Aleph Configuration for ARC 3.0

Document Version 1.0

CONFIDENTIAL INFORMATION

DISCLAIMER

TRADEMARKS

Document released:June 2, 2014

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

Table of Contents

About This Document

This document describes the configuration of ALEPH that should be completed before the first ETL (Extract, Transform, and Load) process can be executed. This configuration is done in order to define the connections between the ALEPH and ARC systems.

This document also describes the ETL initiation script, the setting up of this process, its monitoring, and testing.

The last section of the document describes the possibility of connecting to ARC via the ALEPH GUI client and the configuration it requires.

1 ALEPH Configuration for ARC

ALEPH setup must be carried out before running the first ETL process. The setup includes the mandatory and optional steps described below.

1.1 Defining prof_library

A connection between ALEPH and the ARC databases is required in order to perform the ETL process. In order to set up this connection, the ARC-related section in the ALEPH configuration table (`prof_library`) must be configured. The following are examples of such sections and explanations of how each parameter should be set:

```
# Reporting Center Definitions
#       setenv ARC_TWO_TASK <Server>.<ARC Oracle_SID>
        setenv ARC_ORACLE_SID <ARC Oracle_SID>
        setenv ARC_ADMIN <ADMIN_USER>
        set arc_user = "<ARC Database Name>"
        setenv patron_additional_id_1 "nn"
        setenv patron_additional_id_2 "nn"
```

Of the first two lines (`ARC_TWO_TASK & ARC_ORACLE_SID`) only one line can be active.

Here is an example of a correctly defined ARC setting in `prof_library`:

Reporting Center Definitions

```
#       setenv ARC_TWO_TASK ram19.aleph5
        setenv ARC_ORACLE_SID aleph5
        setenv ARC_ADMIN ARC_ADMIN
        set arc_user = "usmr0"

        setenv patron_additional_id_1 "02"
```

```
setenv patron_additional_id_2 "03"
```

1.1.1 ARC_ TWO_TASK variable setup

Use the ARC_TWO_TASK variable in one of the following two cases:

- When ALEPH and ARC are located on different machines.
- When ALEPH and ARC use different Oracle DBMS versions. For example, the ALEPH installation uses the Oracle 11.2 database and the ARC installation uses the Oracle 11.3 database. (ARC_TWO_TASK variable should be used in this case even though both ALEPH and ARC are installed on the same machine).

The parameter that must be specified under the ARC_TWO_TASK variable is

```
ARC database connection string (TNS NAME).
```

ARC TNS name should be defined in the ALEPH side tnsnames.ora file under $ORACLE_HOME/network/admin.

For more details about ARC_TWO_TASK variable configuration, see Appendix A on page 23.

As mentioned above, when the ARC_TWO_TASK variable is in use, the ARC_ORACLE_SID variable should be commented out.

1.1.2 ARC_ORACLE_SID variable setup

When ALEPH and ARC products are installed on the same machine and in the same Oracle database version, the ARC_ORACLE_SID variable should be used.

The ARC_ORACLE_SID parameter should be set to the ARC DB SID.

When using ARC_ORACLE_SID, comment out the ARC_TWO_TASK variable.

1.1.3 ARC_ADMIN variable setup

The ARC_ADMIN variable must contain the user of the Oracle administrator of the ARC database. The default username is ARC_ADMIN.

```
setenv ARC_ADMIN ARC_ADMIN
```

If for any reason during the ARC installation process another user name was created, it should be specified under the ARC_ADMIN variable. In this case, the customer is notified by the ARC installation team.

If the ARC_ADMIN user password has been changed by the customer, then the following changes should be performed:

- The Oracle password file should be updated accordingly on the ALEPH side. In order to perform the update, the following command should be executed on the ALEPH server:

put_ora_passwd $ARC_ADMIN <new ARC_ADMIN user password>

Note that without executing this command following the password change, the ETL process fails.

- The ARC server side should be updated in the same way:

  put_ora_passwd $ARC_ADMIN <new ARC_ADMIN user password>

1.1.4 arc_user variable setup

arc_user must contain the ARC schema name. The ARC schema is created for the ADM library during the ARC installation process. The naming convention of ARC schema creation is to substitute '5' in the ADM library name with 'R' in the ARC schema name. For example, if the ADM library name is ABC50, the ARC schema name is ABCR0.

1.1.5 patron_additional_id variable set up

patron_additional_id_1 and patron_additional_id_2 are used for the global patron IDs. The variable that can be put in is taken from Z308-KEY-TYPE.

1.1.6 p_arc_01_loop_length

setenv p_arc_01_loop_length 50000

Since bib records, items and events records are extracted in parallel (parallel within a table), the ETL cycles file is constructed so that each related cycle holds a range of records. The number of records to be processed in each cycle is taken from this variable. The default value is 50000 records in each cycle.

1.1.7 p_arc_01_time_loop_length

setenv p_arc_01_time_loop_length 002

OPAC events are also extracted in parallel. Here the length of a cycle corresponds to a period of time .The arc_01_time_loop_length parameter sets the amount of time in weeks. For example, if set to 2, each cycle extracts OPAC events for two weeks. The default value is 2 weeks.

The same variable is used for regular events (z35) cycles in ALEPH 16 and up.
In Aleph 21 and up, this variable is used also for Circulation Logger (Z309) cycles.

1.1.8 p_arc_01_bor_loop_length

setenv p_arc_01_bor_loop_length 3000

Patron Local Info (Z305) table is also extracted in parallel. Each cycle handles and loads a defined range of patron records. The number of patron records to be processed in each cycle is taken from this variable. The default value is 3000 patrons in each cycle.

1.2 License to Perform ETL

It is required to have the following line in the Aleph license file:

```
ARC-ETL                         Y
```

Without this line it is not possible to run the ETL.

1.3 Prof_library configuration tests

After configuration of the above ARC-related setups, perform the following connection test.

1.3.1 p_arc_01_test script

Under `./aleph/proc` directory resides `p_arc_01_test`. You can use this script to test the connection.

```
dlib <ADM library name>
csh -f $aleph_proc/p_arc_01_test
```

The following is an example of the results of the test:

```
Connection test: PASSED
```

If you have a connection problem, see Appendix B for troubleshooting.

1.4 Configuring pc_tab_exp_field.lng (Mandatory)

The ETL process described above extracts data from the ALEPH Oracle tables of the Administrative Library and other relevant libraries and loads data in ARC schema. This process also builds various description tables in ARC schema based on sections of the `pc_tab_exp_field.eng` of the administrative library. For example, an order status description table (`dwh_dim_order_status`) is created by the ETL process based on the values defined in the `ACQ-ORDER-STATUS` section of the `pc_tab_exp_field.eng` configuration table.

The following sections are required by the ETL process and must be present in the `pc_tab_exp_field.eng` table of the customer administrative library. For examples of required values, consult the `pc_tab_exp_field.eng` table of the administrative demo library.

If any of these fields do not exist, then either of the following happens:

- Relevant reports have some fields empty
- Relevant reports present no results at all

```
ACQ_APPROVAL_DEPARTM
ACQ_ARRIVAL_STATUS
ACQ_INVOICE_STATUS
ACQ_INVOICE_TYPE
ACQ_ORDER_MAT_DEL
```

```
ACQ_ORDER_MATERIAL
ACQ_ORDER_METHOD
ACQ_ORDER_STATUS
ACQ_ORDER_TYPE
BUDGET-STATUS
BUDGET-TRANSACTION
BUDGET-TYPE
CASH-STATUS
DEPARTMENT-NAME
DOC-FORMAT
EVENT-CATEGORY
FREQUENCY
HOLD-STATUS
ITEM-LOCATION-TYPE
ITEM-PROCESS-STATUS
LOAN-STATUS
OPAC-EVENT-TYPE
ORDER-INVOICE-STATUS
PRINT-STATUS
RUSH-REQUEST
BUDGET-GROUP-(1-5)
SEND-ACTION
ITEM-DEPOSITORY-ID
ITEM-GAP-INDICATOR
PAYMENT-MODE
RECALL-TYPE
DELIVERY-METHOD
ITEM-STATISTIC
ACQ_ORDER_GROUP
PATRON-LANGUAGE
ACQ_SEND_METHOD
PHOTO-CHARGE
PHOTO-STATUS
CIRC-LOG-ACTION
```

Note that the section entries can be presented and written with underscores as well as with hyphens.

1.5 Call Number Topics

`The tab_topics.lng` table, which resides in the administrative library, should be configured to include call number information of the classification type used by the library.

1.6 Configuring arc_heading (Mandatory)

The `arc_heading` table contains captions for several pre-defined parameters in the ALEPH Reporting Center. The user can assign new captions to these parameters.

To assign new captions, change the third column of the table. The new captions are inserted into the ALEPH Reporting Center during the proximal ETL.

For example, the following lines

```
!!!!-!!!!-!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
1005 0000 L Sunday
1006 0000 L Monday
1007 0000 L Tuesday
1008 0000 L Wednesday
```

can be translated to German to suit a library working in German.

```
!!!!-!!!!-!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
1005 0000 L Sonntag
1006 0000 L Montag
1007 0000 L Dienstag
1008 0000 L Mittwoch
```

The existing entries in the table are:

```
!!!!-!!!!-!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
1000 0000 L UnKnown-
1001 0000 L Morning
1002 0000 L Noon
1003 0000 L Evening
1004 0000 L Night
1005 0000 L Sunday
1006 0000 L Monday
1007 0000 L Tuesday
1008 0000 L Wednesday
1009 0000 L Thursday
1010 0000 L Friday
1011 0000 L Saturday
1012 0000 L Last Week
1013 0000 L Last Month
1014 0000 L Last Quarter
1015 0000 L Last Year
```

For version 16, the `arc_heading` table resides in the `error_eng` directory of the version's original alephe. It should be copied from the version's original alephe directory to your active alephe directory.

For version 17 and 18 the `arc_heading` table resides in the `aleph/error_eng` directory.

1.7 Implementing an Additional Oracle Table (Optional)

If your library intends to produce statistics about Web OPAC usage, you need to implement the configuration steps described below.

In the BIB library, enter util/a/17 to create a new Oracle table (this Oracle table allows the creation of statistical reports about OPAC usage. In ALEPH 500 versions 16 and 17, this Oracle table is called Z69.

For the implementation of this table, refer in V.16.02 to `rep_change 259` and in V.17 to the Z69 document.

1.8 Define Z13

The ETL process reads the ADM record and takes the bibliographic record that is linked to this ADM record. Therefore, in the case of the ILL and Course Reading reports, it reads the Z13 (Short Document Record) of the XXX40 (ILL) and XXX30 (Course Reading) libraries. Creation of the short document is required in these libraries.

1.8.1 Addition of User Defined Fields

The number of the user-defined data fields is now 15.

Additional user defined fields 6-15 can be added to tab22 and they are extracted to ARC. They can be found in the presentation layer under Bib Information as DESC6 to DESC15.

These fields are saved in Z13U Oracle table. Creation of Z13U Oracle table is required in the bibliographic, ILL, and course reading libraries.

Note that the existence of the Z13U table improves ETL performance.

1.9 Adding Library-Defined Fields

It is possible to define up to 70 bibliographic fields that are extracted to ARC.

The Aleph configuration table, `tab22_arc`, is used to define these bibliographic fields that can be found in ARC in the presentation layer under BIB Information as FIELD1 to FIELD70.

Define the fields defined in column 1 as:
ARC-DEF-1 = user defined 1

ARC-DEF-2 = user defined 2
ARC-DEF-3 = user defined 3
ARC-DEF-4 = user defined 4
ARC-DEF-5 = user defined 5
And so on - till ARC-DEF-70.

Each line defines a unique alternative library-defined field.
Note that if similar line of tag + subfield + position + length repeats, the system takes the first of the alternative fields that it finds (group of columns of: tag + subfield + position + length).
When Col.2 is set to 2, the data is taken from the BIB record, using edit_paragraph. In this case, col.3 is used to list the paragraph number, and the remaining columns are left blank.

Structure of the table:
Col. 1. field code;
ARC-DEF-1, ARC-DEF-2, ARC-DEF-3, ARC-DEF-4, ARC-DEF-5 ... until ARC-DEF-70
Col. 2. Function code:
1=data taken from the bib record's tag + subfield + position
2=data taken from the bib, using edit_paragraph;
Col. 3. Field tag + indicators or Paragraph number;
If col.2 is set to 1, enter Field Tag.
If col.2 is set to 2, enter Paragraph number from the edit_paragraph table.
Col. 4. Subfield;
Col. 5. Starting position:
Defines the position from which to take data from a fixed field. For example, if column 1 is YEAR 1, the year might be taken from a fixed field. In this case, define the position in the fixed field from which to begin by taking 4 positions, counting from base 01. If the fixed field has a subfield code, add 3 to the starting position in order to take it into account, for example, 0008 to define the 8th position of the 008 field in MARC21, 0013 to define the 9th position of the 100 field in UNIMARC.
Col. 6. Length: Number of bytes from Starting position. Defines the number of bytes taken from the starting position.

The following is an example of the table:

```
!   1        2   3    4  5    6
!!!!!!!!!!-!-!!!!!-!-!!!!-!!!!
ARC-DEF-1   1 LDR      0006 0001
ARC-DEF-2   1 LDR      0007 0001
ARC-DEF-3   1 FMT
ARC-DEF-4   1 STA    a
ARC-DEF-5   1 007      0000 0001
ARC-DEF-6   1 007      0001 0001
ARC-DEF-7   1 007      0004 0001
ARC-DEF-8   1 008      0006 0001
ARC-DEF-9   1 008      0007 0003
ARC-DEF-10  1 008      0015 0002
ARC-DEF-11  1 008      0021 0001
ARC-DEF-12  1 008      0022 0001
ARC-DEF-13  1 008      0028 0001
```

```
ARC-DEF-14 1 008      0035 0002
ARC-DEF-15 1 020    a
ARC-DEF-16 1 020    z
ARC-DEF-17 1 020    c
ARC-DEF-18 1 022    a
ARC-DEF-19 1 041    a
ARC-DEF-20 1 084    2
ARC-DEF-21 1 084    a
ARC-DEF-22 1 090    a
ARC-DEF-23 1 100    a
ARC-DEF-24 1 110    a
ARC-DEF-25 1 111    a
ARC-DEF-26 1 245    a
ARC-DEF-27 1 245    b
ARC-DEF-28 1 245    c
ARC-DEF-29 1 245    p
ARC-DEF-30 1 245    n
ARC-DEF-31 1 246    a
ARC-DEF-32 1 250    a
ARC-DEF-33 1 260    a
ARC-DEF-34 1 260    b
ARC-DEF-35 1 260    c
ARC-DEF-36 1 300    a
ARC-DEF-37 1 300    b
ARC-DEF-38 1 300    c
ARC-DEF-39 1 310    a
ARC-DEF-40 1 563    a
ARC-DEF-41 1 700##  a
ARC-DEF-42 1 700##  d
ARC-DEF-43 1 700##  e
ARC-DEF-44 1 700##    0000 9999
ARC-DEF-45 1 740##  a
ARC-DEF-46 1 740##  h
ARC-DEF-47 1 740##    0000 9999
ARC-DEF-48 1 710##  a
ARC-DEF-49 1 711##  a
ARC-DEF-50 1 949##  #
ARC-DEF-51 1 5####  a
ARC-DEF-52 1 5####  b
ARC-DEF-53 1 5####  c
ARC-DEF-54 1 5####  d
ARC-DEF-55 1 5####  e
ARC-DEF-56 1 5####    0000 9999
ARC-DEF-57 1 970    a
ARC-DEF-58 1 970    b
ARC-DEF-59 1 971    a
ARC-DEF-60 1 972    a
ARC-DEF-61 1 972    b
ARC-DEF-62 1 830      0000 9999
ARC-DEF-63 1 TYP    a
ARC-DEF-64
ARC-DEF-65
ARC-DEF-66
ARC-DEF-67
ARC-DEF-68
ARC-DEF-69
ARC-DEF-70
```

1.9.1 Adding expanded fields as library-defined fields

It is possible to define expanded bibliographic fields, to be extracted from Aleph to ARC. This is done in tab_expand, by using the "CREATE-ARC" entry in Col. 1 of tab_expand, and defining expand procedures (Col. 2) and arguments (Col. 3).
Then define the expanded field in tab22_arc

For example:
The expand program "expand_doc_bib_avail" brings items and holdings availability information and presents it in an AVA field.
Defined it for ARC in tab_expand as follows:

```
!   1                     2                                3
!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
CREATE-ARC expand_doc_bib_avail           THRESHOLD=020;AVA=NE,RE;
```

Define the following in tab22_arc:

```
!   1       2   3   4  5    6
!!!!!!!!!!-!-!!!!!-!-!!!!-!!!!

ARC-DEF-61 1 AVA   a
ARC-DEF-62 1 AVA   b
ARC-DEF-63 1 AVA   c
ARC-DEF-64 1 AVA   d
ARC-DEF-65 1 AVA   e
ARC-DEF-66 1 AVA   f
ARC-DEF-67 1 AVA   g
ARC-DEF-68 1 AVA   h
ARC-DEF-69 1 AVA   i
```

1.10 Define arc_conf_table.conf

This file is used for defining constrains on some of the data that is extracted through the ETL process.

Syntax

[ARC-HISTORY]
from year = 2007

[OPAC]
from date = 20070101

[EVENTS]
from date = 20070101

[PATRON-DATA]
eliminate_data=N

[CIRC-LOGGER]

from date = 20070101

Location
The file resides under the relevant ADM tables directory such as `usm50/tab/`

Explanation

[ARC-HISTORY]
As a part of the ETL, historical data is retrieved for requests (Z37H), loans (Z36H), and items (Z30). This configuration file allows you to restrict the history period that is loaded.

Historical data is retrieved <from_year> .

For example, if the file states "from year = 1999", all historical records since 1999 (inclusive) are loaded.

If there is no file in the ADM tables directory, all historical data are loaded (as before).

[OPAC]
This definition allows you to restrict the period from which OPAC events are extracted.
All OPAC events beginning from the "from date" until the current date are extracted. The format of the date is: yyyymmdd.

If no entry exists in the configuration file, all OPAC events are extracted.

If the given date is later than the current date, no records are extracted.

[EVENTS]

Regular events (z35) are handled the same as OPAC events.

[PATRON-DATA]

It is possible to eliminate patron identification information from the ARC Portal.
When the line does not exist in `arc_conf_table.conf` or when there is `eliminate_data=N`, a regular ETL is executed. All fields are extracted as before.

When the line is `eliminate_data=Y`, limited ETL in terms of patron's identification is carried out. With this option, it is not be possible to identify a patron. Its IDs are scrambled and most of its data is null.

Note that even when the information is scrambled or null, the links related to the patron are retained.

[CIRC-LOGGER]

This definition allows you to restrict the period from which circulation logger transactions are extracted.

All circulation logger transactions beginning from the “from date” until the current date are extracted.

The format of the date is: yyyymmdd.

If no entry exists in the configuration file, all circulation logger transactions are extracted.

If the given date is later than the current date, no records are extracted.

Note:
The [OPAC], [EVENTS] and [CIRC-LOGGER] definitions are used in Full ETL extract mode (F).

2 Run ETL Process

Another prerequisite of the ARC implementation is the uploading of records from your administrative library into the ARC schema. The batch service that needs to be `run is p_arc_01`, and it is located in the `$aleph_proc` directory.

The parameters are as follows:

- Library name
- Table name or ALL for all tables
- Lock library (Y/N)
- Number of processes (up to 99)
- ETL mode – incremental, complete incremental, or full (I/C/F).
  The default is I.
  The "incremental" extract mode (I) takes only z35 and z69 records which were not yet extracted to ARC as they were created after the last ETL run.

  The Full extract mode (F) takes the same records that are handled by the "Complete Incremental" extract mode (C).

  The Full extract mode (F) takes all the source data from Aleph and replaces all the data in ARC. This mode is the heaviest one, so it is recommended to use it for a first run of ETL, as an initial load of Aleph information to ARC.

  The Complete Incremental extract mode (C) takes only records that have been added, changed, or deleted since last run of ETL. Only new records are added. Changed or deleted records in Aleph are merged with existing ones in ARC.

  The Complete Incremental mode (C) is the recommended one for ongoing use on a regular basis. This can be done by defining the job_list. (For more information, see ETL Configuration in the job_list File on page 20).

Note:
In order to activate the Complete Incremental extract mode, the Aleph Z900 Oracle table, index, and related triggers must be defined in the local file_list file of the ADM, BIB, USR, ILL and Course Reading libraries. Multi ADM environments also need to run the util – x – 11 Clean ARC trigger table .
For more information, see **Appendix C. Aleph Z900 Oracle Table, Index and Related Triggers** on page 25.

The `tab_arc_timestamp`:
Each time the ETL is run and completed successfully, the time of the last extraction is saved in the `tab_arc_timestamp` file, located in the $data_root/tab/ directory of the ADM library.

This time is used for the next complete incremental run.

The `EXTRACT_START_DATE` from `prc_process` table of the last successful and complete ETL run (with 'ALL') is taken. All event records that were created from this date forward are processed in the current ETL.

Note that only an "incremental" run for ALL tables (with 'ALL' in the second parameter) is supported.

Also note that it is recommended to run ETL for the first time with full mode (F).

From the second time and on the ETL can run with Complete Incremental (C) or incremental (I) mode.

- Multi-ADM library (Y/N).
  The default for the multi-ADM parameter value is "N", which means that the ETL runs in single ADM mode. When running in single-ADM mode ("N"), the Z900 (ARC trigger) Oracle table of the related USR BIB libraries is cleaned at the end of the ETL process.

  When running in Multi-ADM mode ("Y"), the Z900 is not cleaned by each ETL run of each ADM. This is so that updates and deletions of records in one ADM do not override updates in other ADM libraries.

  Note:
  In Multi-ADM mode, the Z900 Oracle table needs to be cleaned only after the ETL is run on all ADM libraries.

  For more details, see Appendix D. Z900 Cleanup in Multi-ADM Environment on page 26.

Example of Running ETL:

In USM50 for all tables, it is run as follows (from within the $aleph_proc directory):

```
csh -f p_arc_01 USM50,ALL,Y,4,C,N
```

ETL Log Files:

The logs of the run are located in the library `scratch` directory.

`p_arc_01.cycles` shows the extract progress.

`p_arc_01.log` contains the following:

- Details on the entire ETL process
- Information as to whether or not each table has been created successfully in the extract stage

Errors from the transform and load are shown in `p_arc_01_transform.log`.

Ex Libris usually puts the main log in a report folder under the data root directory. In order to do so, add the following to your command (in one line):

```
csh -f p_arc_01 USM50,ALL,Y,4,I,N & >
$data_root/report/p_arc_01.log.20111218_1113 &
```

2.1 Run ETL in Debug Mode

Running ETL in Debug mode means extracting a single table or ALL tables where the sequential files created by the EXTRACT phase are not erased and the TRANSFORM phase is not applied. The sequential files remain in the ADM `$data_files` directory (usm50/files). To set the `DEBUG_MODE`, do the following from the window from which the ETL is executed:

```
setenv ARC_DEBUG Y
```

2.1.1 Run ETL for a single table

When executing ETL on a single table – for debug purposes only – all the parameters noted above remain the same except for the second parameter (ALL) which now includes a specific table name. For example:

```
cd $aleph_proc
csh -f  p_arc_01 USM50,BIB-INFO,N,3
```

Note that for most tables, one process is enough, but if the tables are processed in parallel, more than one process can be applied. The tables that are processed in parallel are: `BIB-INFO, ITEMS, EVENT, OPAC-EVENT, CIRC-LOGGER, LOCAL-PATRON`.

An ETL run on only one table is only made in full mode.

The table names are as follows:

```
EVENT-CATEGORY
OPAC-EVENT-TYPE
ACQ-ORDER-METHOD
ACQ-APPROVAL-DEPARTM
DEPARTMENT-NAME
ACQ-ARRIVAL-STATUS
ACQ-ORDER-STATUS
```

ACQ-INVOICE-STATUS

ACQ-PAYMENT-STATUS

ACQ-INVOICE-TYPE

VAT-CODE

ACQ-ORDER-MATERIAL

ACQ-ORDER-TYPE

BUDGET-STATUS

BUDGET-TYPE

BUDGET-TRANSACTION

BUDGET

ACQ-ORDER-MAT-DEL

ORDER-INVOICE-STATUS

ITEM-MATERIAL-TYPE

ITEM-STATUS

ITEM-PROCESS-STATUS

ITEM-COLLECTION

ITEM-LOCATION-TYPE

SUB-LIBRARY

ORDERING-UNIT

BIB-LIBRARY

BASE

ORDERS

CURRENCY

INVOICE

CASH

BUDGET-TRANS

EVENT

OPAC-EVENT

SERIAL-CLAIM

LOCAL-PATRON

GLOBAL-PATRON

VENDOR

FREQUENCY

CASH-STATUS

CASH-TYPE

REQUEST

LOANS

SUBSCRIPTION

COURSE

EVENT-TYPE

```
LOAN-STATUS
REQUEST-STATUS
PRINT-STATUS
RUSH-REQUEST
BOR-STATUS
BOR-TYPE
SEND-ACTION
CATALOG
ITEMS
BIB-INFO
CAPTION
CALL-NO-TOPICS
BUDGET-GROUP
USER-ADDRESS-TYPE
ITEM-DEPOSITORY-ID
ITEM-GAP-INDICATOR
PAYMENT-MODE
RECALL-TYPE
DELIVERY-METHOD
ITEM-STATISTIC
ACQ-ORDER-GROUP
PHOTO-CHARGE
CIRC-LOG-ACTION
CIRC-LOGGER
PATRON-LANGUAGE
ACQ-SEND-METHOD
```

3 ETL Configuration in the job_list File

ETL, like other processes, can be set to run on a regular basis from the `job_list`. To set it, you need to add corresponding records to the two tables: `job_list` and `job_list.conf`. Both tables can be accessed via `cd $alephe_tab; ls` or via the utils menu.

In order for the changes made to be active, the job daemon needs to be restarted.

To restart the job daemon:

1 Stop the daemon: UTIL E/15/2

2 Start the daemon: UTIL E/15/1

3 Check that job daemon is running: UTIL E/15/3

The following is an example of the settings of both tables for two cases: full ETL and partial ETL.

Suppose you want your ETL to run once a week on a Sunday morning at 07:00 and the parameters for the run are:

ap;csh -f p_arc_01 ABC50,ALL,N,3,**C**,N

In this case the setting should be the following (add appropriate lines to the relevant tables):

1) Job_list.conf

W8 W YNNNNNN - run on Sunday

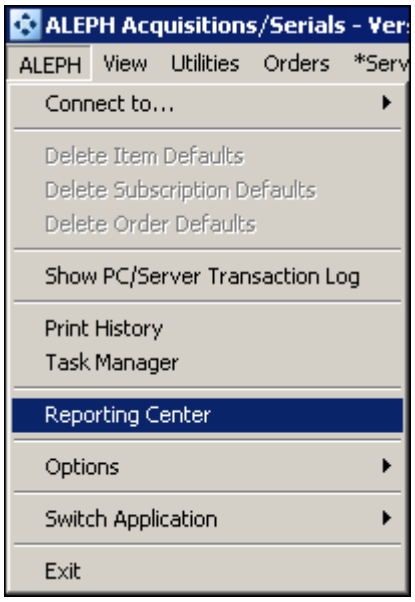
2) job_list

!
! Running ETL once a week (Sunday) for ALL tables of ABC50 library.
!
W8 07:00:00 Y ABC50 p_arc_01 ABC50,ALL,N,03,**C**,N

Note: Multi ADM environments should run the Clean ARC trigger table (Z900) utility after running ETL for all ADM libraries, and last parameter "Y". This can be set to run on a regular basis from `job_list`.
See
Appendix D. Z900 Cleanup in a Multi-ADM Environmenton page 26.

4 Setting Up Access to ARC from ALEPH GUIs

Staff users can access the ARC via the ALEPH menu in the Acquisitions/Serials, Cataloging, and Circulation GUIs:



To enable access to ARC from GUIs:

1 There are two types of ALEPH modules: BIB modules and ADM-based modules.

- For ADM-based modules (Acquisitions and Circulation), tab100 in all the relevant ADM libraries should be updated.
- For the only BIB-based module (Cataloging), tab 100 under the relevant BIB libraries should be modified.
- For the ILL-based module (ILL), tab 100 under the relevant ILL libraries should be modified.

Enter the URL of the ARC in the relevant `tab100` table in the ARC-URL parameter. For example:

```
ARC-URL=http://ram19:8080/cognos/
```

2 In alephcom.ini, define the Web browser to be used in the [Main] section as in the following example:

```
Browser=C:\Program Files\Internet Explorer\IEXPLORE.EXE
```

Appendix A. Configuration of Oracle connection (TNS name) on ALEPH server (file tnsnames.ora)

Note that the steps specified below are valid only when the ARC database is installed by Ex Libris using Ex Libris standard installation procedures. If the ARC database was created by the customer, it is the customer's responsibility to verify that the Oracle listener is configured on the ARC side and that `tnsnames.ora` file on the ALEPH side Oracle database is properly defined.

To configure the Oracle connection:

1 Log in to Oracle software as the Oracle software user (for the Oracle version which is used for the ALEPH version being configured):

```
su – oracle
cd $ORACLE_HOME/network/admin
```

2 Copy from the server where ARC is installed, the following file

```
/exlibris/app/oracle/product/1122/network/admin/tnsnames.ora.crn.

(On the ARC server it is under the directory:
$ORACLE_HOME/network/admin)
```

3 Add the contents of the `tnsnames.ora.crn` file at the end of the `tnsnames.ora` file:

```
cat tnsnames.ora.crn >> tnsnames.ora

In this example the name of the ARC database is "crn".
This is suffix of the filename "tnsnames.ora.crn". In some cases
the DB name could be different.
```

These settings are only valid when the ARC Database was created using the Ex-Libris scripts. If the customer has created the DB in any other method, then the DBA should configure the ALEPH side Oracle listener manually, according to the database settings.

Appendix B. Troubleshooting after p_arc_01_test

If the p_arc_01_test script run and was not successful, it may be due to a TNS error (connection problem) as in the following example:

```
ALEPH/LINUX, Copyright Ex Libris.
version 21 revision 01 copy 1, 18-May-2011

*********
Now checking license permission
There is a valid ARC license
Now testing Oracle connection with ARC DB
Testing TWO_TASK connection to il-arc01.arc4 :
ORA-12154: TNS:could not resolve the connect identifier specified
ORA-12154: TNS:could not resolve the connect identifier specified
Could not connect to ARC_ADMIN (ARC_ADMIN)
****    Aborting    ****
```


```
ALEPH/LINUX, Copyright Ex Libris.
version 21 revision 01 copy 1, 18-May-2011

*********
Now checking license permission
There is a valid ARC license
Now testing Oracle connection with ARC DB
Testing TWO_TASK connection to il-arc01.arc4 :
ORA-12154: TNS:could not resolve the connect identifier specified
ORA-12154: TNS:could not resolve the connect identifier specified
Could not connect to ARC_ADMIN (ARC_ADMIN)
****    Aborting    ****
```


Perform the following checks to correct the connection problem:

1. Check if tnsnames.ora (cd $ORACLE_HOME/network/admin) on the Aleph server contains the ARC server configuration. If it does not, add it and perform the p_arc_01_test.

2. If you are still getting the TNS error, check if the ARC server is configured in the etc/hosts. If it is not, configure it. After configuration, run the p_arc_01_test again.

3. If you are still getting the TNS error, check if port 1521 is open from Aleph to ARC.

Appendix C. Aleph Z900 Oracle Table, Index and Related Triggers

In order to activate the Complete Incremental extract mode, define the Aleph Z900 Oracle table, index, and related triggers in the local file_list file of the ADM, BIB, USR, ILL and Course Reading libraries.

The Z900 Oracle table contains data about records deleted in the Aleph library.

After configuring the Z900 Oracle table, index and related triggers in the file_list run util A – 17 – 1 (Drop & Create Table and Index) and util A – 17 – 6 – 2 (Enable Trigger).

The following are example DEMO definitions, placed in the flie_list files under $aleph_root/tab/.

To implement this, make the definitions in the local libraries file_list file, under $data_root directory:

- file_list – BIB Library

```
!1        2                  3                4                5                6
!!!-!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!
TAB z900               1M               0K               ts0              ram36.aleph1
IND z900_id            100K             0K               ts1              ram36.aleph1
IND z900_id1           100K             0K               ts1              ram36.aleph1
```

- file_list – ADM Library

```
!1        2                  3                4                5                6
!!!-!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!
TAB z900               1M               0K               ts0              ram36.aleph1
IND z900_id            100K             0K               ts1              ram36.aleph1
IND z900_id1           100K             0K               ts1              ram36.aleph1
SEQ last result set
TRI z76 trigger 1
TRI z601_trigger_1
TRI z31_trigger_1
TRI z77_trigger_1
TRI z30 trigger 1
TRI z30h trigger 1
TRI z36 trigger 1
TRI z36h_trigger_1
TRI z37_trigger_1
TRI z20_trigger_1
TRI z68 trigger 1
TRI z16 trigger 1
TRI z501 trigger 1
TRI z70_trigger_1
TRI z08_trigger_1
TRI z38_trigger_1
TRI z309 trigger 1
TRI z78_trigger_1
```

- file_list – Course Reading Library

```
!1        2                  3                4                5                6
!!!-!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!
TAB z900               1M               0K               ts0              ram36.aleph1
```

```
IND z900_id           100K            0K              ts1             ram36.aleph1
IND z900_id1          100K            0K              ts1             ram36.aleph1
SEQ last temp file
SEQ last media number
TRI z127_trigger
TRI z127_trigger_1
TRI z127 trigger 2
TRI z13_trigger_1
```

- file_list – ILL Library

```
!1        2                 3               4               5               6
!!!-!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!
TAB z900              1M              0K              ts0             ram36.aleph1
IND z900_id           100K            0K              ts1             ram36.aleph1
IND z900_id1          100K            0K              ts1             ram36.aleph1
TRI z13_trigger_1
```

- file_list – USR Library:

```
!1        2                 3               4               5               6
!!!-!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!
TAB z900              1M              0K              ts0             ram36.aleph1
IND z900_id           100K            0K              ts1             ram36.aleph1
IND z900_id1          100K            0K              ts1             ram36.aleph1
SEQ last_publish_seq
TRI z303 trigger 1
TRI z304 trigger 1
TRI z305_trigger_1
TRI z308_trigger_1
```

Appendix D. Z900 Cleanup in a Multi-ADM Environment

Overview

In a multi ADM environment, after running ETL for all ADM libraries, run the util/x/11 “Clean ARC trigger table” utility to clean the Z900 Oracle table on the USR and BIB library connected to the ADM libraries.

This utility deletes from Z900 the records that were already handled by the ETL according to the Number of Days to Retain parameter, supplied by the customer.

Perform the cleaning according to the earliest run time of the ETL.

Note:
This utility is relevant only for multi-ADM customers, and should be run in the USR and BIB libraries. In a single ADM, the Z900 (ARC trigger) Oracle table is already cleaned by the ETL process.

Z900 Cleanup Configuration in job_list

Like the ETL process, the Z900 clean on the USR and BIB libraries can be set to run on a regular basis from `job_list`. This is done by adding the `clear_arc` job. The job has two parametres in `job_list` – Number of Days to Retain and Loop Length.

For example:

```
01 10:00:00 Y                    USR00 clear arc                    USR00 2 2000
01 10:00:00 Y                    ABC01 clear_arc                    ABC01 2 2000
```

In this example, 2 is the number of days to retain the data and 2000 is the loop length or the number of records to be processed in each cycle.

- The “Number of Days to Retain” parameter:

  The default value of “Number of Days to Retain” is zero, meaning clean all of the Z900 (ARC trigger) Oracle table.

  The value of this parameter should depend on how frequent the customer runs the ETL process in the different ADM libraries, and the number of days that passed from the run of ETL in the first ADM library - to the run of "clear_arc" (or util X 11) job (ETL ends in all ADM).

  Example:

  An ETL process can set to run on USM50 on Sunday, on USM51 on Monday and on USM52 on Tuesday.

  The "clear_arc" job in the job list – can be set to run on USR and on BIB libs also in Tuesday, after the ETL process ended in the last ADM (USM52).

  In such a case the "number of days back to retain" should be set to 3.

- The “Loop Length” parameter:

  The default “Loop Length” value is 1000.

This parameter defines the number of Z900 records' deletion will be commited to the database.

A change of the default value should be done only in case of performance problems, and should be increased only if the customer has many Z900 records (deletion actions) that are done between each run of the ETL process.