



How to set up “Replication Enhancements”

Version 18 and higher

Confidential Information

The information herein is the property of Ex Libris Ltd. or its affiliates and any misuse or abuse will result in economic loss. **DO NOT COPY UNLESS YOU HAVE BEEN GIVEN SPECIFIC WRITTEN AUTHORIZATION FROM EX LIBRIS LTD.**

This document is provided for limited and restricted purposes in accordance with a binding contract with Ex Libris Ltd. or an affiliate. The information herein includes trade secrets and is confidential.

Disclaimer

The information in this document will be subject to periodic change and updating. Please confirm that you have the most current documentation. There are no warranties of any kind, express or implied, provided in this documentation, other than those expressly agreed upon in the applicable Ex Libris contract.

Any references in this document to non-Ex Libris Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this Ex Libris product and Ex Libris has no liability for materials on those Web sites.

Copyright Ex Libris Limited, 2010. All rights reserved.
Documentation Originally Released July 2009
Document Last Updated August 09, 2010
Document version 1.0.

Table of Contents

1	INTRODUCTION	5
2	REPLICATION OF NEW RECORDS	5
2.1	Functionality - Replication of new records.....	5
2.1.1	Central system.....	5
2.1.2	Local system	6
2.2	Setup - Replication of new records.....	6
2.2.1	Central system.....	6
2.2.2	Local system	6
2.3	Technical process - Replication of new records	7
2.3.1	Dataflow.....	7
2.3.2	ue_11 log file	8
3	ZDB DATA REPLICATION.....	8
3.1	Functionality - ZDB data replication	9
3.1.1	Central System	9
3.1.2	Local system	9
3.2	Setup - ZDB data replication	10
3.2.1	Central system.....	10
3.2.2	Local system	10
3.3	Technical process - ZDB data replication.....	12
3.3.1	Dataflow.....	12
3.3.2	ue_11 log file	13
4	REDIRECTION OF CENTRAL BIBLIOGRAPHIC RECORD	13
4.1	Functionality - Redirection of central bibliographic record	13
4.1.1	Central system.....	13
4.1.2	Local system	14
4.2	Setup - Redirection of central bibliographic record.....	16
4.2.1	Central system.....	16
4.2.2	Local system	18
4.3	Technical process- Redirection of central bibliographic record	19
4.3.1	Dataflow.....	19
4.3.2	ue_11 log file	19
5	AUTOMATIC REPLICATION OF PARENT RECORDS	20

5.1	Functionality - Automatic replication of parent records	20
5.2	Setup - Automatic replication of parent records	21
5.2.1	Central System	21
5.2.2	Local System.....	21
5.3	Technical process - Automatic replication of parent records	22
5.3.1	Dataflow.....	22
5.3.2	ue_11 log file	22

1 Introduction

The replication enhancements are part of the Aleph cluster functionality. They take effect on the data replication between an Aleph central system and the attached Aleph local systems via Aleph Cluster.

The following enhancements have been implemented:

- Replication of new records
- ZDB data replication
- Redirection of Central Bibliographic Record
- Automatic Transfer of Parent Records

2 Replication of new records

Certain bibliographic records are replicated automatically into those local systems which are interested to present those records locally even so they do not yet exist in the local system. This functionality is intended for bibliographic records which are not linked to holdings, like free electronic journals or bibliographic records for free accessible electronic materials.

2.1 Functionality - Replication of new records

This functionality provides distribution of special bibliographic records located in a central catalogue to all local Aleph systems which are interested in storing this record locally. The distribution will be operated automatically via Aleph cluster when an update of the record in the central system takes place.

2.1.1 Central system

The flag that indicates which BIB record should be automatically replicated can either be:

- existence of a certain tag
- existence of a certain tag + certain indicators
- existence of a certain sub field of a certain tag (+ certain indicator)
- existence of a certain string in a sub field of a certain tag (+ certain indicator)

It is possible to define different groups of records which should be automatically replicated. Those groups are necessary to distinguish between different types of data like free ejournals, scanned materials, materials which are only accessible to certain libraries.

An indicator could be for example field 088 with a specific string at subfield \$a:

088__ a free-01

If a record with this indicator is added, updated or deleted in the central system it will be automatically replicated.

2.1.2 Local system

It is possible to configure which local system(s) should receive the automatically replicated records. A local system can choose one or multiple groups in order to receive BIB records assigned to these groups. Those records will be created, updated or deleted automatically. A record will be created if it does not already exist, regardless whether it was added or updated in the central system.

2.2 Setup - Replication of new records

2.2.1 Central system

tab replication group

Those records which should be automatically replicated needs a flag which assign the record to a replication group. This flag can be a string or substring in a certain sub field or just existence of a certain tag. The different flags and the assigned replication groups need to be defined in tab_replication_group.

Example: tab_replication_group
(in data_tab of BIB library)

```
!  
! COL 1. 5; NUM; ;  
! Group;  
! Replication group identifier;  
! !  
! COL 2. 5; ALPHA_NUM, UPPER ; ;  
! Tag name;  
! Tag + indicator;  
!  
! COL 3. 1; ALPHA_NUM, ; ;  
! Subfield;  
! Subfield for match;  
!  
! COL 4. 40; ALPHA_NUM, ; ;  
! String;  
! String or substring which indicates replication of new record;  
!  
!1 2 3 4  
!!!!-!!!!-!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!  
free 087## a free-01  
free 087## a free-02  
scan 088## z
```

The group flag in column 1 will be added to the z105 message of type 1 in case the record will be added, updated or deleted.

2.2.2 Local system

tab_ue11.conf

In the local system those group flags which should cause creation and deletion of records need to be defined in tab_ue11.conf.

Example: tab_ue11.conf
(in data_tab of z105_library)

```
[A]
  FIX-NEWDOC = INS2

[A-1]
  GROUP-LIST = free, scan
  CHECK-TYPE = CATALOG-DELETE
```

FIX-NEWDOC

For the creation of records a Fix routine can be defined. The Fix routine must exist in tab_fix of the BIB library.

GROUP-LIST

In case the z105 message contains a listed group flag the record will be created, updated or deleted. In case the z105 message contains a different group flag the record will not be created or deleted, but updated if it exists already in the local catalogue.

If there is more than one BIB library in the local system it might be necessary to define which group is relevant for which library. E.g. if there are the two BIB libraries XXX01 and YYY01 in a local system the parameter might be defined like this:

```
GROUP-LIST = XXX01:free, YYY01:scan
-> records of group free are relevant for XXX01
   records of group scan are relevant for YYY01

GROUP-LIST = XXX01:free, scan
-> records of group free and scan are relevant for XXX01
   no group is defined for Library YYY01

GROUP-LIST = free, scan
-> records of group free and scan are relevant for XXX01 and YYY01
```

CHECK-TYPE

For the deletion of records validation checks can be defined. The check type defined here must exist in table check_dock of the BIB library.

2.3 Technical process - Replication of new records

2.3.1 Dataflow

Update (new, correction) of a BIB record creates a z105 messages for every target bib library. Based on tab_replication_group the message may contain a group flag. Deletion will only cause z105 message in case the record is flagged according to tab_replication_group. Ue_11 transfers the messages to the target libraries.

Local ue_11 takes every message and checks whether it contains a group flag. If the message does not contain a group flag processing of the messages remains as already known. If the message contains a group flag it will be checked against the relevant entry in tab_ue11.conf. If no corresponding entry in tab_ue_11.conf could be found (or tab_ue11.conf does not exist) further processing of the message remains unchanged.

Only in case the group flag is listed in tab_ue11.conf the system will behave differently. If the record already exists in the local system the corresponding record from the central catalogue will update the local record version. If no corresponding version exists centrally the local record will be deleted. As part of the deletion validation checks will be performed according to CHECK-TYPE in tab_ue_11.conf. If deletion failed a message to the ue_11 log file will be reported. Those messages can be extracted via grep command in order to perform the delete manually.

If ue_11 receives a corresponding record from the central catalogue and no former version exists locally ue_11 will create the new record.

2.3.2 ue_11 log file

All messages send during this replication process are logged in the file run_e_11.nnnnn in the data_scratch directory of the z105_library.

In the following example the central library is CEN01 and the local library is LOC02.

Central ue_11 log file:

```
2008-08-08 15:53:20 Update : 1 00 LOC02 CEN01:000000025 FREE $$aBC-00000026
```

Local ue_11 log file:

```
2008-08-08 15:53:28 Update : 1 00 LOC02 CEN01:000000025 FREE $$aBC-00000026
```

```
2008-08-08 15:53:29 New record '000000006' was added to library 'LOC02'
```

In case the deletion of a BIB record failed appropriate messages will be logged, e.g.:

```
2008-08-08 16:51:23 [wrn] Check doc 'LOC02/000000101' by check-type=CATALOG-DELETE with messages:
```

```
2008-08-08 16:51:23 + W - Administrative record 000000101 in library LOC52 points to current document with link type ADM.
```

```
2008-08-08 16:51:23 + M - Document has 1 item(s) attached to ADM record 000000101 in library LOC52
```

```
2008-08-08 16:51:23 [err] Record '000000101/LOC02' was not deleted, check-doc with 'M'-message(s)
```

3 ZDB data replication

Serial data from German libraries are reported to the German National Library by the local libraries to maintain the central serial database (ZDB). Updates of this database are provided to central catalogues via batch download or OAI. In order to update the local Aleph catalogues the “ZDB data replication” allows the online update via Aleph Cluster replication.

3.1 Functionality - ZDB data replication

3.1.1 Central System

Updates (new, correction) and deletions of ZDB HOL records in the central catalogue trigger the replication of ZDB data based on `tab_z105` and `tab_z105_filter`. A `z105` message with message type (f) will be sent to the local systems.

3.1.2 Local system

The local system will distinguish between new HOL records, updates and deletions.

- If no HOL record with the provided ID could be located in the local HOL library the record will be created according to the standard `ue_11` procedure.
- If a local HOL record could be located the system will overwrite this record with the central HOL information. Defined merge algorithm will be taken into account.
- If a local HOL record could be located but the central version does not exist `ue_11` will consider this as a deletion of the local HOL record.

It may be possible that the BIB record linked to the HOL record is not part of the local bib library (hol-bib relation according to `library_relation`). In this case `ue_11` will fetch this BIB record from the central catalogue and will create a new BIB record. This mechanism is as well used in case the content of the tag 012 (LKR) has been changed and the HOL record needs to be linked to a different BIB record which may not be part of the local bib.

When a HOL record will be deleted in the local system `ue_11` will try to delete item records that are linked via `Z30-HOL-DOC-NUMBER` with the HOL record. Standard check routines will be performed before the deletion of an item record and error messages logged if the deletion is not possible.

In addition `ue_11` will try to delete the linked BIB record of the HOL record that will be deleted. The `ue_11` will also try to delete the “old” BIB record if the HOL record has been linked to a different BIB record. Users have to make sure via a proper definition of check doc routines that the BIB record will only be deleted if no other relevant links exist.

Please note that a BIB record can't be deleted if there are remaining items linked. Items will be deleted only via the `Z30-HOL-DOC-NUMBER` link before the deletion of the bib record.

3.2 Setup - ZDB data replication

3.2.1 Central system

tab_z105

For the replication of HOL records from central to local systems an entry with message type (f) needs to be defined at tab_z105.

Example: tab_z105

(in data_tab of HOL library)

```
!  
! 1          2 3      4      5      6      7      8      9      10 ...  
!!!!!!!!!!!!!!-!-!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!-...  
UPDATE-DOC      f XXX60 YYY60 ZZZ60
```

Column 1: Action which triggers a message: UPDATE-DOC = Update of records (Z00)

Column 2: Message type: f = update of local ZDB HOL record from central ZDB HOL record

Column 3 – 12: Library: Code of library to which message is sent

tab_z105_filter

As only update of ZDB HOL records should trigger those new messages tab_z105_filter needs to be adjusted accordingly.

Example: tab_z105_filter

(in data_tab of HOL library)

```
!  
!1 2 3  
!-!-!!!!!!  
f Y 025z#
```

Column 1: Message type: f = update of local ZDB HOL record from central ZDB HOL

Column 2: Action: Y - Retains field, N - Does not retain field

Column 3: Tag code

The message will only be sent if 025z exists in the central HOL record.

Note: The Boolean operator OR is used to combine several entries of tab_z105_filter!

3.2.2 Local system

tab_ue11.conf

For the deletion of BIB records a check routine should be defined in tab_ue11.conf.

Example: tab_ue11.conf

(in data_tab of z105_library)

[A]

FIX-NEWDOC = INS2

[A-F]

CHECK-TYPE = CATALOG-DELETE

FIX-NEWHOL = ZDBN

FIX-UPDHOL = ZDBO

CHECK-LOW-LIB-RELATION = N

CHECK-LOW-TAB-OWN = N

FIX-NEWDOC

For the creation of BIB records a Fix routine can be defined. The Fix routine must exist in `tab_fix` of the BIB library.

CHECK-TYPE

For the deletion of records validation checks can be defined. The check type defined here must exist in table `check_dock` of the BIB library.

FIX-NEWHOL

For the creation of HOL records a Fix routine can be defined. The Fix routine must exist in `tab_fix` of the HOL library.

FIX-UPDHOL

For the update of HOL records a Fix routine can be defined. The Fix routine must exist in `tab_fix` of the HOL library.

CHECK-LOW-LIB-RELATION

By default the system only checks if the owner (=sub library) of the HOL record exists in `tab_sub_library.lng` to decide if a record is relevant for the local HOL library. If there is more than one HOL library in the local system receiving ZDB records from the central system this is not sufficient. If `CHECK-LOW-LIB-RELATION` is set to Y it will additionally be checked if the ADM library defined in `tab_sub_library.lng` for the owner is related to the HOL library. This must to be done in a multi ADM system e.g. with the following Libraries:

XXX01, XXX50, XXX60 / YYY01, YYY50, YYY60

Parameter `CHECK-LOW-TAB-OWN` must be set to N:

CHECK-LOW-LIB-RELATION = Y

CHECK-LOW-TAB-OWN = N

CHECK-LOW-TAB-OWN

By default the system only checks if the owner (=sub library) of the HOL record exists in `tab_sub_library.lng` to decide if a record is relevant for the local HOL library. If there is more than one HOL library in the local system receiving ZDB records from the central system this is not sufficient. If `CHECK-LOW-TAB-OWN` is set to Y it will additionally be checked if the owner is defined in `tab_own` of the HOL library. This must to be done in a single ADM system e.g. with the following Libraries:

XXX01, XXX60 / YYY01, YYY60 / ZZZ50

Parameter `CHECK-LOW-LIB-RELATION` must be set to N:

CHECK-LOW-LIB-RELATION = N

CHECK-LOW-TAB-OWN = Y

tab_merge

In order to define a merge algorithm for the update of HOL records the special merge routine UE11-F must be added in tab_merge.

Example: tab_merge

(in data_tab of HOL library)

```
!   1                               2                               3
!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
UE11-F      merge_doc_overlay                               10
```

Column 1: Routine name

Column 2: Program name

Column 3: Program arguments (Merge set)

Merge set 10 must be defined at tab_merge_overlay.

3.3 Technical process - ZDB data replication

3.3.1 Dataflow

Update (new, correction) and deletion of a HOL record in the central system creates a z105 message with message type (f) for every target local bib library based on tab_z105 and tab_z105_filter. Ue_11 transfers the messages to the target libraries. The following information will be part of each message:

Source library (central HOL)

Target library (local HOL)

HOL system number in source library

HOL ID

OWN information or in case of MARC sub library code from 852

BIB library

BIB ID

Local ue_11 takes every message and will first check - based on the OWN (sub library) information - if the message is relevant for the local system. For this purpose the OWN (sub library) information will be checked against the list of defined sub library codes according to tab_sub_library.eng. Only sub library codes of type “1” (full sub library) will be considered for this check. If no sub library code similar to the OWN (sub library) information could be found the message will be deleted without further processing.

If the message is relevant for the local system ue_11 will fetch the corresponding HOL record from the central system. If the record exists in the central system ue_11 will update the local version or create a new record if no version exists locally. If the record doesn't exist in the central system ue_11 will delete the local version.

In case the linked BIB record doesn't exist in the central system ue_11 will fetch the record from the central catalogue and create add the record to the local BIB.

3.3.2 ue_11 log file

All messages send during this replication process are logged in the file run_e_11.nnnnn in the data_scratch directory of the z105_library.

In the following example the central library is CEN61 and the local library is LOC62. The owner (OWN) is SL201.

Central ue_11 log file:

```
2008-09-10 16:32:33 Update : f 00 LOC62 CEN61:000000009 CEN01:SL201:$aMH000000005
```

Local ue_11 log file:

```
2008-09-10 16:32:49 Update : f 00 LOC62 CEN61:000000009 CEN01:SL201:$aMH000000005
```

```
2008-09-10 16:32:58 New record '000000024' was added to library 'LOC02'
```

```
2008-09-10 16:33:00 Update : 2 00 CEN61 LOC62:000000005 $aBC-000000039
```

In this example a BIB records has been created in the local system.

The message "...Update : 2 00 CEN61 LOC62:000000005 \$aBC-000000039" can be ignored. For all updates of local HOL records such messages are recorded but shouldn't be sent to the central system for ZDB records, based on tab_z105_filter.

In case the deletion of an item record failed appropriate messages will be logged, e.g.:

```
2008-09-11 12:04:11 + 000000013 : Del record : Item 000000027-000010 has connection(s) - no delete
```

4 Redirection of central bibliographic record

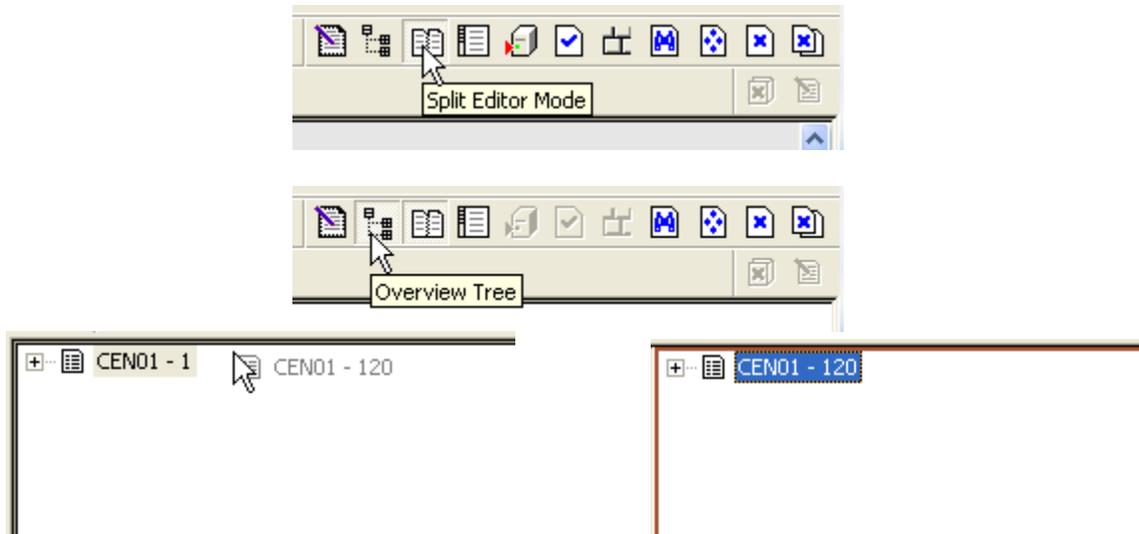
Central bibliographic redirections are required to enable manual data maintenance in the central catalogue. The redirection of a central BIB record will be replicated automatically via Aleph cluster in order to perform this redirection in the local systems as well.

4.1 Functionality - Redirection of central bibliographic record

This functionality consists of a central part which describes the functionality in the central catalogue and a local part which refers to how the messages will be operated in the local system.

4.1.1 Central system

Redirection can be performed in the Aleph GUI based on split editor mode by trekking one BIB record with the mouse on the second record in the overview tree:



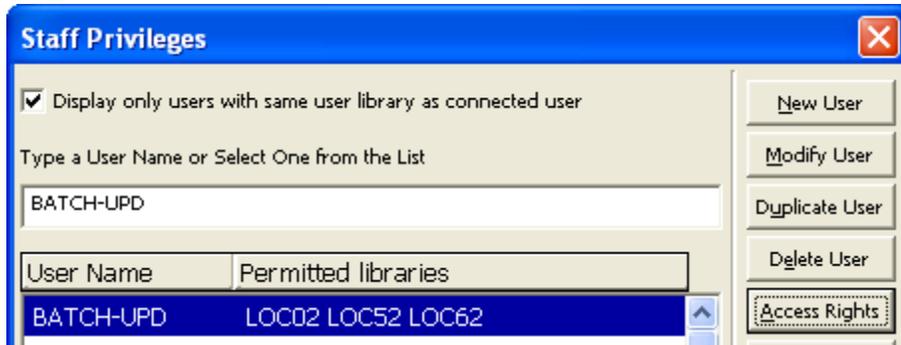
This mechanism is only available if it is activated in `tab_move_record`. The following tasks will be performed:

- Re-link of all HOL records from the source to the target.
- Re-link of all linked BIB record from the source to the target according to `tab_relink_link` (MAB only).
- Move of LOW tags from source to target without creation of duplicate LOW tags.
- Deletion of z403 records assigned to the source and recreation of those z403 records for the target. (Re-link of z403 is not possible.)
- Deletion of all z300 records assigned to the source record
- Only if all tasks listed above are successfully performed and no remaining lower level records are linked to the source record the source will be deleted
- Updates of redirected records trigger z105 update messages with message type (g).

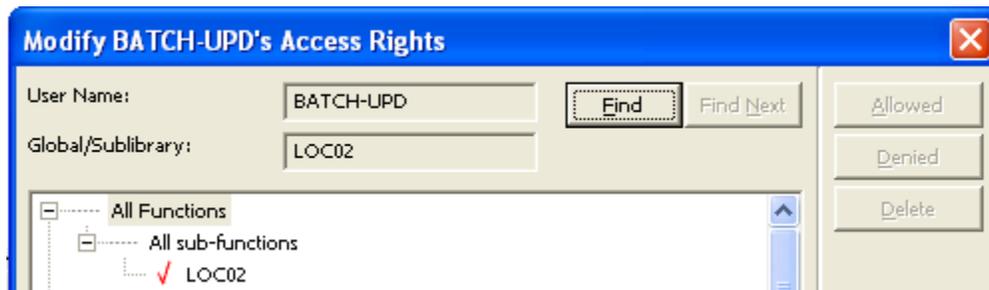
Depending on the amount of linked records it may be possible that not all lower level records could be re-linked to the target record within internal time out. In this case the source record remains in the system and the redirection needs to be manually performed again in order to finalize the redirection. Deletion of z300 will only be performed after all re-linking activities listed above have been finished in order to prevent deletion of z300 if the redirection cannot be finalized within one step.

4.1.2 Local system

In the local System the staff user “BATCH-UPD needs to be created with access rights for the BIB, HOL and ADM library:



All sub-functions should be allowed for all functions:



The local ue_11 will first check if the source record exists in the local system. This is a precondition for further proceeding. If so - ue_11 will check if the target record is already available locally. In case the target is already linked to an ADM record checks according to “move_adm_to_adm” will be performed. In case the check result will prevent the redirection the whole process will be stopped. This will be reported to the ue_11 log file.

If the target record does not exist the ue_11 will fetch the target record from the central system and will create the record locally.

Those are the tasks which will be performed as part of the local redirection:

- ADM environment; tasks will be performed per local ADM library assigned to the BIB according to library_relation
 - If the target is not linked to an ADM record re-link of the linked ADM record from the source to the target
 - If ADM record for the target already exist it will be checked according to “move_adm_to_adm” if full ADM information could be moved to the target BIB
 - If this check results in prevention of the move re-direction activities will be stopped and the issue will be reported to the ue_11 logfile
 - If check result does not prevent the move all ADM data will be move to the target record
 - Re-link of all ITM links from the source ADM record to target ADM record
- Re-link of all HOL records from the source to the target.

- Re-link of all linked BIB record from the source to the target according to tab_relink_link (MAB only).
- Deletion of z403 records assigned to the source and recreation of those z403 records for the target. (Re-link of z403 is not possible.)
- Only if all tasks listed above are successfully performed and no remaining lower level records are linked to the source record the source will be deleted
- Tags of the source record will be transferred to the target record based on merge definition (tab_merge)
- Updates of redirected records trigger z105 update messages (e.g. z30, HOL).

4.2 Setup - Redirection of central bibliographic record

4.2.1 Central system

tab move record

The redirection functionality needs to be activated in tab_move_record.

Example: tab_move_record
(in data_tab of BIB Library)

```
!
! 1          2          3
!!!!!!!!!!!!-!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
BIB          BIB          move_cen_bib_to_bib
```

Column 1: Moving from Record

Column 2: Moving to Record

Column 3: Moving Procedure

tab z105

For the replication of redirections from central to local systems an entry with message type (g) needs to be defined at tab_z105.

Example: tab_z105
(in data_tab of BIB library)

```
!
! 1          2 3 4 5 6 7 8 9 10 ...
!!!!!!!!!!!!-!-!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!-...
RELINK-DOC    g XXX01 YYY01 ZZZ01
```

Column 1: Action which triggers a message: RELINK-DOC = Relink of records (Z00)

Column 2: Message type: g = provision of redirection information to local systems

Column 3 – 12: Library: Code of library to which message is sent

tab relink link

In order to re-link all linked BIB record from the source to the target all link fields needs to be defined in tab_relink_link.

Example: tab_relink_link
(in data_tab of BIB library)

```

!
! COL 1. 5; ALPHA_NUM; #;
!           Field, mandatory;
!           Field code of link field;
! COL 2. 1; ALPHA_NUM; ;
!           Subfield, mandatory;
!           Subfield code with link number;
! COL 3. 3; ALPHA_NUM {SYS, 001}; ;
!           Linked by, mandatory;
!           Type of link number (col.2)
! COL 4. 1; ALPHA_NUM; ;
!           Subfield, optional;
!           Subfield code with link type (i.e. LKR-fields)
! COL 5. 5; ALPHA_NUM; ;
!           Specified link type;
!           Link type text in subfield (col.4)
! COL 6. 1; ALPHA_NUM; ;
!           Subfield, optional;
!           Subfield code with target library;
! COL 7. 3; ALPHA_NUM; ;
!           LKR type, mandatory;
!           Relevant Z103-LKR-TYPE in the Z103-table;
! COL 8. 1; ALPHA_NUM {Y, N}; ;
!           Switch, optional, default=Y;
!           Y: Link field will be deleted if source and target
!               records are identical
! COL 9. 1; ALPHA_NUM {Y, N}; ;
!           Switch, optional, default=N (not Y);
!           Y: Use this field to check the existence of linked
!               record and to create the record (if not found) in
!               ue_11-functions by ue_11_get_linked_records
!           - only relevant for links with 001-field (col.3=001)
!
! 1 2 3 4 5 6 7 8 9
!!!!-!-!!!-!-!!!!-!-!!!-!-!
010## a 001          DN      Y
021## a 001          SEK      Y
022## a 001          PRI      Y
599 # a 001          AND      Y
453## a 001          SRD      Y
463## a 001          SRD      Y
473## a 001          SRD      Y
483## a 001          SRD      Y
493## a 001          SRD      Y
526## 9 001          PAR
527## 9 001          PAR
528## 9 001          PAR
529## 9 001          PAR
530## 9 001          PAR
531## 9 001          PAR
532## 9 001          PAR
533## 9 001          PAR
534## 9 001          PAR

```

```
623## a 001          SRD  Y
629## a 001          SRD  Y
```

4.2.2 Local system

tab ue11.conf

Example: tab_ue11.conf
(in data_tab of z105_library)

```
[A]
    FIX-NEWDOC = INS2
```

FIX-NEWDOC

For the creation of records a Fix routine can be defined. The Fix routine must exist in tab_fix of the BIB library.

tab relink link (BIB)

In order to re-link all linked BIB record from the source to the target all link fields needs to be defined in tab_relink_link.

Example: see “setup central system”

tab merge

In order to define a merge algorithm for the redirection of the local BIB records the special merge routine UE11-G must be added in tab_merge.

Example: tab_merge
(in data_tab of BIB library)

```
!   1                               2                               3
!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
UE11-G      merge_doc_overlay          98
```

Column 1: Routine name
Column 2: Program name
Column 3: Program arguments (Merge set)

Merge set 98 must be defined at tab_merge_overlay. Please note that the merge set of merge routine UE11-1 can't be used because source and target record are inverted.

tab relink link (ADM)

In order to re-link all ITM links if ADM data of the source record have been moved to an already existing ADM record at the target record the LKR field needs to be defined in tab_relink_link.

Example: tab_relink_link
(in data_tab of ADM library)

```
!
! 1  2  3  4  5  6  7  8
```

```
!!!!-!-!!!!-!-!!!!-!-!!!!-!  
LKR## b SYS a ITM 1 ITM
```

For description of columns see “setup central system”

4.3 Technical process- Redirection of central bibliographic record

4.3.1 Dataflow

Redirection of a BIB record in the central system creates a z105 message with message type (g) for every target local bib library based on tab_z105. Ue_11 transfers the messages to the target libraries. The following information will be part of each message:

- Source library (central BIB)
- Target library (local BIB)
- System number of source record in central BIB
- ID of source record in central BIB
- System number of target record in central BIB
- ID of target record in central BIB

Local ue_11 will first check if the source record exists in the local system. If it doesn't exist the message will be deleted without further processing.

If the target record does not exist the ue_11 will fetch the target record from the central system and will create the record locally.

When the ADM data will be moved to the target record the local ue_11 will sent z105 messages for each z30 record. The central ue_11 will fetch the z30 records from the local system in order to create new z300 records for the central target record. This is important because the z300 records have been deleted before by the central redirection process.

During the redirection process a lot of records could be updated that trigger the creation of standard z105 messages. In the central system such messages are created for example for all re-linked bib records. In the local system the re-link of ADM records will trigger such messages. These records will also be re-linked by the replication process so that these messages are not relevant.

4.3.2 ue_11 log file

All messages send during this replication process are logged in the file run_e_11.nnnnn in the data_scratch directory of the z105_library.

In the following example the central library is CEN01 and the local library is LOC02.

Central ue_11 log file:

```
2008-10-27 16:16:27 Update : g 00 LOC02 CEN01:000000107-$$aBC-00000110 TO
```

Local ue_11 log file:

```
2008-10-27 16:16:50 Update : g 00 LOC02 CEN01:000000107-$$aBC-00000110 TO
2008-10-27 16:16:50 [inf] Start relink 'CEN01:000000107->000000112' source-001='$$aBC-00000110' ->
target-001='$$aBC-00000115'
2008-10-27 16:16:50 New record '000000099' was added to library 'LOC02'
2008-10-27 16:16:50 [inf] Start ue_11_move_bib_to_bib for 'LOC02/000000089->LOC02/000000099'
2008-10-27 16:16:51 [wrn] - move_loc_bib_to_bib: Message buffer (error-code=00) of relink
'LOC02/000000089->LOC02/000000099':
2008-10-27 16:16:51 + relink - 'LOC02-000000088' : Update field 010 : $$aBC-00000110 -> $$aBC-
00000115
2008-10-27 16:16:51 + relink - Link DN in record 'LOC02-000000088' changed to 'LOC02-000000099' /
BC-00000115
2008-10-27 16:16:51 + Record no. 89 moved successfully
```

5 Automatic replication of parent records

In a MAB based catalogue records of multi volume monographs and monographs which are part of a series are linked to upper level BIB records. In case of a manual change of the linking information in the child record which causes a change of the link the new father record will be automatically fetched by the local catalogue as part of the Aleph Cluster replication. The links are always checked when a record is updated in the local system via the replication even though the link fields have not been changed. So if a father record is missing for other reasons it will also be fetched automatically.

5.1 Functionality - Automatic replication of parent records

During operation of messages of type 1 (central bib update), type f (ZDB HOL replication) and type g (redirections) BIB records provided by the central catalogue will either update local BIB records or will be used to create a corresponding BIB record locally.

In case the updated record which is taken from the central catalogue by the ue_11 contains link information, ue_11 will compare the link information of the “old” record version stored locally and the new version provided by the central BIB. Which tags should be considered for this check is defined in tab_relink_link. In case ue_11 detects differences concerning the link information it will check if all target records – addressed in the new record - exist in the local catalogue based on the direct index IDN. For new BIB records which did not exist in the local BIB before, ue_11 will try to locate those target records in the local BIB as well. If a target record could not be located in the local catalogue, ue_11 will ask the central PC server to provide full record information of the target to create this record locally before update/creation of the originally provided BIB record will be performed.

If the required BIB record can not be provided by the central catalogue an error message will be reported to ue_11 logfile. This error message does not prevent the update of the local BIB record.

5.2 Setup - Automatic replication of parent records

5.2.1 Central System

This functionality does not require a special setup in the central system.

5.2.2 Local System

tab ue11.conf

Example: tab_ue11.conf
(in data_tab of z105_library)

```
[A]
  FIX-NEWDOC = INS2
```

FIX-NEWDOC

For the creation of records a Fix routine can be defined. The Fix routine must exist in tab_fix of the BIB library.

tab relink link

Via Column 9 of tab_relink_link it can be defined which fields should be used to check the existence of linked record.

Example: tab_relink_link
(in data_tab of BIB library)

```
!
! COL  1.  5;  ALPHA_NUM; #;
!           Field, mandatory;
!           Field code of link field;
! COL  2.  1;  ALPHA_NUM; ;
!           Subfield, mandatory;
!           Subfield code with link number;
! COL  3.  3;  ALPHA_NUM {SYS, 001}; ;
!           Linked by, mandatory;
!           Type of link number (col.2)
! COL  4.  1;  ALPHA_NUM; ;
!           Subfield, optional;
!           Subfield code with link type (i.e. LKR-fields)
! COL  5.  5;  ALPHA_NUM; ;
!           Specified link type;
!           Link type text in subfield (col.4)
! COL  6.  1;  ALPHA_NUM; ;
!           Subfield, optional;
!           Subfield code with target library;
! COL  7.  3;  ALPHA_NUM; ;
!           LKR type, mandatory;
!           Relevant Z103-LKR-TYPE in the Z103-table;
! COL  8.  1;  ALPHA_NUM {Y, N}; ;
!           Switch, optional, default=Y;
!           Y: Link field will be deleted if source and target
```

```

!           records are identical
! COL  9.  1; ALPHA_NUM {Y, N}; ;
!           Switch, optional, default=N (not Y);
!           Y: Use this field to check the existence of linked
!           record and to create the record (if not found) in
!           ue_11-functions by ue_11_get_linked_records
!           - only relevant for links with 001-field (col.3=001)
!
! 1  2  3  4  5  6  7  8  9
!!!!-!-!!!!-!-!!!!-!-!!!!-!-!
010## a 001          DN      Y
021## a 001          SEK      Y
022## a 001          PRI      Y
599 # a 001          AND      Y
453## a 001          SRD      Y
463## a 001          SRD      Y
473## a 001          SRD      Y
483## a 001          SRD      Y
493## a 001          SRD      Y
526## 9 001          PAR
527## 9 001          PAR
528## 9 001          PAR
529## 9 001          PAR
530## 9 001          PAR
531## 9 001          PAR
532## 9 001          PAR
533## 9 001          PAR
534## 9 001          PAR
623## a 001          SRD      Y
629## a 001          SRD      Y

```

5.3 Technical process - Automatic replication of parent records

5.3.1 Dataflow

The check of the existence of linked record will be done by the local ue_11 according to tab_relink_link for new and updated records. If a linked record does not already exist in the local BIB library ue_11 will fetch this record from the central system and will add it to the local BIB. If the record doesn't exist in the central system an error message will be logged.

5.3.2 ue_11 log file

Messages for replication of linked records are logged in the file run_e_11.nnnnn in the data_scratch directory of the z105_library.

In the following example the central library is CEN01 and the local library is LOC02.

Local ue_11 log file:

```

2008-10-27 15:14:42 [inf] Record 'LOC02/000000097' : New linked record
'LOC02/000000098' created from record 'CEN01/000000118'; linked by
field 010 with IDN=$$a BC-00000121

```