USER DOCUMENTATION

How to Use the RLIN Loader
## Table of Contents

1. OVERVIEW ................................................................. 3
2. SETUP OF THE UE_03 DAEMON (RLIN LOADER) ........... 3
3. RLIN LOADER-RELATED TABLES .................................. 4
   3.1 tab_rlin ................................................................. 4
   3.2 tab_z30 ................................................................. 5
   3.3 tab_mapping .......................................................... 6
   3.4 tab_merge ............................................................ 7
   3.5 tab_match ............................................................. 8
   3.6 tab_fix ................................................................. 9
4. MATCHING AND MERGING INCOMING RLIN RECORDS .... 9
1 Overview

Note:
This document focuses on the loading of MARC bibliographic records from the RLIN bibliographic utility into ALEPH.

RLIN® — the Research Libraries Information Network — is an internationally available bibliographic information system, used with the Research Library Group's library resources.

The RLIN record loader is a daemon (UE_03) which runs continuously in the background. It scans specified directories for new MARC-formatted files every 30 seconds, unless otherwise timed, and loads them into the active library in ALEPH.

The incoming record is passed through fix routines specific to RLIN. Additional institution-specific fixes for incoming RLIN data can also be applied. These routines can be directory-sensitive. When loading RLIN records, you can have the system create item records and/or holdings records automatically upon import.

The loader can run in debug mode, in which the database is not changed, according to the DEBUG environment variable. If this variable is set to Y, the input file is processed fully, but not added to the library.

2 Setup of the UE_03 Daemon (RLIN Loader)

In order for the daemon to function properly, carry out the following steps:

1. In the data_root directory of the XXX01, define the input/output directories:
   There may be more than one input directory.

   dlib XXX01
dr
mkdir rlin_output_dir1
mkdir rlin_output_dir2
mkdir rlin_input_dir

2. Add the following lines to tab100 of the bibliographic library to set up the directories:

   RLIN-INPUT-DIR=rlin_input_dir1,rlin_output_dir2
   RLIN-OUTPUT-DIR=rlin_output_dir

3. Add the following line to prof_library to set the hours that UE-03 (the RLIN loader) is active:

   setenv ue_03_active_hours 00-23

4. Move tab_rlin from the $data_tab directory to the data_tab directory of the bibliographic library.
5. Set daemon fix routines and flags for the scanned directory (see the $data_tab
directory section below.

3 RLIN Loader-Related Tables

3.1 tab_rlin

Here is an example of the tab_rlin table in the $data_tab directory:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>BIB</td>
<td>USM01</td>
<td>Y</td>
<td>Y</td>
<td>RLIN</td>
<td>RLIN</td>
<td></td>
</tr>
<tr>
<td>02</td>
<td>BIB</td>
<td>USM01</td>
<td>N</td>
<td>N</td>
<td>RLIN</td>
<td>RLIN</td>
<td></td>
</tr>
<tr>
<td></td>
<td>AUT</td>
<td>USM10</td>
<td>N</td>
<td>N</td>
<td>RLIN</td>
<td>RLIN</td>
<td></td>
</tr>
<tr>
<td></td>
<td>AUT</td>
<td>USM10</td>
<td>N</td>
<td>N</td>
<td>RLIN</td>
<td>RLIN</td>
<td></td>
</tr>
</tbody>
</table>

- Column 1 - 01, 02. corresponds to the order in which the scanned directories
  are defined (in this example 01 corresponds to rlin_input_dir1, 02
  corresponds to rlin_input_dir2).

- Column 2 - contains the three-letter code of the database type: BIB =
  bibliographic database, AUT = authority database.

- Column 3 - lists the name of the library which corresponds to the database
  type: For example the name of the bibliographic database in Ex Libris
  University is EXU01. The name of the authority file is EXU10.

- Column 4 - by default, the program applies all the fix routines under the RLIN
  section in $data_tab/tab_fix to each incoming record. This is the
  appropriate place to specify global fixes, such as, for example, creating a 035
  field from 001/003 - being done by fix_doc_rlin_1. Refer to the tab_fix
  section (3.6) for more information.

- Column 5 above can be set to Y (or N) to create (or not) an item record for
  each incoming RLIN record.

- Column 6 above can be set to Y (or N) to create (or not) a holdings record for
  each incoming RLIN record.

- Column 7 – match routine - determines which MARC field in the incoming
  record will be matched to avoid duplication in the database (corresponds to
  Column 1 in the tab_match table). Refer to the tab_match section (3.5) for
  more information.

- Column 8 – merge routine - this value indicates which routine from the
  tab_merge table will be followed when a bibliographic or authority record in
  ALEPH is created or overlaid. By default, the system performs merging
  according to the routines specified for the RLIN routine in the tab_merge
  table. Refer to the tab_merge section (3.4) for more information.
3.2 tab_z30

Here is an example of the tab_z30 table:

```
! 1 2 3 4 5 6 7 8
!!!!!!!!!!!-!!!!!!!!!!!-!!!!!!!!!!!-!!!!!!!!!!!-!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!-
DEF UEDUC GEN 11 BOOK N
XZLA UEDUC REF 72 BOOK N
XZLD UEDUC GEN 11 BOOK N
```

This table is used for setting the default values of item records that are system-created through the OCLC server and the RLIN loader. The tab_z30 table, which is located in the XXX50 tab directory, defines the sublibrary, collection, item status, material type, and the location that will be assigned to the new item record, based on the 950 $l field in the incoming bibliographic record.

The item records are never created automatically when a bibliographic record is being overlaid. Their creation is determined in tab_rlin discussed above.

The header information in tab_z30 specifies the character length of each column as well as the purpose of each column. For example, Column 7 can accommodate up to 40 characters. The table can hold up to 1000 lines.

It is very important to have a default line as the first line in this table. The default code is DEF (see above). If the holding code from the 950 field in the RLIN record does not match any of the entries in tab_z30, it looks for a default line with DEF as its holding code.

- Column 1 - indicates the holding code that appears in the 950 $l RLIN tag of the bibliographic records. In a multi-ADM environment, the content of the field can appear only once within all the tab_z30 tables in the various ADM libraries.
- Column 2 – is used by OCLC server. Not relevant for RLIN loader.
- Column 3 - lists the sublibrary of the item.
- Column 4 - shows the collection to which the item belongs.
- Column 5- item status.
- Column 6 - material type code of the item.
- Column 7 - indicates whether or not the physical piece is oversize.
- Column 8 - specifies the item call number. Normally this column is left blank. If an institution imports numerous bibliographic records using a specific call number for a set of materials, that call number can be put in this column for
the duration of the project. If this column is left empty, the call number mapped in tab_mapping is taken.

3.3 tab_mapping

Here is an example of the tab_mapping table. The table resides in the $data_tab directory of the bibliographic library and in the administrative libraries as well. RLIN uses the tables which reside in the administrative libraries.

In tab_rlin (above), when there is a Y in Column 6, the system automatically creates a holdings record during the loading process when a new bibliographic record is added. The tab_mapping file defines where information from the RLIN record is placed in the ALEPH holdings record, and is essential for the automatic creation of holdings records during the import of RLIN records. The table below illustrates an institution with two RLIN holding code symbols (BNGG, the main library holding code and BNGD, the government documents holding code).

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>1</td>
<td>2345</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>BNGG</td>
<td>049## a 8520 b s/BNGG/XBIMA/</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BNGG</td>
<td>049## a 852 c s/BNGG/MAIN/</td>
<td>Y</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BNGG</td>
<td>050## a 852 h</td>
<td>Y</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BNGG</td>
<td>050## b 852 i</td>
<td>Y</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BNGG</td>
<td>090## a 852 h</td>
<td>Y</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BNGG</td>
<td>090## b 852 i</td>
<td>Y</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BNGG</td>
<td>086## a 8523 h</td>
<td>Y</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BNGG</td>
<td>099## a 8524 j</td>
<td>Y</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BNGD</td>
<td>049## a 8520 b s/BNGD/XBIMA/</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BNGD</td>
<td>049## a 852 c s/BNGD/MGDOC/</td>
<td>Y</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BNGD</td>
<td>050## a 852 h</td>
<td>Y</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BNGD</td>
<td>050## b 852 i</td>
<td>Y</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BNGD</td>
<td>090## a 852 h</td>
<td>Y</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BNGD</td>
<td>090## b 852 i</td>
<td>Y</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BNGD</td>
<td>086## a 8523 h</td>
<td>Y</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BNGD</td>
<td>099## a 8524 j</td>
<td>Y</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Each of the columns in the table above indicates how many characters the length of the column can contain. For example, Column 6 can accommodate up to 30 characters.

- Column 1 - specifies the RLIN 950 $l$ code of the library.
- Column 2 - the source code from the RLIN record. These are the tags where the information is originally located. The call number tags in the example above are listed (from the top down) in the order of least preferred.
- Column 3 - indicates the source subfields (from within the source code fields).
- Column 4 - lists the target code in the ALEPH Holdings record where incoming data are placed. It can also accommodate the first indicator of the 852 field (which specifies call number type) as shown in the example above.
• Column 5 - specifies the target subfields for the exact location of the incoming data in the ALEPH Holdings record.

• Column 6 – this is where textual information may be entered. For example, on line 1 in the table above the formula, \(s/BNGG/BXIMA/\) instructs the system to substitute the text XBIMA for the text BNGG in subfield b of the 852 field when the data is coming from the subfield l in the 950 field of the RLIN record. On line 2 in the example above, the formula \(s/BNGG/MAIN/\) tells the system to substitute the text MAIN for the text BNGG in subfield c of the 852 field when the data is coming from the subfield l of the 950 field in RLIN. You can also append information using an a for append which adds a prefix/suffix to the input code. For example, this line in tab_mapping:

```
BNGD 050 a 852 b a/<foo_>/<_bar>
```

produces

```
foo_BNGD_bar as 852 subfield b.
```

• Column 7 is the overlay flag. If it is set to Y, all incoming data will overlay any previous data. If it is set to N, there will be no overlay of data

• Column 8 is the New Line Flag which can be set to N or Y. When it is set to N a new line is not created and, based on the value in column 7, data can be overlaid in the ALEPH Holdings record.

### 3.4 tab_merge

Here is an example of the `tab_merge` table in the $data_tab directory:

```
!  1  2  3
 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
OVERLAY-01 merge_doc_overlay 01
OVERLAY-02 merge_doc_overlay 02
OVERLAY-03 merge_doc_overlay 03
OVERLAY-04 merge_doc_overlay 04
RLIN   merge_doc_overlay 01
OCLC   merge_doc_overlay 03
TEST   merge_doc_overlay 03
HVD    merge_doc_adv_overlay 01
```

The `tab_merge` table is located in the bibliographic library in the tab directory.

The three columns in this table list the merge routines, the routines’ program name and the programs’ arguments (the section identifier of `tab_merge_overlay` and `tab_merge_adv_overlay`). The program for RLIN is `merge_doc_overlay` which calls a specific section of the `tab_merge_overlay` table. In this case, the 01 section in `tab_merge_overlay` contains the following lines:
!1 2 3 4
!!-!-!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
01 1 N #####
01 2 Y #####

- Column 1 - contains the merge set appears in column 3 of tab_merge.
- Column 2 - defines lines for the original record (value=1), the document into which fields are pasted, as well as defining the lines for the document from which the fields are copied (value=2).
- Column 3 - determines what the final form of the bibliographic record includes. This is based on the fields from the original record, together with the fields from the copied record, depending on the values Y, N and C.
- Column 4 - contains the lines taken from each document.

In this case, all the lines are taken from the incoming record while none are taken from the existing one. All lines are copied from the incoming record into the existing record.

ue_03 automatically refers RLIN records to the tab_preferred table. tab_preferred lists the table to use for setting the preferred document (that is, the document which is merged into). If the library does not want to specify which is the preferred record (that is, the incoming record is always merged into the database record), the table referred to in Col.3 of tab_preferred must be set to a name such as db_always_preferred, and an empty table called db_always_preferred must be set up.

3.5 tab_match

Here is an example of the tab_match table in the $data_tab directory:

<table>
<thead>
<tr>
<th>!</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!</td>
<td></td>
<td></td>
</tr>
<tr>
<td>YBP match_doc_uid</td>
<td>I-ISBN</td>
<td></td>
</tr>
<tr>
<td>YBG match_doc_gen</td>
<td>I-ISBN</td>
<td></td>
</tr>
<tr>
<td>RLIN match_doc_uid</td>
<td>T-020</td>
<td></td>
</tr>
<tr>
<td>OCLC match_doc_script</td>
<td>tab_match_script_oclc</td>
<td></td>
</tr>
<tr>
<td>MARC match_doc_uid</td>
<td>I-909</td>
<td></td>
</tr>
<tr>
<td>CAT match_doc_uid</td>
<td>I-ISBN</td>
<td></td>
</tr>
<tr>
<td>CAT match_doc_acc</td>
<td>tab_match_acc</td>
<td></td>
</tr>
<tr>
<td>P36 match_doc_acc</td>
<td>tab_match_acc</td>
<td></td>
</tr>
</tbody>
</table>

You must ensure that the tab_match table includes the RLIN match routine, the program name and the program arguments. It is located in the bibliographic library in the tab directory.

- Column 1 - the name of the match routine code. For loading RLIN records the match routine code is RLIN.
- Column 2 - lists the match program used by the match routine. The program used by the RLIN match routine is match_doc_uid.
• Column 3 - indicates the program arguments. In this case, the argument is T-020 which refers to a tag in column 1 of tab1_ind.

3.6 tab_fix

Here are examples of RLIN fix_doc programs in the tab_fix table from the bibliographic library:

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>!1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RLIN</td>
<td>fix_doc_tag_008</td>
<td></td>
</tr>
<tr>
<td>RLIN</td>
<td>fix_doc_tag_008_open_date</td>
<td></td>
</tr>
<tr>
<td>RLIN</td>
<td>fix_doc_punctuation_usm</td>
<td></td>
</tr>
<tr>
<td>RLIN</td>
<td>fix_doc_005</td>
<td></td>
</tr>
<tr>
<td>RLIN</td>
<td>fix_doc_rlin_1</td>
<td></td>
</tr>
<tr>
<td>RLIN</td>
<td>fix_doc_usm_001</td>
<td></td>
</tr>
<tr>
<td>RLIN</td>
<td>fix_doc_non_filing_ind</td>
<td></td>
</tr>
</tbody>
</table>

• Column 1 - routine name

• Column 2 – fix program

• Column 3 - if program arguments are needed, they will appear here.

4 Matching and Merging Incoming RLIN Records

To set up matching and merging of incoming RLIN records, carry out the following:

1. In tab_rlin, Col.7, enter the name of a match section in tab_match to be used when ue_03 runs, for matching incoming RLIN records.

2. In tab_match, define a match routine, selecting the index against which you wish to match incoming RLIN records (such as for example, the direct index of 035).

3. Once the match has been performed, the merge or overlay will be executed according to the routine specified in tab_rlin, Col.8. Enter there RLIN (note that RLIN is a reserved name; see explanation in tab_match’s header).

4. Define the routine also in tab_merge.

5. Check that tab_merge, Col.3 for RLIN defines the section identifier of tab_merge_overlay.

6. Make sure that the fields specified in tab_merge_overlay under that section are indeed those that you want to merge or overlay.