



Aleph Cluster Replication Enhancements

Version 20.0

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. This information is provided AS IS. Unless otherwise agreed, Ex Libris shall not be liable for any damages for use of this document, including, without limitation, consequential, punitive, indirect or direct damages.

Any references in this document to third-party material (including third-party Web sites) are provided for convenience only and do not in any manner serve as an endorsement of that third-party material or those Web sites. The third-party materials are not part of the materials for this Ex Libris product and Ex Libris has no liability for such materials.

TRADEMARKS

"Ex Libris," the Ex Libris bridge, Primo, Aleph, Alephino, Voyager, SFX, MetaLib, Verde, DigiTool, Preservation, URM, Voyager, ENCompass, Endeavor eZConnect, WebVoyage, Citation Server, LinkFinder and LinkFinder Plus, and other marks are trademarks or registered trademarks of Ex Libris Ltd. or its affiliates.

The absence of a name or logo in this list does not constitute a waiver of any and all intellectual property rights that Ex Libris Ltd. or its affiliates have established in any of its products, features, or service names or logos.

Trademarks of various third-party products, which may include the following, are referenced in this documentation. Ex Libris does not claim any rights in these trademarks. Use of these marks does not imply endorsement by Ex Libris of these third-party products, or endorsement by these third parties of Ex Libris products.

Oracle is a registered trademark of Oracle Corporation.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Ltd.

Microsoft, the Microsoft logo, MS, MS-DOS, Microsoft PowerPoint, Visual Basic, Visual C++, Win32, Microsoft Windows, the Windows logo, Microsoft Notepad, Microsoft Windows Explorer, Microsoft Internet Explorer, and Windows NT are registered trademarks and ActiveX is a trademark of the Microsoft Corporation in the United States and/or other countries.

Unicode and the Unicode logo are registered trademarks of Unicode, Inc.

Google is a registered trademark of Google, Inc.

Copyright Ex Libris Limited, 2009. All rights reserved.

Document released: May 24, 2009

Web address: <http://www.exlibrisgroup.com>

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.....	7
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.....	11
3.3	Technical Process - ZDB Data Replication	12
3.3.1	Dataflow.....	12
3.3.2	ue_11 Log File	12
4	REDIRECTION OF CENTRAL BIBLIOGRAPHIC RECORD	13
4.1	Functionality	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 the 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 the local systems that want to present these records locally even though they do not yet exist in the local system. This functionality is intended for bibliographic records that 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 that are interested in storing these records locally. The distribution is operated automatically via the Aleph cluster when an update of the records in the central system takes place.

2.1.1 Central System

The flag that indicates which BIB records should be automatically replicated can either be the existence of a specific:

- tag
- tag + indicators
- sub field of a tag (+indicator)
- string in a sub field of a tag (+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, and materials which are only accessible to certain libraries. An indicator could be, for example, field 088 with a specific string at subfield \$a:

```
$a:  
088_a free-01
```

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

2.1.2 Local System

It is possible to configure which local systems 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 are created, updated, or deleted automatically. A record is created if it does not already exist, regardless of whether or not 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 that should be automatically replicated need a flag that assigns the record to a replication group. This flag can be a string or substring in a certain sub field or just the existence of a specific 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 the 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 is added to the z105 message of type 1 in case the record is added, updated, or deleted.

2.2.2 Local System

tab_ue11.conf

In the local system those group flags that create and delete records need to be defined in `tab_ue11.conf`.

Example: `tab_ue11.conf`
(in `data_tab` of the `z105_library`)

```
[A]
    FIX-NEWDOC = INS2

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

FIX-NEWDOC

A Fix routine can be defined for the creation of records. 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 is created, updated, or deleted. If the `z105` message contains a different group flag the record is not created or deleted, but updated if it already exists in the local catalogue.

CHECK-TYPE

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

2.3 Technical Process - Replication of New Records

2.3.1 Dataflow

When a BIB record is updated, a `z105` messages for every target bib library is created. Based on `tab_replication_group`, the message may contain a group flag. If a BIB record is deleted, a `z105` message is only created if the record is flagged according to `tab_replication_group`. `Ue_11` transfers the messages to the target libraries.

Local `ue_11` checks every message for a group flag. If the message does not contain a group flag, processing of the messages remains is done as before. If the message contains a group flag, it is checked against the relevant entry in `tab_ue11.conf`. If no corresponding entry in `tab_ue_11.conf` is found (or if `tab_ue11.conf` does not exist) further processing of the message is done as before.

Only if the group flag is listed in `tab_ue11.conf` does the system behave differently. If the record already exists in the local system, the corresponding record from the central catalogue updates the version of the local record. If no corresponding version exists

centrally, the local record is deleted. During the validation of the deletion, checks are performed according to `CHECK-TYPE` in `tab_ue_11.conf`. If the deletion fails, a message to the `ue_11` log file is reported. These messages can be extracted via the `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` creates 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 is logged, for example:

```
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 a batch download or OAI. In order to update the local Aleph catalogues ZDB data replication allows the online update via Aleph Cluster replication.

3.1 Functionality - ZDB Data Replication

3.1.1 Central System

Updates 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) is sent to the local systems.

3.1.2 Local System

The local system distinguishes between new HOL records, updates, and deletions:

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

It is 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` fetches this BIB record from the central catalogue and creates a new BIB record. This mechanism is also used if the content of the tag 012 (LKR) has been changed and the HOL record needs to be linked to a different BIB record that is not part of the local BIB library.

When a HOL record is deleted in the local system, `ue_11` tries to delete item records that are linked to the HOL record via `Z30-HOL-DOC-NUMBER`. Standard check routines are performed before the item record is deleted. If the deletion is not possible, error messages are logged.

In addition, `ue_11` deletes the BIB record linked to the HOL record that is to be deleted. The `ue_11` also deletes the any old BIB records that the HOL record was linked to in the past. Users have to make sure via a proper definition of check doc routines that the BIB record is only deleted if no other relevant links exist.

Note that a BIB record cannot be deleted if there are any remaining items linked to it. Items are deleted only via the `Z30-HOL-DOC-NUMBER` link before the BIB record is deleted.

3.2 Setup - ZDB Data Replication

3.2.1 Central System

tab z105

To replicate redirections from the central to the local system, an entry with message type (g) needs to be defined in tab_z105.

Example: tab_z105
(in data_tab of the 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 z105 filter

Since only updates of ZDB HOL records should trigger these new messages, the tab_z105_filter needs to be adjusted accordingly.

Example: tab_z105_filter
(in data_tab of the 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, define a check routine in `tab_ue11.conf`.

Example: `tab_ue11.conf`
(in `data_tab` of the `z105_library`)

```
[A]
    FIX-NEWDOC = INS2

[A-F]
CHECK_TYPE = CATALOG_DELETE
```

FIX-NEWDOC

A fix routine can be defined for creating records. The fix routine must exist in the `tab_fix` table of the BIB library.

CHECK-TYPE

Validation checks can be defined for deleting records. The check type defined here must exist in table `check_dock` of the BIB library.

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

Updates and deletions 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 is 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` checks every message for relevancy for the local system, based on the OWN (sub-library) information. For this purpose the OWN (sub library) information is 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) is considered for this check. If no sub-library code similar to the OWN (sub library) information is found, the message is deleted without further processing.

If the message is relevant for the local system, `ue_11` retrieves the corresponding HOL record from the central system. If the record exists in the central system, `ue_11` updates the local version or creates a new record, if no version exists locally. If the record does not exist in the central system, `ue_11` deletes the local version.

If the linked BIB record does not exist in the central system, `ue_11` retrieves the record from the central catalogue and adds the record to the local BIB library.

3.3.2 `ue_11` Log File

All messages sent 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-00000039
```

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

The message "...Update : 2 00 CEN61 LOC62:000000005 \$aBC-00000039" can be ignored. For all updates of local HOL records, such messages are recorded but should not be sent to the central system for ZDB records, based on `tab_z105_filter`.

If the deletion of an item record failed, an appropriate message is logged, for example:

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

4 Redirection of Central Bibliographic Record

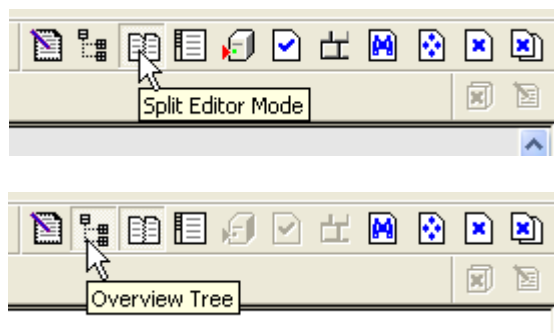
Central bibliographic redirections are required to enable manual data maintenance in the central catalogue. The redirection of a central BIB record is replicated automatically via Aleph cluster in order to perform this redirection in the local systems, as well.

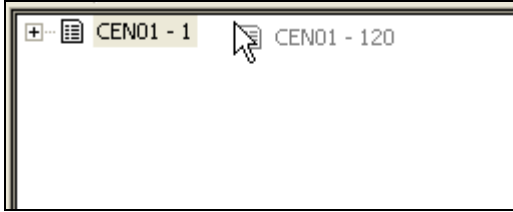
4.1 Functionality

The functionality consists of a central part that describes the functionality in the central catalogue and a local part that refers to how the messages are operated in the local system.

4.1.1 Central System

Redirection can be performed in the Aleph GUI in split editor mode by dragging a BIB record with the mouse to a second record in the overview tree:





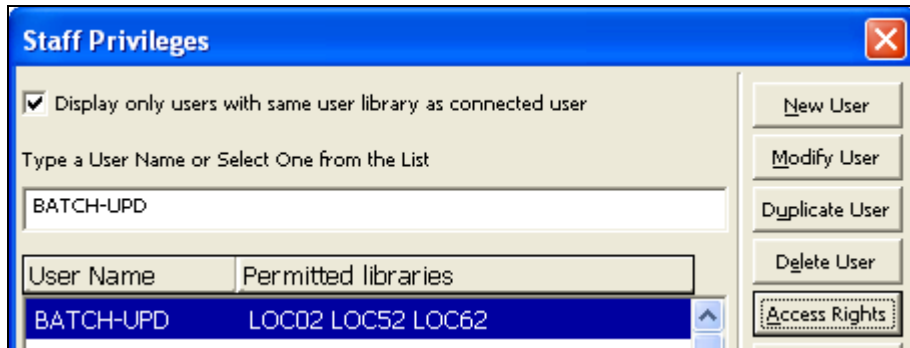
This mechanism is only available if it is activated in `tab_move_record`. The following tasks are performed:

- Re-linking of all HOL records from the source to the target
- Re-linking of all linked BIB record from the source to the target according to `tab_relink_link` (MAB only)
- Moving 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
- Deletion of the source record if all the tasks listed above are successfully performed and no remaining lower level records are linked to the source record
- Updating of redirected records trigger z105 update messages with message type (g).

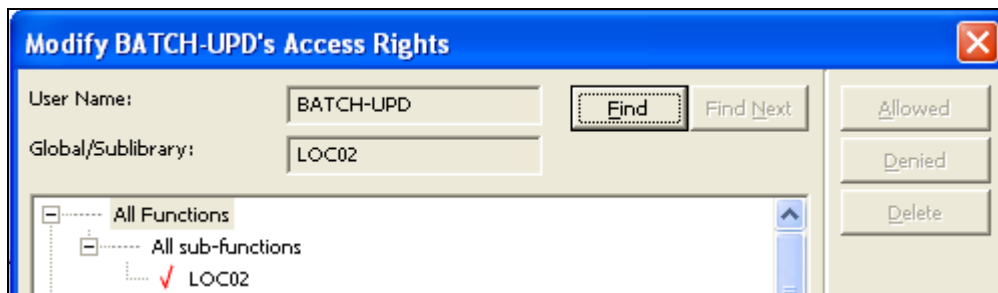
Depending on the amount of linked records it may be possible that not all lower level records could be re-linked to the target record before an internal time-out. In this case, the source record remains in the system and the redirection needs to be performed manually again in order to finalize the redirection. Deletion of z300 is only performed after all re-linking activities listed above have been finished in order to prevent the 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 libraries:



Allow all sub-functions for all functions:



The local `ue_11` first checks if the source record exists in the local system. This is a precondition for further proceeding. If it exists, `ue_11` checks if the target record is already available locally. If the target is already linked to an ADM record, checks are performed according to `move_adm_to_adm`. If the check result prevents the redirection, the whole process is stopped. This is reported to the `ue_11` log file.

If the target record does not exist, the `ue_11` retrieves the target record from the central system and creates the record locally.

The following tasks are performed as part of the local redirection:

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

- Re-linking 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.)
- Deletion of the source record if all the tasks listed above are successfully performed and no remaining lower level records are linked to the source record
- Transfer of the tags of the source record to the target record based on merge definition (`tab_merge`)
- Updating 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 the 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

To replicate redirections from the central to the local system, an entry with message type (g) needs to be defined in `tab_z105`.

Example: `tab_z105`
(in `data_tab` of the BIB library)

!												
!	1	2	3	4	5	6	7	8	9	10	...	
!!!!!!!!!!!!!!-!-!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!-...												
RELINK-DOC	g	XXX10	YYY10	ZZZ10								

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 need to be defined in tab_relink_link.

Example: tab_relink_link
(in data_tab of the 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 need to be defined in tab_relink_link.

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 in tab_merge_overlay. Note that the merge set of merge routine UE11-1 cannot be used because source and target record are inverted.

tab relink link (ADM)

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

Example: tab_relink_link
(in data_tab of the ADM library)

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

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

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 local BIB library target based on `tab_z105`. `ue_11` transfers the messages to the target libraries. The following information is 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` first checks if the source record exists in the local system. If it does not exist, the message is deleted without further processing.

If the target record does exist, `ue_11` retrieves the target record from the central system and creates the record locally.

When the ADM data is moved to the target record, the local `ue_11` sends z105 messages for each z30 record. The central `ue_11` retrieves 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 previously deleted by the central redirection process.

During the redirection process, many records can 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 trigger such messages. These records are also re-linked by the replication process; therefore, these messages are not relevant.

4.3.2 `ue_11` Log File

All messages send during the replication process are logged in the `run_e_11.nnnnn` file 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 that 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 causes a change in the link, the new father record is 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. Therefore, if a father record is missing for other reasons, it is also fetched automatically.

5.1 Functionality - Automatic Replication of Parent Records

During the 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 either update local BIB records or are used to create a corresponding BIB record locally.

If the updated record taken from the central catalogue by the ue_11 contains link information, ue_11 compares the link information of the old record version stored locally and the new version provided by the central BIB. The tags that should be considered for this check are defined in `tab_relink_link`. If ue_11 detects differences concerning the link information, it checks 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 tries 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 asks the central PC server to provide full record information of the target to create this record locally before the update or creation of the originally provided BIB record is performed.

If the required BIB record cannot be provided by the central catalogue, an error message is reported to the 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 the z105_library)

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

FIX-NEWDOC

A fix routine can be defined for the creation of records. The Fix routine must exist in tab_fix of the BIB library.

tab relink link

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

Example: tab_relink_link
(in data_tab of the 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 is 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 retrieves this record from the central system and adds it to the local BIB library. If the record does not exist in the central system, an error message is logged.

5.3.2 ue_11 Log File

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

```