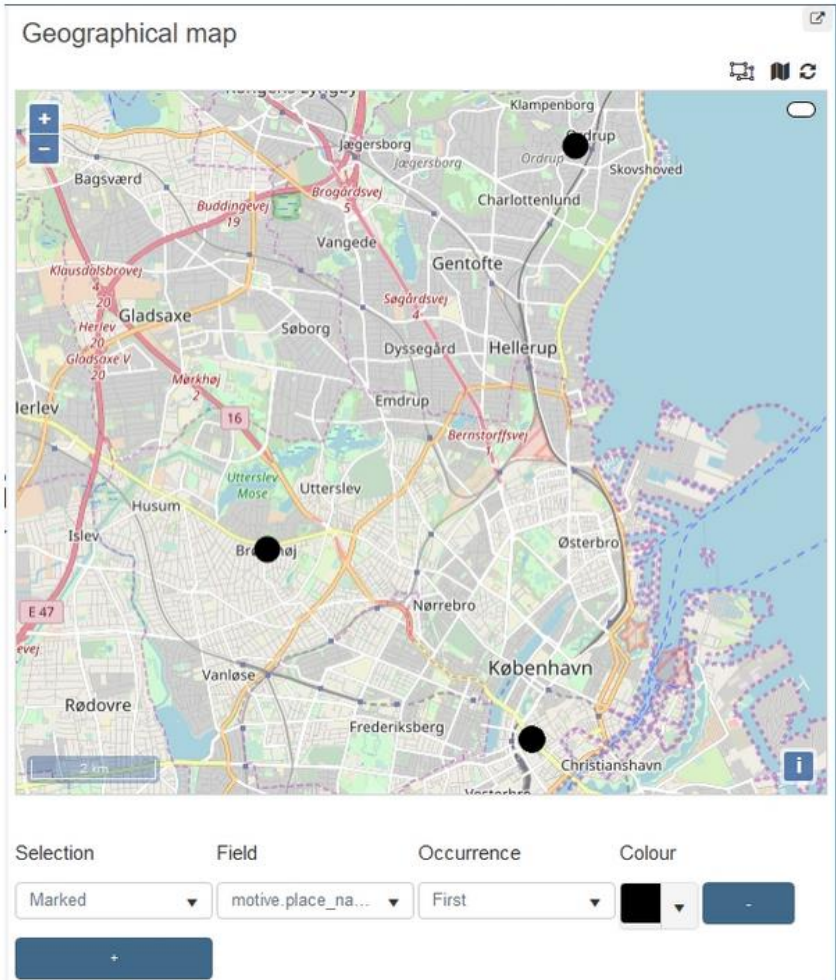
Note that in Axiell Collections, tasks can be opened from the result set context toolbar:





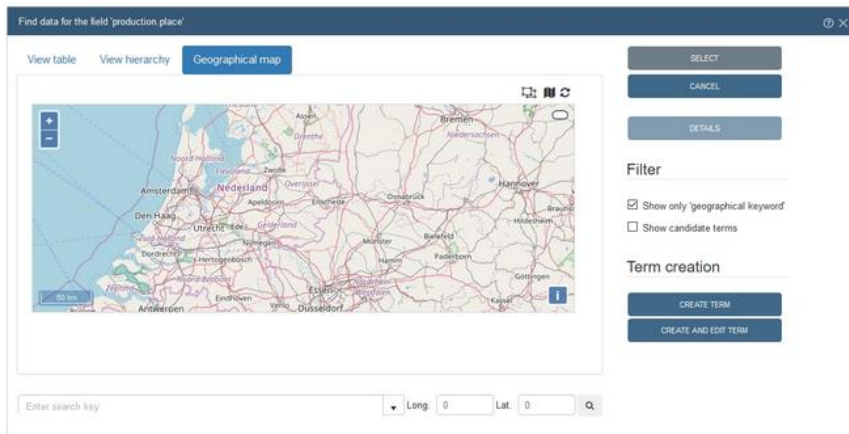
## 2.4 Geographical maps functionality

For enabled fields, Axiell Collections offers geographical map functionality in the form of a [Geographical map view](#) (next to the *Record details* view, the *Result set* and the *Media viewer* and so on).



The view is meant to display locations (as registered in these fields in your records) on a world map for a quick visual overview, the geographical distribution of production places of a certain record selection for example.

Also, the [Find data for the field](#) dialog for linked geographical place fields will display an extra *Geographical map* tab which allows you to enter data into the field by selecting a place on the map.



In model applications 4.5.2, this functionality has been set up for the following fields only:

Database	Field name	Field tag
Collect	production.place	VP
	acquisition.place	PX
	field_coll.place	NF
	owner_hist.place	P9
Document	place_of_publication	pl
	geographical_keyword	GT
People	birth.place	n3
	death.place	n5
	address.place	BL
Exhibition	organiser.place	tL
	venue.place	TP
Research	researcher.place	tL

---

## 2.5 Reformatted print templates and adapls

Printing from Axiell Collections is different from Adlib for Windows. The bad news is that Collections does not support output adapls using `print` and `output` statements to print stuff, nor does it support Word template output formats with the `.dot` and `.dotx` extension. The good news is that Collections does support Word templates with the `.docx` extension, it supports output adapls using the `wordcreatedocument` function (to print from a `.docx` template), it supports output formats which have been specified as a combination of an adapl and a `.docx` template, and it also supports XSLT stylesheets as output formats. By the way, the new `.docx` format is also supported by Adlib for Windows 7.5 or higher.

For model applications 4.5.2, all museum `.dot` templates have been converted to the `.docx` format, allowing you to print to these templates from within Axiell Collections as well as from Adlib for Windows. These are the three *Brief object print (x objects per page)* and *Object ID* template output formats for object data sources, the two *Extended location record listing* template output formats for the *Location and containers* data source and all templates (strictly speaking not output formats) which are being used in the loans data sources to generate letters. Output formats which are compatible with Adlib for Windows only, will still be available in there and simply won't show up in Axiell Collections.

Some plain text print adapls have been recreated, either as `.docx` template or as a combination of a rewritten adapl plus `.docx` output format, to be compatible with both Axiell Collections and Adlib for Windows. These are the *Inventory list* output format for object data sources and the *ISBD listing, Short list (title and copies), Extended format, ISBD listing, incl. analytical cataloguing* and *Accession list* for most library data sources.

### Letter generation in Transport and Loans

In the *Transport/Despatch* and *Incoming loans* and *Outgoing loans* data sources, there are a number of *Template* checkboxes which, once clicked, generate a Word document, a request or freight letter for example, in the selected language, after which the path to the created document is automatically being registered in the *Digital document* field.

Request confirmation

Template English

Date   Dig. doc.

This is handled by two after-field adapls: *loanproc.ada/bin* and *tranproc.ada/bin*. This is not new functionality, but the adapls had to be reprogrammed somewhat to deal with the fact that in Axiell Collections an adapl can't display a confirmation message on screen. This has the following result: for the relevant checkboxes, *loanproc.ada/bin* can present the following questions to users in Adlib for Windows, while in Collections they are skipped and assumed confirmed:

- *Do you want to add the objects (excl. refused/withdrawn) to the exhibition record?*
- *Do you want to create a Despatch record for these objects (excl. refused/withdrawn)?*
- *Do you want to create an entry record for these objects (excl. refused/withdrawn)?*
- *This Word document has been ceated before. Do you still want to create a new document?*

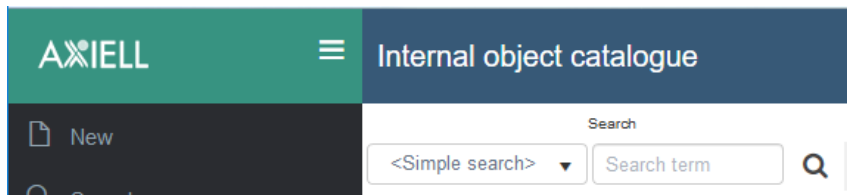
To deal with the automatic confirmation of the fourth question in an intelligent way, the adapl has been changed to prevent earlier created Word documents from being overwritten by default. The sub routine does overwrite the contents of the field but the document it creates will get a unique name so that the old document won't be overwritten.

For the relevant checkbox, *tranproc.ada* can present the following question to users in Adlib for Windows, while in Collections it is skipped and assumed confirmed:

- *Do you want to update the current location for the linked objects with the location you have entered here? Note: If you have not entered a date and/or time here, the current date/time will be used.*

---

## 2.6 Simple search options

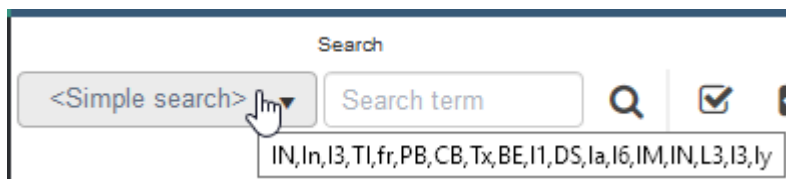


In the result set context toolbar in Axiell Collections, you have the option to quickly execute a new search in the current data source, by selecting a field from the *Search* drop-down list of all indexed fields in this database, after which you simply enter your partial (truncation

with \* required) or whole search key in the *Search term* box and click the magnifying glass.

At the top of the list of indexed fields, all data sources in 4.5.2 have a special simple search option, appropriately called *<Simple search>*, which searches multiple commonly used fields that have an index, at once: use *<Simple search>* if you're not sure how to find what you're looking for.

If you're curious to find out which fields will be searched in the currently opened data source, then hover the mouse cursor over *<Simple search>* and a tooltip listing all searched field tags will appear. Via Adlib Designer you could then look up the field names associated with these tags, in the relevant database structure (.inf) file.



## 2.7 Bulk creating records

For Axiell Collections only, the *Internal* and *External object catalogue* and the *Archives (catalogue)* data sources in model 4.5.2 offer an option to quickly [create a batch of records all at once](#). (In other data sources, clicking the *Bulk create* icon will allow the user to create just a single new record.)



For object records, an *Identification | Production* and *Physical characteristics* screen (*objbulk.fmt* and *fysic\_bulk.fmt*) with the most commonly used fields are available, while for archive catalogue records only a single screen (*arch\_id\_bulk.fmt*) with often used fields has been implemented. In the left upper corner of the *Bulk create* dialog there's a *Count* box, in which the user must specify the number of records to create. The created records will have identical data, except for uniquely indexed fields which data will automatically be made unique with a sequential number. After the records have been created, the user can of course edit them one by one to add other data unique to each record.

Bulk create records

Count: 1

Identification | Production

Identification

Institution name

Object number

Part  Tot. number

Distinguishing featu...

Object name

Object category

Object name

Other name

Name type

Title and description

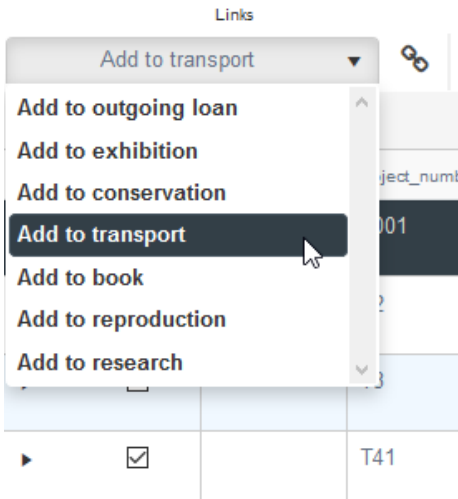
Title

OK

CANCEL

## 2.8 Bulk linking records

In Axiell Collections, in the result set context toolbar of the *Internal object catalogue*, the *External object catalogue*, *Books*, *Multimedia documentation*, *Transport*, *Incoming loans*, *Outgoing loans*, *Assessments and treatments*, *Exhibitions*, and the *Research/use* data sources of model application 4.5.2, you'll find a *Links* drop-down list plus chain icon. Per data source it varies which link options are available in the drop-down list.



The functionality allows the user to link one or more marked (source) records in the current result set to one or more existing (target) records, or a new target record, in another data source (so the links will be registered in the target records in the other data source). The user must simply mark the records he'd like to link, select the desired option from the drop-down list and click the chain icon. A *Search* dialog will open, allowing him to search for one or more target records to which the current record selection must be linked: the user must then mark the desired found records in the separate result set window and click *OK* or click the *Create* button to create a new target record to which the marked source records will be linked.

## 2.9 A location context column in the link window

The screenshot shows a window titled "Find data for the field 'current\_location.name'". It has two tabs: "View table" (selected) and "View hierarchy". The table contains the following data:

Name	Barcode	Type	Context
shelf A	L001	LOCATION	Main building/Room A/Bay 1/Vertical A
Shelf A	L012	LOCATION	Main building/Room A/Bay 1/Vertical B
Shelf A	L017	LOCATION	Main building/Room A/Bay 2/Vertical A
Shelf A	L019	LOCATION	Main building/Room A/Bay 2/Vertical B

Below the table is a search input field containing "Shelf A". To the right of the table are buttons for "SELECT", "CANCEL", and "DETAILS". Below these is a "Filter" section with a checkbox for "Show candidate terms" and a "Term creation" section with buttons for "CREATE TERM" and "CREATE AND EDIT TERM".

The link screen (*Ink\_location.fmt*) for linked location fields (used for the *View table* tab in the *Find data for the field* window when looking up or validating an entered value in a location field in an object or location record or in the *Change locations* procedure), now contains a *Context* field column showing the upwards hierarchy of the found locations. This makes it easier to select the proper location when you have a lot of non-unique location names.

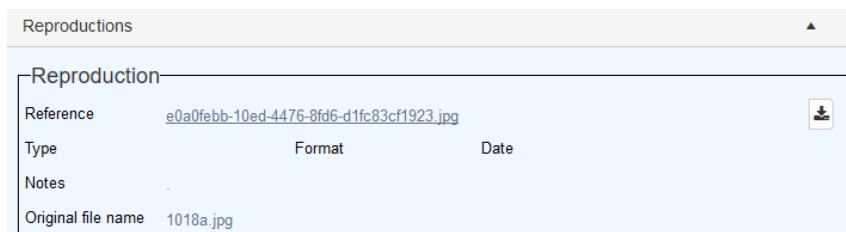
The improvement is visible in both Axiell Collections and Adlib for Windows.


## 2.10 Original linked file names

When you link an image file to a record, the selected image file will be copied from the original location to the local *\images* subfolder of your

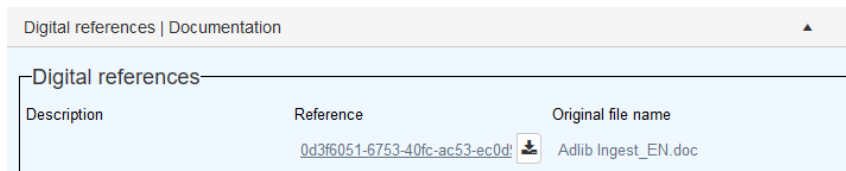
Axiell Collections folder for this purpose and will then be assigned a new file name to prevent it from accidentally overwriting any earlier linked files with the same name. This new, unique file name will then automatically be registered in the media record.

However, this file name is not based on the original file name and maybe you would like to register this original file name in the media record too, because the name contains an indication of what the image portrays or because you'd like to be able to search on the file name. With model application 4.5.2, Axiell Collections will now automatically register the original file name in the media record too, whenever you link a new image file anywhere in the application. The original file name is also a read-only merged-in field with linked image fields in catalogue records.



Reproduction		
Reference	<a href="#">e0a0febb-10ed-4476-8fd6-d1fc83cf1923.jpg</a> 	
Type	Format	Date
Notes		
Original file name	1018a.jpg	

Files uploaded as digital references get a new file name too, so their original file name might be relevant too. Since digital reference fields are not linked fields, the original file name will be stored in a new read-only field in the relevant catalogue record, next to the digital reference itself.



Digital references   Documentation		
Digital references		
Description	Reference	Original file name
	<a href="#">0d3f6051-6753-40fc-ac53-ec0d!</a> 	Adlib Ingest_EN.doc


## 2.11 Pseudonyms

Pseudonyms are a new internal link type in Axiell Collections and Adlib for Windows 7.5, implemented in model applications 4.5.2. In a relation of this type you can associate proper names (aka "main" names) with pseudonyms. This allows you to register e.g. the proper name of an author as well as his or her pseudonym(s) in the *Persons and institutions* data source and link these names to each other in a way that doesn't prefer one name over the other. This type of relationship is somewhere in between a preferred term relation and an equivalent



relation, because between names in a pseudonym relationship a hierarchy does exist but no non-preferred term substitution will ever take place. An equivalent relation has no hierarchy.

When you create a pseudonym relation, a reciprocal record will automatically be created, just like with preferred term relations. Here, *Pseudonym* is a repeated field, while *Pseudonym for* is not repeatable and you cannot fill in both fields.

Identification	
<b>Authorised form of name</b>	
Name	<input type="text" value="Hoofdakker, Rutger Hendrik van den"/>
Name type	<input type="text" value="author"/>
Reference number	<input type="text"/>
Name note	<input type="text"/>
<b>Other forms of name</b>	
Use	<input type="text"/>
Used for	<input type="text"/>
Pseudonym for	<input type="text"/>
Pseudonym	<input type="text" value="Kopland, Rutger"/>
Equivalent	<input type="text"/> 

The *Standard* and *Advanced* search in Axiell Collections allow the use of a new `pseudonym` operator when you search *Persons and institutions* or when you search a field in another data source, linking to *Persons and institutions*. When you are searching using `pseudonym`, you search on the proper name and all its pseudonyms as specified in the *Pseudonym* entry field (in *Persons and institutions*), provided these names have the same domain (name type) as the linked field you are searching. It doesn't matter if the name you enter is a proper name or pseudonym and it also doesn't matter if the search key actually does appear in records for the search to succeed on the other names in the pseudonym relation. For example:

```
author.name pseudonym "Kopland, Rutger"
```

In this example all records would be retrieved in which the author name is either the search key itself or the proper name or any pseudonym of the search key.

In Adlib for Windows on the other hand, this is implemented in a more limited way: mark the *Use relations* checkbox before a search to expand your search on a name to its pseudonyms (the other way around isn't implemented and the `pseudonym` operator isn't available).

When you've actually registered pseudonyms in *Persons and institutions* records and you try to validate field data linking to *Persons and institutions*, you may encounter the following icons in the first column of the list on the *View table* tab of the *Find data for the field* window:

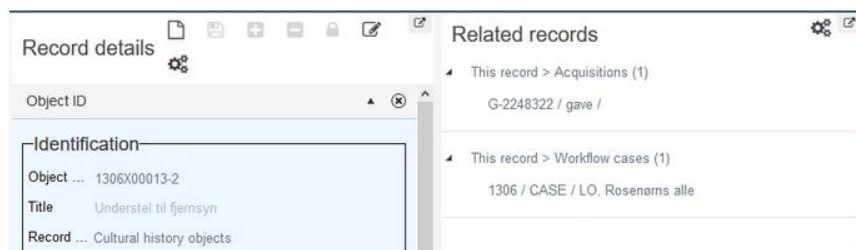


A grey star indicates a proper name while an open star indicates a pseudonym. Pseudonyms can be registered in linked fields just like proper names: there won't be any automatic substitution of names.

---

## 2.12 The *Related records* view

The [Related records view](#) in Axiell Collections provides an alternate overview of all linked records in one or more linked fields, grouped per linked data source, while in the linked data source it provides an overview of all records in that data source linking to the current record.



The *Related records* view is only visible in a data source if at least one (single-sided or reversely) linked field from or to the underlying database has been set up for this view. After configuration, the actually displayed relationship is only valid for the specifically set up linked field.

For model application 4.5.2, many linked fields have been set up for this view.

## 2.13 Inherited fields in Archive data sources

The fields *access\_category.notes* (tag F2, the *Conditions governing access* entry field on the *Condition of access and use (ISAD)* screen) and *rights.notes* (tag RO, the *Conditions governing reproduction* entry field on the *Condition of access and use (ISAD)* screen) have been added as [inheritable fields](#) to the two archive data sources of model applications 4.5.2. For both Axiell Collections and Adlib for Windows, this means that data from these fields from the first record higher in the hierarchy in which this field has actually been filled in will be displayed automatically in the same field in the current record (if it is still empty), allowing the user to copy that data easily if desired.

The screenshot shows a web interface for 'Conditions of access and use (ISAD)'. It is divided into two main sections: 'Conditions governing access' and 'Conditions governing reproduction'. Under 'Conditions governing access', there are two fields: 'Period closed...' and 'Closed ...'. Under 'Conditions governing reproduction', there is a note: 'These are inheritable/inherited conditions'. The text in the reproduction section is greyed out, indicating it is inherited data.

Data inherited from a record higher up in the hierarchy will initially be displayed greyed out in the current record, if the field was still empty. Even though the copied content is displayed in the record, it is not yet part of the current record. To save the copied contents, put the record in edit mode and simply double-click the relevant field or put the cursor in the field and start typing new text and/or delete copied text to fully activate the field contents (the text colour changes to normal). Now saving the record includes the activated field contents: a quick way to (partially) copy data from other records! Remember that typing new text in such fields is possible as well, so you're not stuck with the copied text.

If you don't want to duplicate data from the higher records at all, you'll probably never activate the data and will just enjoy the fact that data from parent records is conveniently visible in the edited record. Do observe that inherited data may come from different elders: one field might display data from the direct parent record, while for another field the direct parent doesn't have any information while its grand-parent does.

---

## 2.14 EAD export format changes

The EAD export format, as used in the *Archive (catalogue)* data source, has been changed somewhat, plus it now functions in combination with an `adapl` that is executed for every exported record, just before record data is transformed to the EAD format by the XSLT stylesheet. The `adapl` retrieves some extra data from linked records, which will then end up in the EAD export result. The changes are the following:

- The header `<eadid>` is now filled with the *Reference code* (tag `IN`) of the fonds level record and the `identifier` attribute of that node now contains a placeholder.
- The header `<titleproper>` is now filled with the fonds level title, instead of remaining empty; the empty date node inside it has been removed.
- The header creation `<date>` is now filled with the export date.
- For the `archdesc <unitid>` element, the `repositorycode` attribute is now filled with the institution code, while the `identifier` attribute is now filled with the reference code of the processed record.
- The `archdesc <unitdate>` element now has a `normal` attribute which is filled with the normalized date range, using `production.date.start` and `production.date.end`, separated by a forward slash.
- The *Term code* of the inscription language term record in the thesaurus will be used as the `langcode` attribute for the `langmaterial <language>` node. So for the *English* thesaurus record for example, the *Term code* field should contain the value `eng`.
- Geographical (place) names from the `content.subject` field will now be complemented with their broader term (a country name for example) from the thesaurus record. So the EAD `geogname <emph>` tag will now contain strings like *Amsterdam, Netherlands* if *Netherlands* is the broader term of the geographical name *Amsterdam*.

---

## 2.15 Broken reverse location/collect link

In 4.5.2, the reverse link between `current_location.name` (tag 2A) in the *collect* database and `object.object_number` field (tag IN) in the *location* database has been broken. `current_location.name` still links to the *location* database, so that in object records a link to a location can be registered, but the other way around – links from a location record to all objects residing at that location – is no longer possible. The reason is that location records with hundreds or thousands of linked object records cause performance issues when loading or displaying such location records.

`object.object_number` has not been removed from the *location* database but is now a normal field with access rights *None* for role *\$REST*, so that the old reverse link setup might be restored easily if required.

To compensate for the broken link, the *Related records* view for the *Location and containers* data source in Axiell Collections lists all object records registered on the currently selected location.