



Universal Gateway

Version 18.01 and Later

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: August, 2009

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

Table of Contents

1	OVERVIEW.....	4
1.1	Architecture.....	4
1.1.1	ALEPH GUI Search Workflow.....	4
1.1.2	Web OPAC Workflow	4
1.1.3	Universal Gateway Workflow.....	4
2	UNIVERSAL GATEWAY LIBRARIES.....	5
3	UNIVERSAL GATEWAY CONFIGURATION.....	5
3.1	General Settings.....	5
3.1.1	Find Request Settings.....	6
3.1.2	Response Record Settings.....	6
3.1.3	Scan Request Settings	7
3.1.4	Sort Request settings.....	8
3.1.5	Holdings Information Settings	8
3.1.6	ES Update Settings.....	9
3.1.7	Circulation Status Settings	10
3.2	Transformations	10
3.2.1	Transformation Group Syntax.....	10
3.2.2	Examples.....	12
3.3	SUTRS Configuration.....	13
4	EXAMPLE OF A TARGET CONFIGURATION	16
5	TEST LIBRARY CONFIGURATION.....	17
6	TERM TRANSFORMATION ROUTINES.....	18

1 Overview

The Universal Gateway is the ALEPH utility for accessing remote libraries via the Z39.50 protocol.

1.1 Architecture

1.1.1 ALEPH GUI Search Workflow

- If a library is local - `pc_server` works with the local database.
- If a library is remote - `pc_server` connects to the **Universal Gateway** utility.

1.1.2 Web OPAC Workflow

- If a library is local - `www_server` works with the local database.
- If a library is remote - `www_server` calls to the **Universal Gateway** utility.

1.1.3 Universal Gateway Workflow

The Universal Gateway offers the following options, after it connects to the library using the Z39.50 protocol:

Search Request: transforms the ALEPH query into a query suitable for the library (character conversion, sophisticated term optimization); performs search using the Z39.50 protocol.

Present Request: gets records from the library using the Z39.50 protocol; transform the records into ALEPH format (character conversion, record format conversion, fix routines).

Scan Request: transforms the ALEPH query into a query suitable for the library (character conversion, sophisticated term optimization); performs a scan using Z39.50 protocol; transforms the headings into ALEPH format (character conversion)

Holdings Request: gets holdings from the library using the Z39.50 protocol.

ES Update Request: updates records in a remote catalog using the Z39.50 protocol.

2 Universal Gateway Libraries.

- VIR01 – temporary database, where the Universal Gateway stores records retrieved from remote libraries. This database must be cleaned periodically.

To clean VIR01, from the command line:

```
dlib vir01
util x/8
```

or add `clear_vir01` to `job_list`

3 Universal Gateway Configuration.

For each remote library accessed using the Universal Gateway, a configuration file is created in the `$alephe_gate` directory.

File name: `<target>.conf` (for example, `oxford.conf`), where the target is the logical name of this library used in all levels of ALEPH configuration (for example, `tab_base.<lng>`, `z39_gate` configuration, and so on)

Notes

1. `<target>` is in lowercase
2. The file contains list of settings. Each setting must appear in a separate line.
3. Lines starting with “#” or “!” are comments.

3.1 General Settings.

General Settings define general aspects of the target (access-method, character conversion, and so on).

- **Z58-BASE-NAME** `<name>`

`<name>` is the library name which occurs on WEB OPAC screen.

Must be defined if SID tag or Item Remote link is shown on Full Record screen.

- **Z58-ACCESS-METHOD** `<access-method>`

`<access-method>` is the method by which remote library is accessed. `<access-method>` must be **Z39**.

This setting is mandatory.

Note: `z39_gate` configuration for this library must be defined (see the `z39_gate` document). That is the following files for this library must be defined:

```
$alephe_gate/<target>.conf - <target> is in lowercase. For example:
$alephe_gate/oxford.conf
```

```
$alephe_tab/z39_gate/z39_gate_<target>.conf – the same <target> is in
uppercase. For example, $alephe_tab/z39_gate/z39_gate_OXFORD.conf
```

-Z58-BASE-REMOTE <name>

<name> is the logical name of target used in z39 configuration. (for example, OXFORD). This setting is mandatory

3.1.1 Find Request Settings

- Z58-OUT-FIND-CHAR-CONV <char-conv>

This parameter defines the character conversion routine applied in a search query. ALEPH formulates queries in UTF-8 which must be converted to the character set required by the target database.

The following are possible values of <char-conv>:

- ❑ UTF_TO_ANSEL
- ❑ UTF_TO_MAB
- ❑ UTF_TO_8859_1 and so on.

All of the possible values are defined in:

\$alephe_unicode/tab_character_conversion_line

This setting must be used if the encoding used by the target database for search queries differs from UTF-8.

3.1.2 Response Record Settings

- Z58-IN-RECORD-TYPE <response-record-type>

where <response-record-type> is the type of records in the remote library.

The following values are possible: USMARC, UNIMARC, DANMARC and MAB.

This setting is mandatory.

- Z58-IN-RECORD-CREATE <program>

where <program> is used for converting the incoming records into standard ALEPH format.

Possible values are: `vir_z00_z39_usmarc` (used when Z58-IN-RECORD-TYPE is USMARC, UNIMARC or DANMARC); `vir_z00_z39_mab` (used when Z58-IN-RECORD-TYPE is MAB); `vir_z00_z39_sutrs_<target>` (used when Z58-IN-RECORD-TYPE is SUTRS).

This setting is mandatory.

- Z58-IN-RECORD-CREATE-PARAM <param>

<param> is a parameter used by the Z58-IN-RECORD-CREATE program.

If the Z58-IN-RECORD-CREATE program is `vir_z00_z39_sutrs_<target>`, the parameter is the conversion table's name. The table is used for converting incoming SUTRS records into standard ALEPH format. The conversion table must reside in `./<external library code>/tab` (for example, `./ext01/tab`)

- Z58-IN-RECORD-CHAR-CONV <char-conv>

ALEPH converts all records to Unicode (UTF-8). This parameter defines the character conversion routine to be used for the records from the target database.

The following are possible values of <char-conv>:

- ❑ ANSEL_TO_UTF
- ❑ MAB_TO_UTF
- ❑ 8859_1_TO_UTF and so on.

All of the possible values are defined in
\$alephe_unicode/tab_character_conversion_line

This setting must be used if the records encoding differs from UTF-8.

-Z58-IN-RECORD-FIX <fix routine>

where <fix routine> defines the fix routine called for the incoming records. <fix routine> must be defined in tab_fix of the EXT library to which the target is connected (for example, EXT01).

This setting is optional.

-Z58-IN-DIRECT-KEY <key>

<key> - defines how subfield \$c of SID tag of the record is filled.

Possible value: TAG.

This setting is optional

-Z58-IN-DIRECT-KEY-PARAM <param>

<param> - parameter for Z58-IN-DIRECT-KEY.

Possible values are: 001, 035, 092, 016 and 544.

Defines from which tag of the incoming record the value for subfield \$c of SID tag is taken.

This setting must be defined if Z58-IN-DIRECT-KEY is defined.

3.1.3 Scan Request Settings

-Z58-OUT-SCAN-CHAR-CONV <char-conv>

This parameter defines the character conversion routine applied in a scan query. ALEPH formulates queries in UTF-8 which must be converted to the character set required by the target database.

The following are possible values of <char-conv>:

- ❑ UTF_TO_ANSEL
- ❑ UTF_TO_MAB
- ❑ UTF_TO_8859_1 and so on.

All the possible values are defined in:
\$alephe_unicode/tab_character_conversion_line.

This setting must be used if the encoding used by the target database for the scan queries differs from UTF-8.

- Z58-IN-SCAN-CHAR-CONV <char-conv>

This parameter defines which character conversion routine is used for conversion of the received Headings (scan results) to Unicode(UTF-8).

The following are possible values of <char-conv>:

ANSEL_TO_UTF
MAB_TO_UTF
8859_1_TO_UTF and so on.

All the possible values are defined in:

`$salephe_unicode/tab_character_conversion_line`

This setting must be used if the encoding of the incoming scan headings differs from UTF-8.

3.1.4 Sort Request settings

- Z58-SORT <Y/N>

This parameter defines whether or not a library supports the sorting of result sets.

If 'Y' is entered:

- in the Search function: the user can sort result sets.
- in the Web OPAC: the column headers in the Result List screen are hyperlinked, enabling the user to click on them and sort the Result List.

If 'N' is entered:

- in the Search function: the user cannot sort result sets (an error message is shown)
- in the Web OPAC: the column headers in the Result List screen are not hyperlinked, so the user cannot sort the Result List.

Default: N

3.1.5 Holdings Information Settings

To add Item Remote tag to Full record display - add ITM6 to `edit_doc_999.<lng>` of the corresponding EXT library.

Web OPAC:

- Z58-HOLDING-METHOD <method>

<method> defines how to obtain Holdings information from the remote base.

Possible values:

OPAC – to retrieve OPAC records

BB1 – to retrieve Holdings Schema records, ESN B1, according to Bath Profile 2.0

BB2 - to retrieve Holdings Schema records, ESN B2, according to Bath Profile 2.0

BC2- to retrieve Holdings Schema records, ESN C2, according to Bath Profile 2.0

When the user clicks on the Item Remote link, Holding Info is obtained from the library using the defined method (Z39.50 Present Request for OPAC record or for

Holdings Schema record is sent to the target and the values from the received record are displayed)

Note:

If Z58-HOLDING-METHOD is not defined, Item Remote tag is not added to Full record screen; if Z58-HOLDING-METHOD is defined, Z58-BASE-NAME must be defined (it is used in the tag content).

SEARCH FUNCTION

When user clicks on Item Remote tag, the **Items** button appears on the right. When user clicks **Items**, holdings information is obtained from the library using the OPAC method (Z39.50 Present Request for OPAC record is sent to the target and the values from the received OPAC record are displayed)

3.1.6 ES Update Settings

Z58-ES-INSERT <program>

<program> - A program which converts the local record into the form required by the Remote Catalog, inserts it into the Remote Catalog and updates the local record by the Remote Catalog information.

Possible values:

es_insert_pica – used for the PICA Union Catalog

es_insert_aleph – used to update the ALEPH Catalog via the Z39.50 protocol

es_insert_keris - used for upload new/updated documents from local BIB to KERIS

Z58-ES-REPLACE <program>

<program> - A program which converts the local record into the form required by the Remote Catalog, uses it to replace the current record in the Remote Catalog and updates the local record by the Remote Catalog information.

Possible values:

es_replace_pica – used for the PICA Union Catalog

es_replace_aleph – used to update the ALEPH Catalog via the Z39.50 protocol

es_replace_keris - used for upload repaired documents to KERIS

Z58-ES-DELETE <program>

<program> - A program which deletes the current record in the Remote Catalog and deletes the Remote Catalog information from the local record.

Possible values:

es_delete_pica – used for the PICA Union Catalog

es_delete_aleph – used to update the ALEPH Catalog via the Z39.50 protocol

es_delete_keris - used for upload deleted documents from local BIB to KERIS

Z58-ES-DOWNLOAD-ACTION <program>

<program> - defines a program that is activated when a record is downloaded from the target.

Possible values:

es_download_keris – used for KERIS

3.1.7 Circulation Status Settings

- Z58-CIRC-PROGRAM <program>

<program> is a program which is involved when the **Circ Status** link is pressed in a Full Record view in a Web OPAC Union Catalog.

The program takes a circulation status from the campus, which is the target and displays it in the Web OPAC interface.

Possible values:

www_f_circ_status_cdl - The program brings circulation status according to "Obtaining the circulation statuses for Melvyl Catalog items" algorithm.

www_f_union_circ_nsls - The program brings circulation status according to the NSLS algorithm.

3.2 Transformations

The ALEPH search/scan query has the form ALEPH Search/Scan code = term. As a different target may have different search/scan syntax, you may want to perform some transformation on the term according to the target's needs. The transformation is defined per CCL code + action (search/scan). Example of transformation: "insert AND between words in the search term". If you search for WTI=(medieval history):

- If no transformation is defined for search using WTI in the Universal Gateway configuration, this search query is passed as is to z39_gate. z39_gate translates it to Z39.50 language (BIB-1 attributes) and sends to the target.
- If a transformation is defined for search using WTI (for example, "insert AND between words in the search term") this search query is transformed by the Universal Gateway into WTI=(medieval AND history) and this new query is sent to z39_gate.

To define a transformation for a CCL code, you must define several lines (Transformation group) in the Universal Gateway configuration file.

3.2.1 Transformation Group Syntax.

The Transformation group consists of the following lines:

```
Z58-TAB01
Z58-TAB01-TYPE          <type>
Z58-TAB01-CODE         <ALEPH code>
Z58-TAB01-CODE-TARGET <Target code>
Z58-TAB01-FILING      <transformation routines>
Z58-TAB01-SUB-CODE    <subfields>
```

The lines must be written in the given order.

Z58-TAB01

Defines the start of the transformation group.

This line must start each transformation group.

Z58-TAB01-TYPE <type>

Defines the type of action for which the transformation is performed.

Possible values of <type>: find/scan/sfind

(This enables you to define a separate transformation for Search using TIT and for Browse using TIT).

This line must be defined in each transformation group.

Example: Z58-TAB01-TYPE find

Z58-TAB01-CODE <ALEPH code>

Defines the ALEPH CCL code for which the transformation is performed.

Possible values of <ALEPH code>: any ALEPH CCL code (for example, WTI, TIT).

This line must be defined in each transformation group.

Example: Z58-TAB01-CODE WTI

Z58-TAB01-CODE-TARGET <Target code>

After the search/scan query (for example WTI=history) is passed to z39_gate, the ALEPH CCL code (for example, WTI) is translated to Z39.50 attributes. This is done according to the CCL to Z39.50 translation line in the z39_gate configuration (for example, find **WTI** u=4 t=1,r,b c=1).

The <target code> must be equal to the CCL code used in Z39.50 translation line.

In most cases, it is also equal to <ALEPH code>. This line must be defined in each transformation group. When sfind is used as the Z58-TAB01-TYPE, the Z58-TAB01-CODE-TARGET must include a different code.

Example: Z58-TAB01-CODE-TARGET WTI

Z58-TAB01-FILING <transformation routines>

This line defines the sequence of the transformation routines to be performed sequentially on the find/scan term when the query is of the type defined by Z58-TAB01-TYPE and the ALEPH CCL code is as defined in Z58-TAB01-CODE.

<transformation routines> contains one or several transformation routines.

- ❑ Each transformation routine has a number.
- ❑ Transformation routine can have parameters. If a transformation routine has parameters, the transformation routine number is followed by 'P', which is followed by parameters, which are followed by 'P'.
- ❑ Transformations routines are separated by '_'. See the Term_Transformations document for a description of the available transformation routines.

Example: Z58-TAB01-FILING 7P([P_2P,.;P_3

This sequence contains three transformation routines: **7**, **2** and **3**.

- Transformation routine **7** has parameters. It removes part of the term, starting from a special character. In this case, the part of the term starting from '(' or '[' is deleted.
- Transformation routine **2** has parameters. It replaces special characters with blanks. In this case, all occurrences of ',', '.', and ';' in the term are replaced by blanks.

- Transformation **3** has no parameters. It inserts **AND** between words in the term. Accordingly, if the initial term is “**Tolstoy, Leo. (1828–1910)**”, it is transformed to “**Tolstoy AND Leo**”.

Z58-TAB01-SUB-CODE <subfields>

The term can contain subfield signs (for example, \$\$a)

<subfield> defines from which subfields of the term the information is taken.

By default the information is taken from all the subfields.

Example: Z58-TAB01-SUB-CODE ab

3.2.2 Examples

Example 1:

```
Z58-TAB01
Z58-TAB01-TYPE      find
Z58-TAB01-CODE      WTI
Z58-TAB01-CODE-TARGET  WTI
Z58-TAB01-FILING    3
```

If the user searches for Title equal to “Great Expectations”, then:

- The ALEPH CCL code WTI is transformed to the Target Code WTI
- The term “Great Expectations” is transformed to “Great AND Expectations” by transformation routine #3. Therefore, the target searches for “Great AND Expectations” in its WTI index.

Note:

The translation of Target_Code to Z39.50 attributes must be defined in the z39_gate configuration of this target. Therefore,

```
$alephe_tab/z39_gate/z39_gate_<TARGET>.conf
```

must contain

```
find WTI <z39 attributes>
```

Example 2:

```
Z58-TAB01
Z58-TAB01-TYPE      scan
Z58-TAB01-CODE      AUT
Z58-TAB01-CODE-TARGET  AUT
Z58-TAB01-FILING    7P([P
```

If the user browses for AUT equal to “Tolstoy, Leo (1828-1910)”, then:

- ALEPH CCL code AUT is transformed to Target Code AUT
- The term “Tolstoy, Leo (1828-1910)” is transformed to “Tolstoy, Leo” by transformation routine 7. Therefore the target scans for “Tolstoy, Leo” in its AUT index.

Note

The translation of target_code to Z39.50 attributes must be defined in the z39_gate configuration of this target. Therefore,

```
$alephe_tab/z39_gate/z39_gate_<TARGET>.conf
```

must contain

```
scan AUT <z39 attributes>
```

Example 3:

```
Z58-TAB01
Z58-TAB01-TYPE      sfind
Z58-TAB01-CODE      TIT
Z58-TAB01-CODE-TARGET SWTI
Z58-TAB01-FILING    28P"P
```

If the user scans titles and then clicks on a headings entry to perform search for “Art and History”, then:

- ❑ The ALEPH CCL code TIT is transformed to the Target Code SWTI
- ❑ The term “Art and History” is transformed to (art "and" history) by transformation routine #28. Therefore, the target searches for “art "and" history” in its WTI index.

Note:

The translation of Target_Code to Z39.50 attributes must be defined in the z39_gate configuration of this target. Therefore,

```
$alephe_tab/z39_gate/z39_gate_<TARGET>.conf
```

must contain

```
find SWTI <z39 attributes>
```

3.3 SUTRS Configuration

The Z39.50 gate program (z39_gate) can accept records in SUTRS format. SUTRS records are converted into ALEPH internal format and displayed to users as regular ALEPH records.

For each target returning SUTRS records, a special conversion program must be written. The conversion program must be named vir_z00_z39_sutrs_<target>.cbl (for example, vir_z00_z39_sutrs_copac.cbl).

Each target must have a conversion table named tab_sutrs_<target> (for example, tab_sutrs_copac). The conversion tables must reside in ./<external library code>/tab (for example ./ext01/tab). For each conversion table, a corresponding header must be defined.

All the conversion tables have four columns:

- Col.1 - SUTRS field name (in upper case)
- Col.2 - ALEPH field code
- Col.3 - ALEPH subfield code (not relevant for ALEPH fields 001-009, for all other ALEPH fields, the default is 'a')
- Col.4 - separator character (if separator is ';' it means that the SUTRS field contains several sub elements separated by ';'. From each sub element separate Aleph field is created)

Example of `tab_sutrs_<target>` table

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!-!-!
ISBN/ISSN                020##
MAIN AUTHOR              100##
TITLE DETAILS           245##
PUBLISHER                260##
PHYSICAL DESC.         300##
SERIES                  440##
NOTES                   500##
LANGUAGE                546##
SUBJECT(S)              650##
HOLDING LIBRARIES      852## ;
```

In the table, the "ISBN/ISSN" element is converted to the ALEPH 020 field; the "MAIN AUTHOR" element is converted to the ALEPH 100 field and so on.

The "HOLDING LIBRARIES" element can contain several sub-elements separated by ';'. From each "HOLDING LIBRARIES" sub-element, a separate ALEPH 852 field is created.

To accept records in SUTRS the following configuration must be performed:

The `./alephe/tab/z39_gate/z39_gate_<target>.conf` file must contain the line "recordtype SUTRS"

The `./alephe/gate/<target>.conf` file must contain:

```
Z58-IN-RECORD-TYPE      SUTRS
Z58-IN-RECORD-CREATE    <conversion program>
Z58-IN-RECORD-CREATE-PARAM <conversion table>
```

For example the `./alephe/gate/copac.conf` file contains:

```
Z58-IN-RECORD-TYPE      SUTRS
Z58-IN-RECORD-CREATE    vir_z00_z39_sutrs_copac
Z58-IN-RECORD-CREATE-PARAM tab_sutrs_copac
```

Example of Conversion of SUTRS record from COPAC Union Catalog (UK):

Main Author: Cohen, Albert, 1929-
 Title Details: Music in the Paris Academy of Sciences, 1666-1793 : a source archive in photocopy at Stanford University : an index / compiled by Albert Cohen and Leta E. Miller
 Series: Detroit studies in music bibliography ; no. 43
 Publisher: Detroit : Information Coordinators, 1979
 Physical Desc.: 69 p ; 24 cm
 ISBN/ISSN: 0911772960

Subject(s): Music - France - 17th century - History and criticism
Music - France - 18th century - History and criticism
Archives - France - Paris - Inventories, calendars, etc.
Archives - California - Stanford - Inventories, calendars,
etc.
Music - History and criticism - Sources, Catalogs
Music - Bibliography - Catalogs
Académie des sciences (France) - Archives, Catalogs
Stanford University. Music Library, Catalogs
Paris Academy of Sciences

Document Type: Bibliography included

Language: English

Holding Libraries: Aberdeen - Queen Mother Library ; Ref 780 Det 43,
Reference only
Cambridge - contact Cambridge University Library :
M420.c.02.35
Nottingham - Music Library ; Reference ML120.F7 CON
Oxford - contact the Bodleian Library
Southampton - Hartley Library ; Reference - Z/ML 270
ULL - University of London Library ; 016.78 [Paris,
Académie]

The conversion is done using `vir_z00_z39_sutrs_copac` and `tab_sutrs_copac`

100 |a Cohen, Albert, 1929-

245 |a Music in the Paris Academy of Sciences, 1666-1793 : a source archive in photocopy at Stanford University : an index / compiled by Albert Cohen and Leta E. Miller

260 |a Detroit : Information Coordinators, 1979

300 |a 69 p ; 24 cm

440 |a Detroit studies in music bibliography ; no. 43

546 |a English

650 |a Music - France - 17th century - History and criticism

650 |a Music - France - 18th century - History and criticism

650 |a Archives - France - Paris - Inventories, calendars, etc.

650 |a Archives - California - Stanford - Inventories, calendars,

650 |a etc.

650 |a Music - History and criticism - Sources, Catalogs

650 |a Music - Bibliography - Catalogs

650 |a Académie des sciences (France) - Archives, Catalogs

650 |a Stanford University. Music Library, Catalogs

650 |a Paris Academy of Sciences

852 |a Aberdeen - Queen Mother Library ; Ref 780 Det 43, Reference only
Cambridge - contact Cambridge University Library : M420.c.02.35
Nottingham - Music Library ; Reference ML120.F7 CON
Oxford - contact the Bodleian Library
Southampton - Hartley Library ; Reference - Z/ML 270
ULL - University of London Library ; 016.78 [Paris, Académie]

4 Example of a Target Configuration

Assume that you are defining a configuration for Test Library:

- ❑ The target name is TEST.
- ❑ The test library is accessed via Z39.50.
- ❑ The target has records in USMARC format
- ❑ The character encoding used by this library is ISO 8859_1
- ❑ This library supports the sort of result sets and OPAC record syntax.

The following files must be defined:

1. \$alephe_tab/test.conf (see below)
2. \$alephe_tab/z39_gate/z39_gate_TEST.conf. (see the *z39_gate* document)

5 Test Library Configuration

#General Settings

Z58-BASE-NAME TEST
Z58-BASE-REMOTE TEST
Z58-ACCESS-METHOD Z39

#Find Request Settings

Z58-OUT-FIND-CHAR-CONV UTF_TO_8859_1

#Response Recode Settings

Z58-IN-RECORD-TYPE USMARC
Z58-IN-RECORD-CREATE vir_z00_z39_usmarc
Z58-IN-RECORD-CHAR-CONV 8859_1_TO_UTF

#Scan Request Settings

Z58-OUT-SCAN-CHAR-CONV UTF_TO_8859_1
Z58-IN-SCAN-CHAR-CONV 8859_1_TO_UTF

#Sort Request Settings

Z58-SORT Y

#Holdings Info Settings

Z58-HOLDING-METHOD OPAC

#Find Transformations

Z58-TAB01
Z58-TAB01-TYPE find
Z58-TAB01-CODE WTI
Z58-TAB01-CODE-TARGET WTI
Z58-TAB01-FILING 7P([/ <P_2P. , ; ; ! " - P
Z58-TAB01
Z58-TAB01-TYPE find
Z58-TAB01-CODE WAU
Z58-TAB01-CODE-TARGET WAU
Z58-TAB01-FILING 7P([/ <P_3
Z58-TAB01
Z58-TAB01-TYPE find
Z58-TAB01-CODE TIT
Z58-TAB01-CODE-TARGET TIT
Z58-TAB01-FILING 7P([/ <P_2P. , ; ; ! " - P_8P7P

Scan Transformations

Z58-TAB01
Z58-TAB01-TYPE scan
Z58-TAB01-CODE AUT
Z58-TAB01-CODE-TARGET AUT
Z58-TAB01-FILING 7P([/ <P_2P.P
Z58-TAB01
Z58-TAB01-TYPE scan
Z58-TAB01-CODE TIT
Z58-TAB01-CODE-TARGET TIT
Z58-TAB01-FILING 7P([/ <P_2P. , ; ; ! " - P

6 Term Transformation Routines

Term Transformation Routines are programs written to optimize search/scan terms sent by the Universal Gateway to remote libraries (see the Universal_Gateway document).

Transformation Routines have the following characteristics:

- ❑ Each Transformation Routine has a unique number.
- ❑ Each Transformation Routine performs an action with the term.
- ❑ A Transformation Routine can have parameters.
- ❑ The Universal Gateway can perform several Transformation Routines sequentially on a term.
- ❑ The Universal Gateway configuration specifies the sequence of Transformation Routines as follows:

<Routine>_<Routine>_<Routine>...

- ❑ If <Routine> has no parameters, then <Routine> is equal to the <Routine number>
- ❑ If <Routine> has parameters, then <Routine> is equal to:
<Routine number>P<parameters>P

Example:

1_7P([P_3

(1 – Transformation routine without parameters

7 – Transformation routine with parameters

3 – Transformation routine without parameters)

Note As an underscore is used to separate Routines in the Universal Gateway, configuration, if you need to use an underscore as part of a Routine parameter, it must be preceded by a backslash.

Example: 7P([_P_3

The following is a list of available Transformation Routines:

Transformation number:1

Purpose: To delete subfield signs from the term

Parameters: None

Example: Using 1 changes “\$\$ahistory” to “history”

Note: this transformation is not in use - the subfield signs are removed automatically

Transformation number:2

Purpose: To replace the characters listed in the Parameters with spaces

Parameters: String of characters

Example: Using 2P-P changes “History-Ancient” to “History Ancient”

Transformation number:3

Purpose: To insert the Boolean operator AND between words in the term

Parameters: None

Example: Using **3** changes “**early medieaval history**” to “**early AND medieaval AND history**”

Note: AND is not inserted if one word is a Boolean AND, OR or ANDNOT

Transformation number:4

Purpose: To delete a Boolean AND, OR or ANDNOT from the term

Parameters: None

Example: Using **4** changes “**coal and mines**” to “**coal mines**”

Transformation number:5

Purpose: To delete short words from the term

Parameters: A number indicating the maximum length of the word to be deleted

Example: Using **5P3P** changes “**Over the rainbow**” to “**Over rainbow**”

Note: If the term contains a single word, it is not removed

Transformation number:6

Purpose: To change the case of the term:

if the Parameter is equal to U, the term is changed to uppercase

if the Parameter is equal to L, the term is changed to lowercase

Parameters: character: U or L

Example: Using **6PUP** changes “**snow**” to “**SNOW**”

Using **6PLP** changes “**SNOW**” to “**snow**”

Transformation number:7

Purpose: To remove part of the term starting from any special character specified in Parameters.

Parameters: String of characters

Example: Using **7P[P** changes “**Hemingway, Ernest [1899-1961]**” to “**Hemingway, Ernest**”

Note: If the term starts with a special character, it is left unchanged

Transformation number:8

Purpose: To shorten the term by limiting the number of words

Parameters: Number of words to leave

Example: Using **8P3P** limits the length of the string to 3 words, so “**The Platonic tradition in Anglo-Saxon philosophy**” is changed to “**The Platonic tradition**”

Transformation number:9

Purpose: If the term contains more than one word, leave only the first letter of the last word

Parameters: None

Example: Using **9** changes “**Hemingway, Ernest**” to “**Hemingway, E**”

Note: Useful for databases which list authors' names as LASTNAME INITIAL only. If the term contains boolean words (“and”, “or”, “andnot”), it is left unchanged.

Transformation number:10

Purpose: To add a truncation symbol ('?') to the beginning or end of the term

R stands for Right and adds a question mark to the right side of the term.

L stands for Left and adds a question mark to the left side of the term

Parameters: Character: **R** or **L**

Example: Using **10PRP** changes “**Elliot T**” to “**Elliot T?**”

Using **10PLP** changes “**Elliot T**” to “**?Elliot T**”

Transformation number:11

Purpose: To delete words preceded by the pattern in the Parameters

Parameters: String

Example: Using **11P<<P** changes “<<**The**>>**History of Art**” to “**History of Art**”

Note: If the term consists of a single word, it is left unchanged

Transformation number:12

Purpose: To insert a comma after the first word of the term

Parameters: None

Example: Using **12** changes “**Jackson Michael**” to “**Jackson, Michael**”

Note: Useful for authors names. If the search string consists of a single word, or already contains a comma, it is left unchanged.

Transformation number:13

Purpose: To delete words appearing between patterns specified in the Parameters

Parameters: start pattern**a**end pattern (**a** separates start pattern from end pattern)

Example: Using **13P<<a>>P** changes “**Linder R <<van den>>**” to “**Linden R**” (word appearing between “<<” and “>>” are deleted)

Note: If the term starts from the start pattern and ends with the end pattern, it is left unchanged.

Transformation number:14

Purpose: Delete words equal to the string specified in Parameters from the term.

Parameters: String of characters

Example: Using **14PxxP** changes “**art xx literature**” to “**art literature**”

Note: If the search string consists of a single word, it is left unchanged.

Transformation number:15

Purpose: For each word in the term – if the word ends up with a string pattern specified in the Parameters – remove this pattern from the word.

Parameters: String of characters

Example: Using **15PxxP** changes “**modernxx artxx history**” to “**modern art history**”

Transformation number:16

Purpose: To append a string pattern to the term.
 Parameters: String of characters
 Example: Using **16P?P** changes “**Smith, G**” to “**Smith, G?**”
 Note: If the term consists of a single word, it is left unchanged

Transformation number:17

Purpose: Useful for transforming Authors into a special form needed by some targets.

If the number of words in the term is not equal to 2, then the term is left unchanged.
 If the length of the second word is greater than 2, then this transformation leaves only the first letter of the second word and appends a question mark to it.

Connects the words by hyphen.

Parameters: None
 Example: Using **17** changes “**Friedman, Milton**” to “**Friedman-M?**”
 Using **17** changes “**Elliot TS**” to “**Elliot-TS**”

Transformation number:18

Purpose: To append a string to each word in the term, except for Boolean operators.

Parameters: String
 Example: Using **18P[TITL]P** changes “**war and peace**” to “**war[TITL] and peace[TITL]**”

Transformation number:19

Purpose: To append a string to the end of the term,
 Parameters: String
 Example: Using **19P[AUTH]P** changes “**Smith, John**” to “**Smith, John[AUTH]**”

Transformation number:20

Purpose: To remove stopwords from the term. The stopwords are specified in tab03 or in tab03.<number> of the EXT library to which the target is connected (for example, EXT01)

Parameters: Parameter may be empty or may contain number.
 If the parameter is empty the stopwords are taken from tab03.
 If the parameter is a number the stopwords are taken from tab03.number.

Example 1: Parameter is empty. If tab03 contains as a word “the”, using **20** changes “**the art**” to “**art**”.

Example 2: Parameter is 4. If tab03.4 contains a word “the”, using **20** changes “**the art**” to “**art**”.

Transformation number:21

Purpose: To limit length of the term to given number of characters
 Parameters: Number
 Example: Using **21P20P** changes “**Advanced readings in history**” to “**Advanced readings**”

Transformation number:22

Purpose: Delete given number of characters from given position in the term
Parameters: <Number1>_<Number2> (<Number1> defines position, <Number2> defines number of characters to delete)
Example: Using **22P4_3P** deletes 3 characters starting from character #4 in the term: “**\$\$\$axxxhistory AND art**” is changed to “**\$\$\$ahistory AND art**”

Transformation number:23

Purpose: If the string contains single word append given string to this word
Parameters: String
Example: Using **23P-?P** changes “**history**” to “**history-?**”

Transformation number:24

Purpose: To add given string to the beginning of the term
Parameters: String
Example: Using **24P[AUTH]P** changes “**Smith, John**” to “**[AUTH]Smith, John**”

Transformation number:25

Purpose: To concatenate all the words from the term using given string separator.
Parameters: String
Example: Using **25P-P** changes “**Smith John**” to “**Smith-John**”

Transformation number:26

Purpose: To replace given string by given pattern. The replacement can occur in the beginning of the term only; in the end of the term only; in the whole term.
Parameters: <string1>a<string2>a<number>
<string1> is the string to be replaced
<string2> is the replacement pattern
<number> is the replacement indication:
1 – in the beginning of the term only
2 - in the end of the term only
3 – in the whole term
Example 1: Using **26P?a*a1P** replaces “?” by “*” in the beginning of the term: “**?history?**” is changed to “***history?**”
Example 2: Using **26P?a*a2P** replaces “?” by “*” in the end of the term: “**?history?**” is changed to “**?history***”
Example 3: Using **26P?a*a3P** replaces “?” by “*” in the whole term: “**?history?**” is changed to “***history***”

Transformation number:27

Purpose: To delete given string from the term. The string can be deleted in the beginning of the term only; in the end of the term only; in the whole term.

Parameters: <string>a<number>
<string> is the string to be deleted
<number> is the indication where to delete the string:
1 – in the beginning of the term only
2 - in the end of the term only
3 – in the whole term

Example 1: Using **27P?a1P** deletes “?” from the beginning of the term: “**?history?**” is changed to “**history?**”

Example 2: Using **27P?a2P** deletes “?” from the end of the term: “**?history?**” is changed to “**?history**”

Example 3: Using **27P?a3P** deletes “?” from the whole term: “**?history?**” is changed to “**history**”

Transformation number:28

Purpose: Mask Boolean operators by surrounding logical words in the term

Parameters: Usually a quote sign, or a double quote sign

Example: Using **28P”P** changes "art and history" to (art "and" history)

Note: Z58-TAB01-CODE-TARGET must be "Word" code; otherwise the result will have double quotes inside double quotes, which is not correct.

Transformation number:29

Purpose: Deletes initial articles if they are "stop" words.

Parameters: The parameter can be empty or can contain a number from 1 till 999.If the parameter is empty ./<ext library>/tab/tab03 is used as the "stop" words table. If the parameter contains a number, for example 5, then the “stop” words will come from the ./ext01/tab/tab03.5 table.

Example: Using **29P5P** changes "the history of the art" to “history of the art” if “the” is in the ./ext01/tab/tab03.5 table.