USER DOCUMENTATION

How to Set Up and Configure Union View for Your OPAC

Last Update: November 11, 2009
Document version 1.3

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.

Documentation produced January 2004.

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

Table of Contents

1. Introduction: What is Union View?

Union View is a tool for creating de-duplicated and merged result sets in the public OPAC. Result sets are de-duplicated using pre-constructed record equivalency tables. The tables are built using a sophisticated algorithm developed in conjunction with the California Digital Library (CDL) project.

1.1 Who needs Union View?

Union View is ideal for environments where there are two or more institutions sharing a database but not sharing bibliographic records. In these environments, there will be several versions of the same record; staff needs to see all versions but end users do not. Union View offers a convenient tool for end-user navigation and resource discovery that does not interfere with staff productivity.

1.2 Setting up Union View

The Union View functionality is easy to set up and use. The package is an add-on that can be set up and implemented during or after ALEPH implementation. Several programs make use of existing ALEPH indexes, so sites interested in implementing a Union View should be sure to include the necessary indexes.

1.3 Why have a Union View?

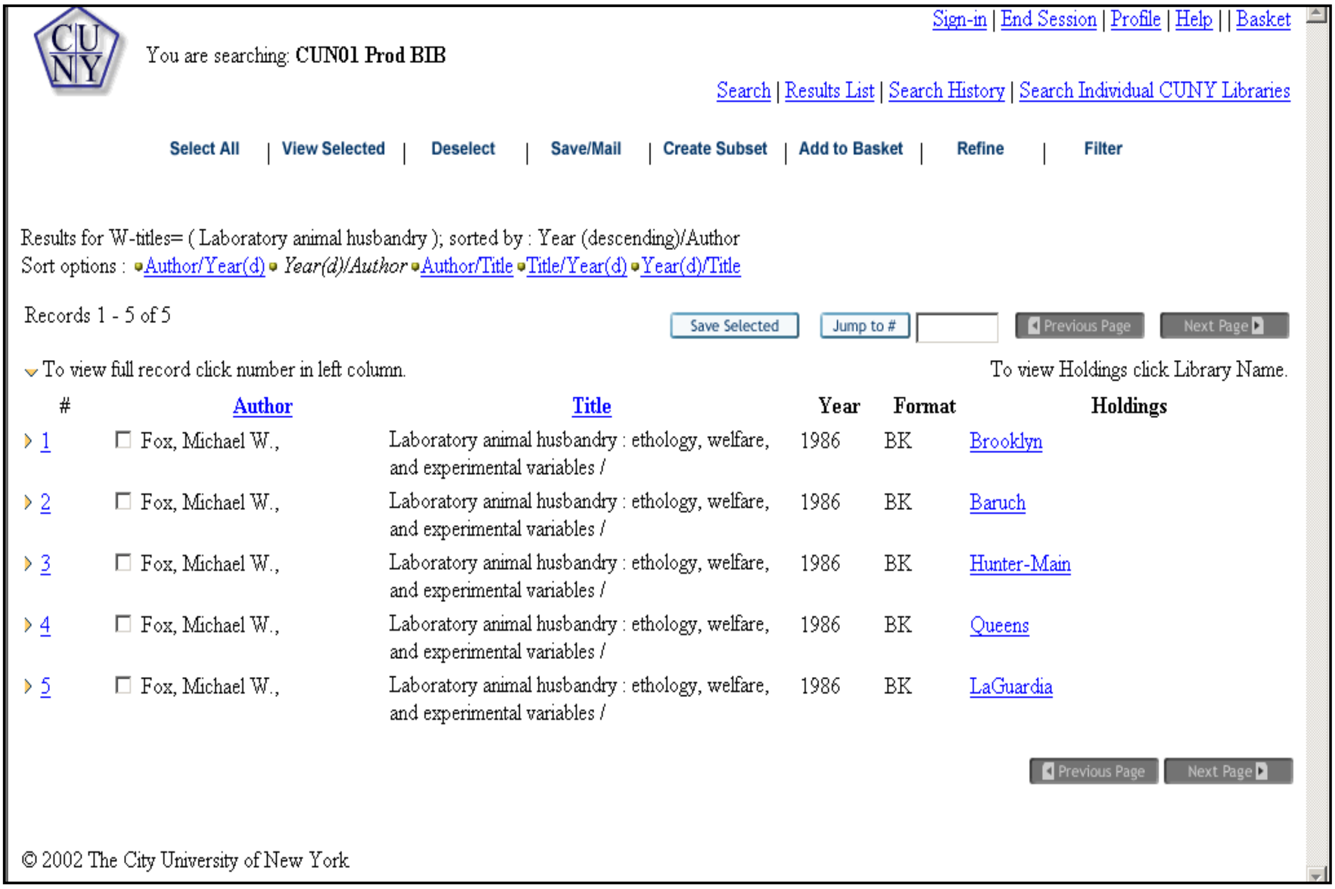
Union View provides useful functionality to the end user and does so without impacting on response time (record equivalencies are pre-built). It also requires almost no extra storage space, very little setup, and because it is modular, it can be turned on and off without impacting the overall system.

1.4 Who has access to the Union View?

In ALEPH, the Union View is configured as a logical base and it is available in the Web OPAC and in z39.50. Union View can be the only logical base you offer the public, or more commonly, it is one of many bases you offer. In a typical Union View environment, there are logical bases that correspond to each of the individual institutions as well as a Union View base, a Union View of Serials, etc.

1.5 Illustrations

Search in Standard OPAC (without Union View):



Sign-in | End Session | Profile | Help | | Basket
Search | Results List | Search History | Search Individual CUNY Libraries
Select All | View Selected | Deselect | Save/Mail | Create Subset | Add to Basket | Refine | Filter
Save Selected
Jump to #
Previous Page
Next Page

You are searching: **CUN01 Prod BIB**

Results for W-titles= (Laboratory animal husbandry); sorted by : Year (descending)/Author
Sort options : Author/Year(d) *Year(d)/Author* Author/Title Title/Year(d) Year(d)/Title

Records 1 - 5 of 5

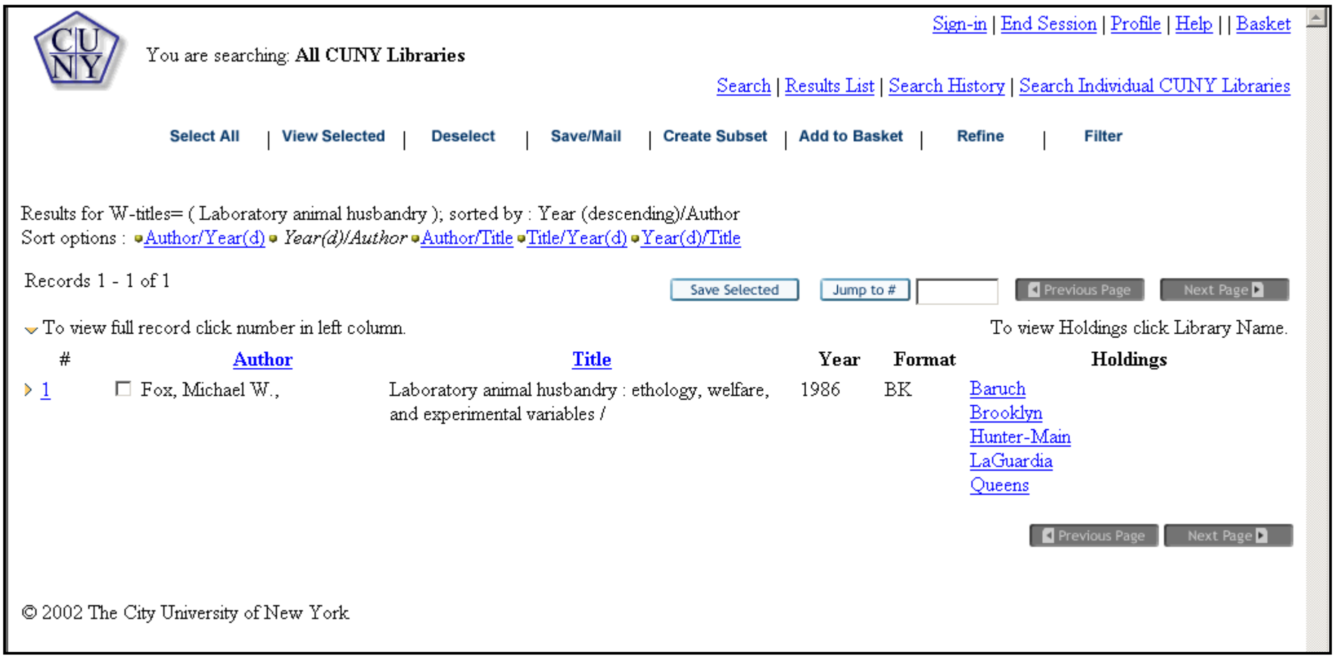
To view full record click number in left column.

To view Holdings click Library Name.

#		Author	Title	Year	Format	Holdings
1	□	Fox, Michael W.,	Laboratory animal husbandry : ethology, welfare, and experimental variables /	1986	BK	Brooklyn
2	□	Fox, Michael W.,	Laboratory animal husbandry : ethology, welfare, and experimental variables /	1986	BK	Baruch
3	□	Fox, Michael W.,	Laboratory animal husbandry : ethology, welfare, and experimental variables /	1986	BK	Hunter-Main
4	□	Fox, Michael W.,	Laboratory animal husbandry : ethology, welfare, and experimental variables /	1986	BK	Queens
5	□	Fox, Michael W.,	Laboratory animal husbandry : ethology, welfare, and experimental variables /	1986	BK	LaGuardia

© 2002 The City University of New York

Search in Union View OPAC



Sign-in | End Session | Profile | Help | | Basket

You are searching: **All CUNY Libraries**

Search | Results List | Search History | Search Individual CUNY Libraries

Select All | View Selected | Deselect | Save/Mail | Create Subset | Add to Basket | Refine | Filter

Results for W-titles= (Laboratory animal husbandry); sorted by : Year (descending)/Author
Sort options : Author/Year(d) *Year(d)/Author* Author/Title Title/Year(d) Year(d)/Title

Records 1 - 1 of 1

Save Selected
Jump to #
Previous Page
Next Page

To view full record click number in left column. To view Holdings click Library Name.

#		Author	Title	Year	Format	Holdings
1	□	Fox, Michael W.,	Laboratory animal husbandry : ethology, welfare, and experimental variables /	1986	BK	Baruch Brooklyn Hunter-Main LaGuardia Queens

© 2002 The City University of New York

2. Configuring a Union View Catalog

2.1 Enabling Union View in your library

To enable Union View functionality in your library, the library must be defined as a Union View library. This is set in the UNION-LIBRARY variable in tab100. The appropriate value is 1.

2.2 Setting up Union View bases

Logical bases (defined in ./alephe/tab/tab_base.<lng>) that begin with U- are Union View bases. Libraries can set up multiple Union View and non Union View bases. In Union View libraries, bases that begin with U- will use pre-existing record equivalency tables to de-duplicate the result set; whereas bases that do not begin with U- will not be de-duplicated and retrieval sets will display all versions of equivalent records.

Note that the only difference between Union View bases and all other bases is that the result set in Union View bases uses record equivalency tables. They are identical in all other ways. Thus, please be sure to set all columns in tab_base.lng and be sure to run the same processes as are run for any logical base (p_manage_32). See ALEPH documentation for more information on Logical bases.

3. Configuring Your OPAC Interface for Union View

3.1 Union View settings in www_server.conf

The following setting in www_server.conf applies to Union View bases in version 14.2 and 15.5. This setting was removed from version 15.2, 16.2 and all subsequent versions:

Illustration 1 - `www_server.conf [alephe_root]`

```
setenv www_sort_limit               800
```

The sort limit is also used as the de-duplication limit. Only that number of records will be de-duplicated. This means that if a retrieval set is larger than the limit, only part of the retrieval set will be de-duplicated. The de-duplicated records should be the first to be displayed. In versions without this setting, the system has no limit on the retrieval set size, but there is a system wide limit of 1,000 records for display, sorting and de-duplicating.

3.2 Merged display

The system uses the configuration table tab_merge_union located in the data_tab directory of the bibliographic library to configure the merged display. This table controls record merging for display and allows you to set, on a tag-by-tag basis, which fields will be displayed from the preferred and non-preferred records.

Illustration 2 - tab_merge union [data_tab]

```
!1 2 3          4
!!-!-!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
90 1 Y #####
90 1 N SID##
90 1 N 852##
90 1 N 856##

90 2 Y 5050#,u,*
90 2 Y 852##
90 2 Y 856##
```

Key to the table:

Column 1 – Merge set – will always be 90 for Union Catalog merge

Column 2 – Merging direction (1 refers to preferred record, 2 to each equivalent record)

Column 3 – Action (Y, N, C) Y – retain field, N – delete field, C – retain field if it is original)

Column 4 – Field tag.

Column 4 (continuation after ,) –subfield and contents to match on. Example: 01 2 Y 590##,5,*abc* (meaning “if there is a subfield 5 which contains ‘abc’)

Note: The program expand_doc_merge_union must be called in tab_expand for each instance where a merged display is required.

For example, a Union Catalog site might want to have a merged display in the Web OPAC and Z39.50 but not in the GUI. If this were the case, tab_expand would be set up as follows:

Illustration 3 – tab_expand [data_tab]

```
WEB-FULL     expand_doc_merge_union

WEB-MAIL     expand_doc_merge_union

Z39_SERVER   expand_doc_merge_union
```

3.3 Displaying items and holdings

Union View libraries should use the ITM1 and/or the ITM3 links to display holdings in the Web OPAC. LOC and PST expands do not work in a Union View environment. More information about these holdings links can be found in the header of `edit_doc_999.eng`.

For libraries that choose to use the ITM1 link, the order of Holdings records in the "all items" window can be controlled. In versions 14.2 and 15.2, the ./`alephe/tab/tab_base.conf` table configures the appropriate field for obtaining the value to be sorted on and lists the correct sort order. From version 16, the library's `tab_expand_local_notes.conf` table is used for this purpose. The library's `tab_expand` table must include a reference to `expand_doc_bib_local_notes` in the relevant sections (WEB-FULL and WEB-SHORT) of the table. Note that if you plan to implement `tab_base.conf`, the table `www_server.conf` (`www_server_defaults` in version 14.2) will need to be set up appropriately. The appropriate settings are included below in *Illustration 5*.

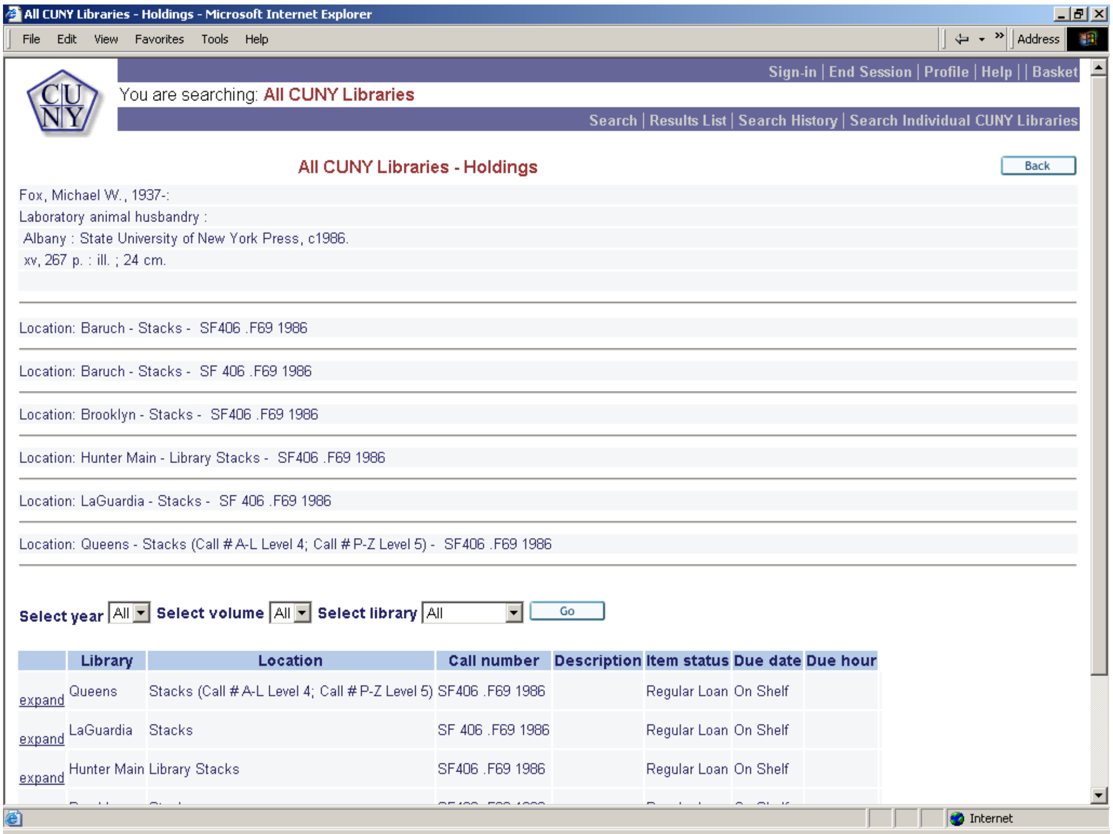
Illustration 4 – `tab_base.conf [alephe_tab] / tab_expand_local_notes.conf [library's tab]`

```
[U-USM01]
     owner tag =OWN
     owner subfield = a
     owner alternative tag = 590, 690
     owner alternative subfield = 9
     mapping section = LCN-2-BIB
```

Illustration 5 – `www_server.conf [alephe]`

```
setenv hol_sort_op           BASE
setenv hol_filter_op         BASE
```

Illustration 6 - “All items” display: [web OPAC]


All CUNY Libraries - Holdings - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address
Internet
CU NY
Sign-in | End Session | Profile | Help | | Basket
Search | Results List | Search History | Search Individual CUNY Libraries
Back
Go

You are searching: All CUNY Libraries

All CUNY Libraries - Holdings

Fox, Michael W., 1937-:
Laboratory animal husbandry :
Albany : State University of New York Press, c1986.
xv, 267 p. : ill. ; 24 cm.

Location: Baruch - Stacks - SF406 .F69 1986

Location: Baruch - Stacks - SF 406 .F69 1986

Location: Brooklyn - Stacks - SF406 .F69 1986

Location: Hunter Main - Library Stacks - SF406 .F69 1986

Location: LaGuardia - Stacks - SF 406 .F69 1986

Location: Queens - Stacks (Call # A-L Level 4; Call # P-Z Level 5) - SF406 .F69 1986

Select year All Select volume All Select library All

	Library	Location	Call number	Description	Item status	Due date	Due hour
expand	Queens	Stacks (Call # A-L Level 4; Call # P-Z Level 5)	SF406 .F69 1986		Regular Loan	On Shelf	
expand	LaGuardia	Stacks	SF 406 .F69 1986		Regular Loan	On Shelf	
expand	Hunter Main	Library Stacks	SF406 .F69 1986		Regular Loan	On Shelf	

4. Equivalent Records

4.1 What are equivalent records?

Equivalent records are records that are in the same database and are similar enough to be considered duplicate. Equivalent records are usually owned by different institutions and have different OWN fields and holdings locations. ALEPH uses an algorithm that was developed in conjunction with Melvyl (California Digital Libraries), to identify equivalent records.

4.2 Storing equivalent records

There is an Oracle table called z120, which is used to store a list of equivalent records. Note that each bibliographic record has its own z120 record.

The z120 table contains the following data elements: system number, total number of equivalent records, system numbers of all equivalent records, system number of the preferred record, update flag. Z120 records are built after the initial data load (using p_union_01 and p_union_02) and then updated each time any one of the equivalent records is updated. When a bibliographic record is updated, its own z120 record and all z120 records in which it is found are flagged, so that they can be rebuilt.

Illustration 7 – z120 table definition for versions 14.2 –15.5

```
          02 Z120-REC-KEY.
            03 Z120-DOC-NUMBER                   PICTURE 9(9).
          02 Z120-REC-KEY-1.
            03 Z120-PREFERRED-DOC-NUMBER         PICTURE 9(9).
          02 Z120-UPDATE-FLAG                    PICTURE X(1).
          02 Z120-SAME-NO-LINES                  PICTURE 9(3).
          02 Z120-SAME OCCURS 100.
            03 Z120-SAME-DOC-NUMBER              PICTURE 9(9)
```

Illustration 8 – z120 table definitions for version 16.2 +

```
        01  Z120.
          02 Z120-REC-KEY.
            03 Z120-DOC-NUMBER                   PICTURE 9(9).
          02 Z120-REC-KEY-1.
            03 Z120-PREFERRED-DOC-NUMBER         PICTURE 9(9).
          02 Z120-UPDATE-FLAG                    PICTURE X(1).
          02 Z120-SAME-NO-LINES                  PICTURE 9(3).
          02 Z120-SAME OCCURS 500.
            03 Z120-SAME-DOC-NUMBER              PICTURE 9(9).
```

The `z120_update_flag` field has three possible values; N, C, and E. Records that are flagged N need to be rebuilt either by the `p_union_02` or by the `p_union_04` process, C for Checked or already built, and E for Error. A z120 record is flagged E when there is some problem with the data; for example, if there is a z120 record but no bibliographic record.

4.3 Building equivalent record tables

There are two ALEPH processes that build the tables of equivalent records, `p_union_02` and `p_union_04`. `P_union_02` is usually run only once after conversion or after significant changes to the tables that control equivalencies. `P_union_04` is run nightly; it builds equivalencies for new records and rebuilds equivalencies for updated records. Please note that libraries should not run p_union_02 on part but not all of the database. Whenever a portion of the database is being updated, the `p_union_04` process should be used.

`P_union_02` has the following parameters: active library, `start_doc_number`, `end_doc_number`, rebuild links, number of processes, mode – test or batch. In version 16.2 the parameter "rebuild links" was removed.

`P_union_04` has the following parameters: active library

The variable in `prof_library` that controls the `p_union_02` loop length is:

Illustration 9– `prof_library [data_root]`

```
15.2      setenv union_loop_length  10000

16        setenv p_union_02_loop_length    5000
```

For more information on setting loop lengths see the document entitled, *"How to Run Index Jobs"*.

5. Equivalency Algorithm

There are three phases in the equivalency building process:

- Candidate Selection
- Duplicate Detection
- Preferred Record Selection

Each of these phases is controlled by a combination of programs and configuration tables.

5.1 Candidate Selection

During the Candidate Selection phase of the equivalency building process, up to a set number of candidates are retrieved. The number of allowed candidates is set in the table union_global_param. In versions 14.2 through 15.5 this number can not be greater than 100, whereas in version 16.2 and on it can be up to 500.

Searching on the following indexed fields retrieves candidate records:

- LCCN – 010 subfields a and z
- ISSN or ISBN – 020 $$a and $$z and 022 $$a and $$z
- Main title – 245 $$a,b,n,p

Note that there is an "or" relationship between the search terms (lccn **or** issn **or** title)
If more than one hundred records are retrieved, the search is refined. For non-serial records, year of publication is added ((lccn or isbn or title) and year); for serial records, place of publication is added ((lccn or issn or title) and place).

5.1.1 Server tables

The following line should be entered in the server table `union_global_param`. This table controls the individual components of the equivalency algorithm.

Illustration 10 – `union_global_param [alephe_tab]`

```
CUN01 B candidate_prog       union_candidate_cdl
```

5.1.2 Direct indexes

The ISSN, ISBN, and LCCN fields need to be indexed as direct indexes. They should go through expand_doc_extract so that each subfield is indexed separately.

Illustration 11 – `tab11_ind [data_tab]`

```
a010                     010
z010                     010
a020                     020
z020                     020
a022                     022
y022                     022
z022                     022
```

Illustration 12 – `tab_expand[data_tab]`

```
INDEX      expand_doc_extract
```

Illustration 13 – `tab_expand_extract [data_tab]`

```
010## a a010
010## z z010
020## a a020
020## z z020
022## a a022
022## y y022
022## z z022
```

The filing routine below must be used for all three indexes. (This is currently routine 21):

Illustration 14 – `tab_filing`

```
!* The following sixroutines (70, 71, 72, 90, 91) are needed for
UnionView
!* and UnionCatalog
!* 70 - LCCN normalizationfor LCCN index to be used by Equivalency
building
!* program andunion_match_cdl_mo
70  del_subfield
70   compress           -
70  compress_blank
70  non_numeric
!* 71 - ISBN normalizationfor ISBN index to be used by Equivalency
building
!* program andunion_match_cdl_mo
71  isbn
!* 72 - ISSN normalizationfor ISNN index to be used by Equivalency
building
!*  program andunion_match_cdl_se
72   issn
```

Please note that if the ISSN, ISBN and LCCN fields are not correctly indexed, the merge algorithm will not function properly. It is especially important to remove alphabetic characters from the indexes.

5.1.3 Keyword indexes

The main title (245) should be sent to a keyword index called NTL. This index is used for title searching in the Candidate phase of equivalency building. Before indexing a title, an expand program called `expand_doc_ntl` normalizes the title by stripping initial articles (using non-filing indicators), stripping punctuation and diacritics and compressing spaces. The resulting NTL field is indexed as a keyword.

Illustration 15 – `tab11_word [data_tab]`

```
NTL##                    03   NTL
```

Illustration 16 – `tab_expand [data_tab]`

```
WORD        expand_doc_ntl
```

Illustration 17 – `tab00.eng`

```
H NTL   W-035    00     06     Normalized Title
```

5.2 Duplicate detection

The second phase in the equivalency building process is Duplicate Detection. During this phase, all candidate records are compared and those found to be equivalent are recorded in the z120 record. While much of this process is hard-coded, there are a number of configuration tables involved. For more information about the matching algorithm see the document *“Matching Algorithm”*.

5.2.1 Server tables

The following line should be entered in the server table `union_global_param`. This table controls the individual components of the equivalency algorithm.

Illustration 18 – `union_global_param [alephe_tab]`

```
CUN01 B match_prog           union_match_cdl
```

5.2.2 Assigning weights for record matching

The program `union_match_cdl` uses three configuration tables: `tab_cdl_mo_weights`, `tab_cdl_se_weights` and `tab_com_tit_cdl`. These tables set the record equivalency threshold, i.e., the number of points that must be reached in order for records to be considered equivalent. The tables also set the number of points that are assigned for each stage in the match algorithm. The first table, `tab_cdl_mo_weights`, establishes weights for merging formats other than serials; the second table, `tab_cdl_se_weights`, establishes weights for merging serials. Note that the text in the first column is fixed. Both tables are located in the data_tab directory of the Union View library [the xxx01 bibliographic library]. Record equivalency building processes need to be rerun after changes are made to either of these tables.

Illustration 19 - `tab_cdl_mo_weights [data_tab]`

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!>
threshold                                           + 875

010aa                                              + 200
010az                                              + 100
010z#                                              + 050
010mismatch                                        - 320

020aa                                              + 085
020az                                              + 030
020z#                                              + 010
020mismatch                                        - 225
```

```
date exact match                                   + 200
date within 2                                      - 025
date mismatch                                      - 250
```

Illustration 20 `tab_cdl_se_weights [data_tab]`

```
threshold                                          800

010aa match                                        + 200
010az match                                        + 100
010zz match                                        + 050
010aa mismatch                                     - 470
010az mismatch                                     - 050

022aa match                                        + 200
022ay match                                        + 100
022az match                                        + 050
022yy match                                        + 050
022yz match                                        + 030
022zz match                                        + 010
022aa mismatch                                     - 250
```

5.2.3 List of common serial titles

The list of common serial titles is consulted during the matching stage of the cdl equivalency building process. Title matches in serial records that have titles that are considered “common” receive significantly fewer points than title matches for other serial records (135 instead of 600). The table lists “common” titles in a given database or Union Catalog. If the contents of the table change, `p-union-02` should be rerun.

Illustration 21 - `tab_com_tit_cdl [data_tab]`

```
ANNUAL REPORT
ANNUAL REPORT FOR
ANNUAL REPORT FOR THE FISCAL YEAR ENDED MARCH
BIENNIAL REPORT
BULLETIN
CALENDAR
CATALOGUE
CIRCULAR
```

5.3 Additional table settings

There are two additional settings `that` must be in place for the equivalency building processes to work. The first setting is in `tab_expand`. `Tab_expand` has a section that lists expand programs that are to be called by the two equivalency building programs, `p_union_02` and `p_union_04`. Note that both `p_union_02` and `p_union_04` use the UNION-02 section.

Illustration 22 - `tab_expand [data_tab]`

```
UNION-02   expand_doc_extract
UNION-02   expand_doc_ntl
```

These settings should not be changed.

The second table that needs to be set up is `tab_filing`. Three filing routines are used by the equivalency building processes, they are: 90, 91 and 92.

Illustration 23 - `tab_filing [data_tab]`

```
90 # del_subfield
90 # to_lower
90 # non_filing
90 # to_blank             !@#$%^&*()_+-={}[]:";<>?,./~`
90 # char_conv             FILING-KEY-01
90 # compress_blank
90 # first_25

! for match
91 # del_subfield
91 # to_upper
91 # suppress
91 # numbers
91 # compress             '
91 # to_blank             !@#$%^&*()_+-={}[]:";<>?,./~`
91 # expand_num
91 # non_filing
91 # pack_spaces
! for match: ISBN/ISSN
92 # compress             -
92 # compress_blank
92 # non_numeric
```

5.4 Selecting the preferred record

The third phase of the equivalency building process is selecting the preferred record. The program for selecting preferred records is identified in the `union_global_param` table in `alephe_tab` and the routine for selecting preferred records is configured in a single configuration table, `union_preferred.`

5.4.1 Server table

The following line should be entered in the server table `union_global_param`. This table controls the individual components of the equivalency algorithm.

Illustration 24 – `union_global_param [alephe_tab]`

```
CUN01 B preferred_prog       union_preferred_cdl
```

5.4.2 Library tables

The table `union_preferred` configures the basis for selecting a preferred record from a set of equivalent records by assigning points for field presence, and/or subfield or fixed field values. After each record in a set of equivalent records gets a weight, the record with the greatest weight becomes the preferred record.

Illustration 25 – `union_preferred [alephe_tab]`

```
!!!!!-!!!!!!-!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!-!!!
LDR   F05-01 EQUAL      d                          -10
LDR   F17-01 NOT-EQUAL  1,2,3,4,5,7,8,u,z          001
040   a      EQUAL      DLC                        003
100##        PRESENT                               001
110##        PRESENT                               001
111##        PRESENT                               001
130##        PRESENT                               001
24###        PRESENT                               001
6####        PRESENT                               001
700##        PRESENT                               001
710##        PRESENT                               001
711##        PRESENT                               001
730##        PRESENT                               001
800##        PRESENT                               001
810##        PRESENT                               001
830##        PRESENT                               001
880##        PRESENT                               001
```

Key to the table:

Column 1 – Field tag

Column 2 – Subfield or fixed field position and count

Column 3 – Operator (Equal, Not equal, Present):

Column 4 – Subfield or fixed field value. If there are multiple values, separate them with a comma.

Column 5 - Points

6. Batch Processes

There are three processes for building and maintaining record equivalencies.

1. **`p_union_01`** – Run after initial conversion and data load. Builds empty z120 records for each bibliographic record. Parameters are database names.

2. **`p_union_02`** – Run after database has been indexed. Populates empty z120 records. Can be run with multiple processes and on ranges of records. Parameters are database, start number, end number, rebuild links, number of processes, batch/test. This process locks the library.

3. **`p_union_04`** **–** Ongoing, rebuilds record equivalencies for records that have a z120 record flagged N for update. Z120s are flagged N when the corresponding bibliographic record is updated. Note that this service should be restarted periodically –for example, every night (from job_list or online). This is because the service stops when all the Z120 records are finished being updated (with Z120_UPDATE_FLAG=N).Therefore, if new records arrive not from p_union_13, they are not updated until you run p_union_04 again.

4. **create_z127 –** From version 15.2 and on, it is necessary to run this process to create z127 records after running p_union_02 to create z120 records. The process does not have any parameters – the syntax for running it is: `csh –f $aleph_proc/create_z127`

5. **load_z127_to_mem –** After creating z127 records they need to be loaded into the memory. The syntax for running this job is: `csh –f $aleph_proc/ load_z127_to_mem`

If a library is a Union Catalog or a Union View library, new z120 records are created for records that are created in the Cataloging module or loaded into ALEPH using `p_manage_18`, OCLC server, RLIN server, or any other load process.

7. Troubleshooting

The UTIL F menu contains a section for testing the merge algorithm. UTIL/F/21/A allows you to enter a system number and retrieve all “candidate records”. The candidate selection is the first phase of the merge process.

Illustration 25 UTIL/F/21/a

```
***** util_f_21 tests de-duplication *****

  a) find duplicate docs
  b) check if 2 docs are duplicate

 enter option: a
Enter doc number :
000000118
Docs    :
> 000000118
> 000080063
```

UTIL/F/21/b allows you to enter two system numbers and report if they are equivalent. This is the Match phase of the merge process. If you want to see how many points were assigned or lost at each step of the match process, then enter the following command before entering the UTIL menu – “`setenv match_debug Y`”. After entering this command type **“util”**.

Illustration 26 UTIL/F/21/b

```
  b) check if 2 docs are duplicate

 enter option: b
Load: /exlibris/a52_5/alephe/tab/union_global_param
Enter doc number  1:
118
Enter doc number  2:
000080063
Load: /exlibris/u52_5/cun01/tab/tab_cdl_mo_weights
compare-date
COMPARE-RATIO: +000000200
```