

adlib



model applications 3.4
release notes

Adlib Information Systems

Copyright © 1992-2006 Adlib Information Systems B.V.® All rights reserved. Adlib® is a product of Adlib Information Systems B.V.®

The information in this document is subject to change without notice and should not be construed as a commitment by Adlib Information Systems. Adlib 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 Improvements in 3.4 applications	5
1.1 Unicode UTF-8.....	5
1.2 French and Greek	5
1.3 Reorganization of folders and files.....	5
1.4 Adlib Museum	6
1.4.1 Improvements in files	6
1.4.2 Improvements in terminology	6
1.4.3 Solved problems.....	6
1.5 Adlib Museum Basis	7
1.5.1 Improvements in files	7
1.5.2 Improvements in terminology	7
1.5.3 Solved problems.....	8
1.6 Adlib Library	8
1.6.1 Improvements in files	8
1.6.2 Improvements in terminology	8
1.6.3 Solved problems.....	9
1.7 Adlib Archive	9
1.7.1 Improvements in files	9
1.7.2 Improvements in terminology	10
1.7.3 Solved problems.....	10
1.8 Adlib Loans	10
1.8.1 The improved remind procedure	10
1.8.2 Improvements in files	16
1.9 Adlib Acquisitions	16
1.9.1 Solved problems.....	16
1.10 Adlib Serials	16
1.10.1 Solved problems.....	16
1.11 General	17
1.11.1 Solved problems.....	17

1 Improvements in 3.4 applications

These release notes only describe the changes from 3.3 to 3.4.

1.1 Unicode UTF-8

All Adlib files (databases, screens, application structures, help texts, system texts, adapls, etc.) in the 3.4 model applications are now coded in Unicode UTF-8 (previously ANSI). This allows you to include exotic characters and texts in non-western languages in these files, like the titles of screens, in messages that Adlib shows, and for example in the labels of screen fields.

The database option *Locale* is English by default, but can easily be set to your own language. (This setting is relevant for the order in which records are sorted. See the Designer Help for more information about this option.)

1.2 French and Greek

Besides Dutch, English and German, the user interface of Adlib 3.4 applications can be switched to French and Greek as well.

1.3 Reorganization of folders and files

1. The `\bin` subfolder will be called *lexexecutables*, from now on.
2. All `\adapl` source files have been collected in the `\adapl sources` subfolder.
3. From 3.4, all screens are located in the `\screens` subfolder.
4. Titles (menu texts) of output formats can be provided in the `\adapl` itself or in the application structure (`.pbk` file). All those texts have been moved to the application structures of the model applications 3.4, so that they can be read in Adlib Designer and edited in there as well, if necessary.
5. In older application versions, some English text files had a zero behind their name (to indicate that the file was in the English language), while others had no number at all behind their name (since English is the default language). In 3.4, all English help and text files no longer have a zero behind their name; it is now more clear that the English files are the default ones.

6. The texts from the Museum Basis help files have been included in the general help files (which apply to multiple applications) in the `\texts` subfolder.

Subfolders and files which became obsolete as a consequence of the above reorganizations, have been removed.

1.4 Adlib Museum

1.4.1 Improvements in files

1. In the *Documentation* data source and the underlying data sources, the *Sort year* field (tag *sj*), which was already being used to be able to sort document records on the year of publication, is now also used to search on the year of publication. Therefore the name of the field has been changed from *Sort year* into *Search year*. The field is automatically filled with a year, if in the free text fields *Year* (tag *ju*), or *Year (source)* (tag *sy*) a year appears. The *Year of publication* access point in the relevant applications, now searches the index on the tag *sj* of the *Search year* field, instead of the word index on the tag *ju* (which was free-text searching).
2. In the *Persons and institutions* and *Thesaurus* data sources in 3.4 applications, the list of domains which you can assign to a term or name, are displayed alphabetically sorted.
3. In the object collection data sources of Museum applications, the access points *Content period* and *Associated period* have been added. These search the already existing fields *content.date.period* (tag *ib*) and *association.period* (tag *kb*).

1.4.2 Improvements in terminology

1. In the English field labels, the terms “article” and “lead word” were both used. Now, “lead word” is used consistently.
2. In the English field labels, the terms “creator” and “maker” were both used. Now, “creator” is used consistently.
3. The German term “Objectkategorie” has been replaced by “Sachgruppe”.
4. The English field label “Location fitness’ has been replaced by “Location suitability”.

1.4.3 Solved problems

1. All ISBD presentations (generated via `adapls`) have been improved: multiple publishers for one publication are now separated by a colon.

2. In the application structure file, the *Despatch* data source only pointed to the *TRANSPOR* database but not also to the *DESPATCH* dataset in it. So you could open and search the data source, but it was impossible to create new records in it.
3. When you searched the *Name (person keyword)* field (tag *PT*) in the *Documentation* data source, you would never find equivalent terms, because for this field in the data dictionary no *Equivalent field* had been set.
4. In the *THESAU* database (*Thesaurus* data source), the never used *category* field (tag *dr*) has been removed.

1.5 Adlib Museum Basis

1.5.1 Improvements in files

1. In the *Documentation* data source and the underlying data sources, the *Sort year* field (tag *sj*), which was already being used to be able to sort document records on the year of publication, is now also used to search on the year of publication. Therefore the name of the field has been changed from *Sort year* into *Search year*. The field is automatically filled with a year, if in the free text fields *Year* (tag *ju*), or *Year (source)* (tag *sy*) a year appears. The *Year of publication* access point in the relevant applications, now searches the index on the tag *sj* of the *Search year* field, instead of the word index on the tag *ju* (which was free-text searching).
2. In the *Persons and institutions* and *Thesaurus* data sources in 3.4 applications, the list of domains which you can assign to a term or name, are displayed alphabetically sorted.
3. In the object collection data sources of Museum applications, the access points *Content period* and *Associated period* have been added. These search the already existing fields *content.date.period* (tag *ib*) and *association.period* (tag *kb*).

1.5.2 Improvements in terminology

1. In the English field labels, the terms “article” and “lead word” were both used. Now, “lead word” is used consistently.
2. In the English field labels, the terms “creator” and “maker” were both used. Now, “creator” is used consistently.
3. The German term “Objectkategorie” has been replaced by “Sachgruppe”.

1.5.3 Solved problems

1. When you searched the *Name (person keyword)* field (tag *PT*) in the *Documentation* data source, you would never find equivalent terms, because for this field in the data dictionary no *Equivalent field* had been set.
2. In the *THESAU* database (*Thesaurus* data source), the never used *category* field (tag *dr*) has been removed.
3. On the detail screen *acqui_bs.fmt (Field coll. | Acquisition | Value | Location)*, the tag *VJ* has been replaced by *V3* (*V3* already existed in the database). *VJ* was the wrong tag to use here.
4. In the Museum Basis help file there were (duplicate) help labels commented out. When Adlib encountered such a help label, the rest of file was not searched any further for other occurrences of this label, so any existing help text was then ignored and not shown in the application.
The Museum Basis help texts have been reviewed and included in the general help files, which solves the problem.

1.6 Adlib Library

1.6.1 Improvements in files

1. In the *Full catalogue* data source and the underlying data sources *Books*, etc., the *Sort year* field (tag *sj*), which was already being used to be able to sort document records on the year of publication, is now also used to search on the year of publication. Therefore the name of the field has been changed from *Sort year* into *Search year*. The field is automatically filled with a year, if in the free text fields *Year* (tag *ju*), or *Year (source)* (tag *sy*) a year appears. The *Year of publication* access point in the relevant applications, now searches the index on the tag *sj* of the *Search year* field, instead of the word index on the tag *ju* (which was free-text searching).
2. In the *Persons and institutions* and *Thesaurus* data sources in 3.4 applications, the list of domains which you can assign to a term or name, are displayed alphabetically sorted.

1.6.2 Improvements in terminology

1. In the English field labels, the terms “article” and “lead word” were both used. Now, “lead word” is used consistently.

1.6.3 Solved problems

1. All ISBD presentations (generated via adapls) have been improved: multiple publishers for one publication are now separated by a colon.
2. When you searched the *Name (person keyword)* field (tag *PT*) in the *Full catalogue* data source or its underlying data sources, you would never find equivalent terms, because for this field in the data dictionary no *Equivalent field* had been set.
3. In the *THESAU* database (*Thesaurus* data source), the never used *category* field (tag *dr*) has been removed.
4. In the *Full catalogue* and *Serials* data sources in the Library OPAC, for the *ISSN* access point the wrong index was used. And in the *Articles* data source, the *ISSN* access point has been removed since articles don't need an ISSN.
5. On the general search form *qbfdoc.fmt*, the tag *JU* has been corrected into *fg*, and *ju* into *JU*.
6. In the *Extended format* output format (used in the *Full catalogue* and underlying data sources), with serials the holding is now printed as well.

1.7 Adlib Archive

1.7.1 Improvements in files

1. In the *Documentation/Full catalogue* data source and the underlying data sources *Books*, etc., the *Sort year* field (tag *sj*), which was already being used to be able to sort document records on the year of publication, is now also used to search on the year of publication. Therefore the name of the field has been changed from *Sort year* into *Search year*. The field is automatically filled with a year, if in the free text fields *Year* (tag *ju*), or *Year (source)* (tag *sy*) a year appears. The *Year of publication* access point in the relevant applications, now searches the index on the tag *sj* of the *Search year* field, instead of the word index on the tag *ju* (which was free-text searching).
2. In the *Persons and institutions* and *Thesaurus* data sources in 3.4 applications, the list of domains which you can assign to a term or name, are displayed alphabetically sorted.

1.7.2 Improvements in terminology

1. In the English field labels, the terms “article” and “lead word” were both used. Now, “lead word” is used consistently.

1.7.3 Solved problems

1. All ISBD presentations (generated via adapls) have been improved: multiple publishers for one publication are now separated by a colon.
2. A wrong dataset name was used in the field *adapl collect.ada*. Due to this error, the screen field *Closed until* in an Archive application was not automatically calculated after you had filled in *Date (late)* and *Period closed in years*.
3. In the application structure file, the *Despatch* data source only pointed to the *TRANSPOR* database but not also to the *DESPATCH* dataset in it. So you could open and search the data source, but it was impossible to create new records in it.
4. When you searched the *Name (person keyword)* field (tag *PT*) in the *Documentation* data source or its underlying data sources, you would never find equivalent terms, because for this field in the data dictionary no *Equivalent field* had been set.
5. In the *THESAU* database (*Thesaurus* data source), the never used *category* field (tag *dr*) has been removed.
6. In detail screens of the two Archive datasets in all relevant applications equipped with these, for the *Part* screen field in the *Dimensions* field group, a wrong tag was used.

1.8 Adlib Loans

1.8.1 The improved remind procedure

■ New files

In your new Adlib Loans Management model application 3.4 you'll find the following new files for the remind procedure: *adlib#.txt*, *reminder.dot*, *reminder_email.dot*, *reminder_nl.dot*, *reminder_email_nl.dot*, *reminder_de.dot*, *reminder_email_de.dot*, *reminder_fr.dot*, *reminder_email_fr.dot*, *reminder_gr.dot*, *reminder_email_gr.dot*, *reminder.ada*, and *reminder.bin*.

- **adlib#.txt**: extended system texts for all adapls together, English (), Dutch (1), French (2), German (3), Arabic (4) and Greek (6); these text files are located in the *\texts* subfolder in your Adlib Software

folder and contain all texts (per file in one language) that are used by all adapls.

Open each of the new *.txt* files (or just the languages which you use) in Windows Notepad and apply the following changes:

1. Behind line number 192, type the name of the SMTP server which should be used for sending e-mail.
2. Behind line number 278, type the sender e-mail address, for example the general e-mail address of your library.
3. When sending reminders from the file menu in Adlib (in the first step of the *Search wizard*), Adlib possibly generates a *reminder.log* file in the *\wadcirc* or *\library loans management* folder, in which any error message are saved (then the error messages won't be displayed on screen). If you don't want to save such a file, but wish to send it by e-mail to e.g. the application manager, then behind line number 1051 enter the e-mail address of that person or department. If you do not provide an e-mail address, then the file will be stored in the current application folder by default (*\wadcirc* or *\library loans management*), unless you have entered a different path and file name behind line number 1055; then the latter is considered the preferred location.

Save each file after adjusting it.

- **reminder.dot (in five languages)**: examples of a Word template for printing with the new remind procedure.
- **reminder_email.dot (in five languages)**: examples of a Word template for sending e-mail with the new remind procedure.

These files are located in the *Worddoc\templates*-subfolder in your Adlib Software folder.

In your previous application you could probably only print or send reminders with the plain texts that you store in the *Letters* file, and to which you have linked per borrower (if the concerning borrower should be able to receive reminders). By default you can link to three subsequent reminder letters this way.

The new remind procedure makes it possible to print/send reminders via Word templates, in stead of as plain text. The result is now nicely laid out.

You now have to choose how you want to print or send reminders.

- If you are satisfied with the plain texts which you are already using, then you don't have to do anything with the reminder templates: you can throw them away or store them somewhere.

- If, for the processing of reminders per e-mail, you wish to use the new template, then only adjust the fixed texts in the *reminder_email.dot* file to your own situation (you can do that in Microsoft Word), and save the file again with the *.dot* extension (not as *.doc*). In this file, use fixed texts in the language in which you want send reminders via templates: so, type Dutch fixed texts in *reminder_email.dot* (by replacing the English ones), or rename the *reminder_email.dot* file into *reminder_email_en.dot* and *reminder_email_nl.dot* into *reminder_email.dot* so that you can edit the Dutch template now used by Adlib directly, if you would like to send reminders in the Dutch language. (This means that in 3.4 applications it is not possible to send reminders in different languages, e.g. depending on the borrower's own language.) You have to use this *reminder_email* template as the basis for three reminder templates in one language of choice. These templates must be named: *reminder_email1.dot*, *reminder_email2.dot*, and *reminder_email3.dot*. The first file will be used for the first reminder, the second for the second, etc. Put these three templates in the folder that holds your Loans Management application file: *\wadcirc* or *\library loans management*, for instance. Adlib can only find the templates if they are in this folder. Note that *reminder_email_nl.dot* is not being used by the procedure, it just serves as an example.

- If, for the printing of reminders (to the printer), you wish to use the new template, then only adjust the fixed texts in the *reminder.dot* file to your own situation (you can do that in Microsoft Word), and save the files again with the *.dot* extension (not as *.doc*). In this file, use fixed texts in the language in which you want send reminders via templates: so, type Dutch fixed texts in *reminder.dot* (by replacing the English ones), or rename the *reminder.dot* file into *reminder_en.dot* and *reminder_nl.dot* into *reminder.dot* so that you can edit the Dutch template now used by Adlib directly, if you would like to send reminders in the Dutch language. (This means that in 3.4 applications it is not possible to send reminders in different languages, e.g. depending on the borrower's own language.) You have to use this *reminder* template as the basis for three reminder templates in one language of choice. These templates must be named: *reminder1.dot*, *reminder2.dot*, and *reminder3.dot*. The first file will be used for the first reminder, the second for the second, etc. Put these three templates in the folder that holds your Loans Management application file: *\wadcirc* or *\library loans management*, for instance. Adlib can only find the templates if they are in this folder.

Note that *reminder_nl* is not being used by the procedure, it just serves as an example.

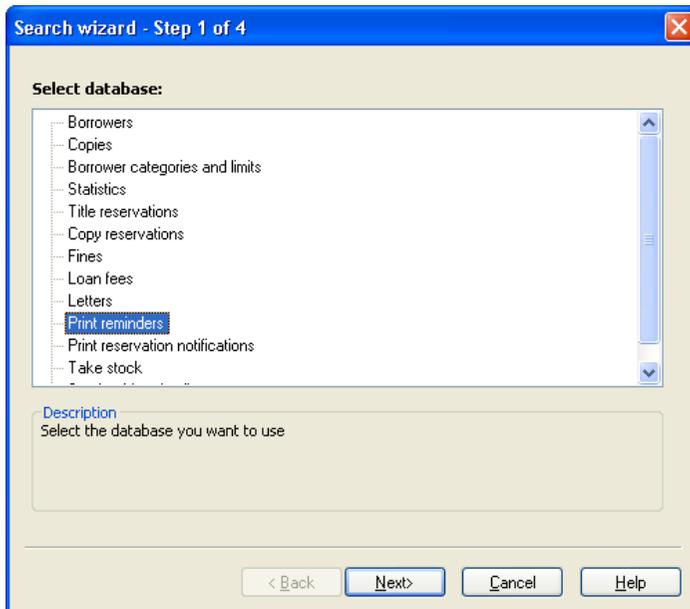
If you have created new templates and put them in the right folder, then in 3.4 applications you don't have to make any links to these templates. When printing or sending reminders, Adlib (do use the most recent update of the Adlib executables) always first checks whether there might be an applicable template present. If it isn't, Adlib automatically uses the plain text letters which you had set already per borrower; then too you profit from the new remind procedure, because more information will be generated (outstanding fines are added) and overdue copies are now sorted by reminder letter type as well.

- **reminder.ada**: the source code of the new remind adapl.
- **reminder.bin**: the compiled adapl;

For these files you don't have to make any settings.

■ Printing reminders

Now you are ready to send reminders. You can do this in two/three ways.



1. In the first step of the *Search wizard* you'll find the *Print reminders* option (see the screenshot above). This is not a database, but a link to an adapl program which checks for all copies on loan whether they are overdue, and subsequently automatically prints or sends a reminder. (Processing a Word template takes a couple of seconds per overdue copy.) Double-click *Print reminders* (and in the following confirmation window click *Yes*) to start the procedure.

2. It's possible as well to print reminders for a selection of borrowers or copies only. Search the desired records in the *Borrowers* or *Copies* database and mark them in the *Brief display* of the search result (see the Adlib User Guide for information about standard functionality). Then you must sort the search result in the *Copies* database on borrower number: this way, copies of a borrower for which the same reminder letter must be made are placed on one letter. Click the *Print* button to open the *Print wizard*. In it, choose *Create a report with a predefined output format*. In the next step of the *Print wizard* you'll find a format for printing reminders, in the list with available output formats. Select it and click *Finish*. In the Windows *Print* window which opens, leave the *Print to file*, *Send by e-mail* and *MS-DOS text* options **unmarked**. Click *OK*, and Adlib automatically determines which reminders must be sent by e-mail and which must be printed.

■ Remarks

- In the *Copies* database, you can find out per copy how many reminders (*Recalls*) already have been sent, and when the last one was sent.

Copy details	Acquisition	Location and counters	Serial details	Management details
DGM 348598/2		Antarctic environmental protocol and its domestic legal implementation		
Locations				
Home site	Den Haag			
Current site	Den Haag			
Location notes				
Counters				
Reservations	Accumulated loans			
Renewals	Recalls		2	
	Last recall sent on		30/01/2006	

- Reminders will always be sent in the correct order: a borrower can never receive *reminder2* before having received *reminder1*, nor *reminder3* before having received *reminder2*. So if you start printing/sending reminders for the first time after a holiday of three weeks, and a borrower with an overdue book has not yet received a reminder, then he or she now receives *reminder1* (even though the book is so much days overdue that normally already a second or third reminder would have been sent). On the other hand, if the library did send *reminder1* prior to the holiday, then now *reminder2* will be printed/sent, etc. Moreover, the difference between the set *Days overdue* for two subsequent letters will still apply for sending the next reminder. So, if the reminder terms are e.g. 7, 14 and 19, and after a holiday of three weeks you start printing reminders again on Monday, and repeat it on Wednesday, then a first reminder from Monday won't be followed up by a reminder two after these two days. Only after 7 (14-7) days you will be able to print reminder number two.
- If some time earlier a reminder was made for an overdue copy, but it isn't known when that was done, then the same reminder will be sent again and the current date will be saved in the relevant copy record.
- Reminders are put together per borrower and per reminder term. This means that all copies of a borrower for which reminder 1 must be made, will be printed on one letter. And all copies of that borrower for which reminder 2 must be made, will be printed together on a subsequent letter. This way, the character of the letter text is always applicable to all copies in that letter.
- If a borrower has received the maximum number of reminders for a copy already, then no further reminders will be created, yet a warning appears on screen or that message will be registered in the log file.
- If a borrower has outstanding fines for copies which were returned too late in the past, then on every reminder letter also an overview of those fines will be printed.
For copies which are currently overdue, no provisional fines will be calculated. Of course you can include a fixed text in your reminder letters, to remind the borrower about the fine per day or week which will be charged for overdue copies.
- Reminders will be sent by e-mail automatically if an e-mail address has been registered for the borrower. If no e-mail address is known, or if e-mailing fails, then the reminder will be printed on paper.

1.8.2 Improvements in files

1. For the reminder procedure of the Library Loans module, the adapls *borr_rem.ada* (borrower database), *copyrem.ada* (copies database), *remind.ada* (adapl command in database menu of application) and *remind.inc* have together been replaced by the new *reminder.ada*, as are the compiled versions (*.bin*) of these files of course. Now, in reminders, information will be added about outstanding fines relating to earlier returned copies. Also, overdue copies will not only be sorted per borrower but per reminder term as well (so a borrower may receive more than one reminder letter), and you can use Word templates for the reminder data. This has altered the operation of the reminder procedure, as described above.

1.9 Adlib Acquisitions

1.9.1 Solved problems

1. Because of a missing subroutine in the *budget.ada* adapl, cost centres weren't updated correctly: the expenditures remained zero after a payment.
2. Records in the *Payment history* data source (*COSTS* database) are now correctly filled during their automatic creation: previously, sometimes empty records were created, due to an error in the *ordcalc* adapl.
3. The contents of the tag *BA* (the commitments) will no longer be shown on the brief display of the *Orders* data source (by the *br_order* before-screen adapl), because users found that information to be unnecessary and confusing.

1.10 Adlib Serials

1.10.1 Solved problems

1. Because of a missing subroutine in the *budget.ada* adapl, cost centres weren't updated correctly: the expenditures remained zero after a payment.
2. For the *Distribution lists* data source, the output format *Print of readers lists* has been improved significantly.
3. Taking in a serial while no readers list is linked to that serial, previously still initialized the printer. (If such a list is linked indeed, then it will be printed automatically for the subsequent reader(s).) In

3.4 however, the printer will only be initialized if something can actually be printed.

Also, a problem has been solved in calculating the date for the next issue.

4. In readers lists, the linked title was displayed without any lead word. In 3.4 the lead word is merged again, as intended.
5. For the data dictionary field *borrower name* (tag *s1*) in the *Distribution lists* data source (*CIRCUIT* database), an erroneous merged-field definition has been corrected: previously, the *titles* field was filled with the fax number of the relevant reader. The error was visible on readers lists, on which the fax number of the reader was printed instead of the titles of the person.

1.11 General

1.11.1 Solved problems

1. To a number of output formats (print adapls) some English texts have been added, where those were missing before.
2. Some missing Dutch and German application texts, like field labels, have been added in 3.4.
3. There were some data dictionary field names with spaces in them, while that is not allowed; those spaces have now been removed.
4. In the general English help file, some texts were still missing. Translations of the Dutch texts have now been added.
5. Where applicable, the spelling of the text "email" has been corrected, and now reads "e-mail".