# Table of Contents

1 SYSTEM ARCHITECTURE OVERVIEW ....................................................... 4
   1.1 ALEPH Server Architecture ............................................................... 5
      1.1.1 Presentation Services & Logic .................................................. 5
      1.1.2 Application Logic ..................................................................... 5
      1.1.3 Data Services & Logic ................................................................ 6
      1.1.4 Other characteristics of ALEPH’s architecture: .......................... 6

2 ALEPH UNIX LOGINS ........................................................................... 7

3 ALEPH DIRECTORY STRUCTURE ...................................................... 8
   3.1 The aleph Root Directory ................................................................. 8
   3.2 The alephe Directory ........................................................................ 9
      3.2.1 The Main Configuration File - aleph_start .................................. 10
   3.3 Library Structure ............................................................................. 12

4 LIBRARY UTILITIES ........................................................................... 13
   4.1 UTIL C - Monitor batch jobs ......................................................... 14
   4.2 UTIL X - Clean Up ......................................................................... 15

5 SERVERS, DAEMONS, BATCH JOBS AND PROBLEM DIAGNOSIS ... 16
   5.1 UTIL W - Server management (Monitor, Stop, Start, Log files) ........ 16
   5.2 Server Logs ..................................................................................... 17
   5.3 Starting Your Own Server for Testing .......................................... 17
   5.4 PC Client / PC Server ...................................................................... 18
   5.5 Connecting ..................................................................................... 18
   5.6 Analyzing PC Server Problems ...................................................... 19
   5.7 PC Server Configuration (pc_server_defaults) .............................. 21
      5.7.1 Max Response Time .................................................................. 21
      5.7.2 Backend Servers ...................................................................... 21
   5.8 ue daemons .................................................................................... 21
   5.9 Batch jobs ..................................................................................... 21
   5.10 Services ....................................................................................... 22
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.11</td>
<td>Job List</td>
<td>23</td>
</tr>
<tr>
<td>5.12</td>
<td>cron Jobs</td>
<td>23</td>
</tr>
<tr>
<td>5.13</td>
<td>www_server (Public)</td>
<td>23</td>
</tr>
<tr>
<td>5.14</td>
<td>www_server (Staff)</td>
<td>23</td>
</tr>
<tr>
<td>5.15</td>
<td>Z39.50 Servers</td>
<td>24</td>
</tr>
<tr>
<td>5.16</td>
<td>OCLC Servers</td>
<td>24</td>
</tr>
<tr>
<td>5.17</td>
<td>SIP2 Server (3M Standard Interchange Protocol)</td>
<td>24</td>
</tr>
<tr>
<td>5.18</td>
<td>NCIP Server (NISO Circulation Interchange Protocol)</td>
<td>25</td>
</tr>
<tr>
<td>5.19</td>
<td>Interlibrary Loan Server (ILL Server)</td>
<td>25</td>
</tr>
<tr>
<td>5.19.1</td>
<td>Running the ILL Server</td>
<td>25</td>
</tr>
<tr>
<td>5.19.2</td>
<td>The ILLSV Library</td>
<td>25</td>
</tr>
</tbody>
</table>

6 DATA WAREHOUSE

27
1 System Architecture Overview

ALEPH is a library services agent, providing application services to clients via its APIs (Application Program Interfaces). ALEPH’s architecture is based on a multitier, client/server model. Client/Server communication is based on a stateless (self-contained) transaction model, nonetheless, ALEPH Application Servers keep continuous connections (with time-out) to the database, to ensure high performance.

ALEPH features a flexible database design. Each ALEPH site is composed of seven interrelated, yet separate, units: Authority, Bibliographic, Holdings, Administrative, Interlibrary Loan, Course Reading and a system-wide administration unit. Each database unit can have many occurrences with many-to-many links to the other database units. ALEPH's database design supports a wide range of database configurations and implementations, including independent installations of different units on different hosts. ALEPH's database design plays a crucial role in the system's scaleability potential.

ALEPH’s architecture is based on a scaleable, distributed logic model and relies on an object-oriented design. Two key features of ALEPH’s architecture are:

Multitier, Client/Server model - ALEPH is split into logical segments with a clearly defined interface based on message passing. See Figure 1 below.

Modularity - The key notion of the distributed logic design that underlies ALEPH is modularity - both vertical (between tiers) and horizontal (within the tiers). This ensures that the system is easy to maintain and extend, and that new technologies and concepts can be quickly integrated.

ALEPH can be installed on the following platforms:

- IBM/AIX
- ALPHA UNIX
- SUN SOLARIS
- LINUX REDHAT
1.1 ALEPH Server Architecture

**Multitier Architecture - OPAC**

![Diagram of ALEPH server architecture](image)

The ALEPH server is composed of the following tiers:

1.1.1 Presentation Services & Logic

Provides the interface with which the user interacts.

1.1.2 Application Logic

**Application Servers tier** - A front-end tier which is composed of dedicated servers for each interface. Each application server receives a query from a source client, translates the query to a uniform format and directs it to the relevant Application Service object (API). Once translated, all queries have the same format, regardless of their origin (Z39.50 client, ALEPH client, WWW browser or standard character mode terminal).

**Application Services tier (APIs)** - The heart of ALEPH is the application services tier, composed of sets of APIs. The APIs provide library services for the different clients. For example, a FIND API provides FIND services to all clients after a FIND query is invoked at one of the clients (WWW, Z39.50, telnet or ALEPH proprietary). As part of its open system architecture, ALEPH includes mechanisms to integrate new APIs as well as to extend the Application Services to other clients or applications. This provides considerable potential for extensibility.
1.1.3 Data Services & Logic

**ALEPH Database Middleware (or I/O Engine)** - This is a high level database management layer. A logical server provides data services to the application services objects. It contains a group of objects which intermediates between the application and the database. The I/O engine translates an application request to a sequence of database commands. In addition, the I/O engine provides SQL enhancement which is required because of the textual, non-formatted nature of library catalog data.

ALEPH's I/O engine also exploits the knowledge that the system has about the DBMS's special characteristics, in order to optimize data updating and retrieval. It is in this logical layer that ALEPH incorporates its experience and know-how of libraries' special data structures and formats.

Having an intermediate level of the I/O Engine between the application and the DBMS ensures maximum flexibility of DBMS logical and physical design.

**ALEPH Database** - The ALEPH database runs under Oracle RDBMS.

1.1.4 Other characteristics of ALEPH’s architecture:

**Network Orientation** - ALEPH's distributed logic is designed to fit into scaleable network configurations. With its Application Servers tier and Database Middleware tier (the I/O engine), ALEPH is suitable for intra-networking and inter-networking. ALEPH not only supports a range of clients access (WWW, Z39.50, ALEPH proprietary and Telnet Clients) but also accommodates access to heterogeneous database resources.

Special features are included to overcome the diversity of databases: MATCH to locate remote/external documents, and Multi-FIND to broadcast a search across multiple databases. In addition, ALEPH includes enhanced functionality to support centralized, de-centralized and union catalogs.

**Scaleability** - ALEPH's multitier, distributed architecture provides a wide range of scaleability possibilities: distribution of data across disks/servers, distribution of services across servers or even a multi-server configuration with shared data.
2 ALEPH Unix Logins

The ALEPH server requires the following Unix users:

<table>
<thead>
<tr>
<th>Login</th>
<th>Home Directory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>m505</td>
<td>../a50_5/alephm</td>
<td>ALEPH administrator</td>
</tr>
<tr>
<td>oracle</td>
<td>$ORACLE_HOME</td>
<td>Oracle administrator (DBA)</td>
</tr>
</tbody>
</table>

The person who is in charge of all the libraries, the System Administrator, can login as m505, and thus will be able to modify the parameters and data of all libraries.

**m505** – The ALEPH administrator (System Administrator and/or System Librarian) uses the m505 account for various online utilities and command line activity. The m505 user has access and control over all the libraries in the system.

**oracle** – The oracle DBA uses the oracle account for DBA activity outside the scope of online utilities **UTIL O - Oracle Management** and **UTIL A - File Administration and Building**.
3 ALEPH Directory Structure

3.1 The aleph Root Directory

An aleph root is composed of three types of directories:

1. ALEPH software
   - alephm, aleph, tmp, log

2. Demo libraries
   - usm01, usm10, usm20, usm30, usm50, usm60
   - uni01, uni10, uni20, uni30, uni50, uni60
   (usm = MARC21, uni=UNIMARC)

3. Node management
   - alephe

As part of the installation phase, the alephe directory (node management) is relocated to a different directory (for example u50_5). This directory will normally contain site-specific libraries as well. A sample scheme is given in Figure 3.
This structure supports the ability to upgrade the ALEPH software (under \texttt{a50\_5}) without interfering with local customization (under \texttt{u50\_5}).

In addition to the core software directories \texttt{aleph} and \texttt{alephm}, the system has directories which include management tables and parameter files pertaining to the specific installation (\texttt{alephe}), and to each of the libraries.

### 3.2 The \texttt{alephe} Directory

The primary directories in the \texttt{alephe} tree are:

- \texttt{tab} node management tables
- \texttt{www\_<type>\_<lng>} HTML files for Web OPAC, Web services, Course Reading and others
- \texttt{pc\_b\_<lng>} GUI services
- \texttt{scratch} intermediate files and logs
- \texttt{apache} \texttt{conf/htdocs/logs}
- \texttt{error\_<lng>} messages that are displayed to the end user
- \texttt{gate} Z39.50 set up
- \texttt{unicode} tables which translate from and to Unicode
3.2.1 The Main Configuration File - aleph_start

aleph_start is ALEPH 500's main configuration file. It contains definitions of ALEPH libraries (abc01, usm01, etc.) and environment variables. It also contains logical assignments (the only place with physical references to ALEPH directories). aleph_start is in the alephe directory (cd $aleph_root).

In order for changes in aleph_start to take effect, you must exit ALEPH, re-login, and then restart the daemons and servers. Running aleph_shutdown stops all daemons and servers. Running aleph_startup restarts daemons and servers depending on definitions in aleph_start and aleph_start.private.

Following are primary portions of aleph_start:

```
setenv ALEPH_VERSION 16
setenv ALEPH_COPY 1
setenv ALEPH_APP_VERSION a${ALEPH_VERSION}_${ALEPH_COPY}
setenv ALEPH_REVISION 02
setenv WWW_SERVER_PORT 499${ALEPH_SUB_PORT}
setenv PC_SERVER_PORT 699${ALEPH_SUB_PORT}
setenv Z39_GATE_PORT 799${ALEPH_SUB_PORT}
setenv HTTPD_PORT 8991
setenv Z39_SERVER_PORT 999${ALEPH_SUB_PORT}
setenv usm01_dev ${ALEPH_MOUNT}/a${ALEPH_VERSION}_$ALEPH_COPY
setenv usm10_dev ${ALEPH_MOUNT}/a${ALEPH_VERSION}_$ALEPH_COPY
setenv usm11_dev ${ALEPH_MOUNT}/a${ALEPH_VERSION}_$ALEPH_COPY
setenv usm12_dev ${ALEPH_MOUNT}/a${ALEPH_VERSION}_$ALEPH_COPY
setenv usm14_dev ${ALEPH_MOUNT}/a${ALEPH_VERSION}_$ALEPH_COPY
setenv usm15_dev ${ALEPH_MOUNT}/a${ALEPH_VERSION}_$ALEPH_COPY
setenv usm19_dev ${ALEPH_MOUNT}/a${ALEPH_VERSION}_$ALEPH_COPY
setenv usm20_dev ${ALEPH_MOUNT}/a${ALEPH_VERSION}_$ALEPH_COPY
setenv usm23_dev ${ALEPH_MOUNT}/a${ALEPH_VERSION}_$ALEPH_COPY
setenv usm30_dev ${ALEPH_MOUNT}/a${ALEPH_VERSION}_$ALEPH_COPY
setenv usm50_dev ${ALEPH_MOUNT}/a${ALEPH_VERSION}_$ALEPH_COPY
setenv usm51_dev ${ALEPH_MOUNT}/a${ALEPH_VERSION}_$ALEPH_COPY
setenv usm60_dev ${ALEPH_MOUNT}/a${ALEPH_VERSION}_$ALEPH_COPY
setenv usm90_dev ${ALEPH_MOUNT}/a${ALEPH_VERSION}_$ALEPH_COPY
setenv pw_library USR00
setenv z105_library USR00
setenv usr_library USR00
```
setenv QUE_STARTUP_LIBS "usm01 usm50"

setenv WORD_STARTUP_LIBS "usm01 usm50"
setenv REQUEST_STARTUP_LIBS "usm50"
setenv ACC_AUT_STARTUP_LIBS "usm01 usm50"
setenv MESSAGE_STARTUP_LIBS "usm01 usm50"

setenv RLIN_STARTUP_LIBS "usm01"
setenv SLNP_STARTUP_LIBS ""
setenv SIP2_STARTUP_LIBS "usm50"

setenv Z39_SERVER_STARTUP Y
setenv Z39_GATE_STARTUP Y
setenv OCLC_SERVER_STARTUP Y
setenv NCIP_SERVER_STARTUP Y

setenv ORACLE_SID aleph2
setenv ORACLE_OWNER oracle
setenv ORACLE_VERSION 92
setenv ORACLE_CONF ${ALEPH_MOUNT}/ora_aleph
setenv NLS_LANG American_America.UTF8
setenv ORACLE_HOME /aleph/app/oracle/product/920

setenv aleph_db $(ORA_HOST).$(ORACLE_SID)

setenv aleph_dev ${ALEPH_MOUNT}/a${ALEPH_VERSION}_${ALEPH_COPY}
setenv alephm_dev ${ALEPH_MOUNT}/a${ALEPH_VERSION}_${ALEPH_COPY}
# setenv alephe_dev ${USER_MOUNT}/u${ALEPH_VERSION}_${ALEPH_COPY}
setenv alephe_dev /exlibris/aleph/a16_1
setenv alephe_synch_dev ${ALEPH_MOUNT}/a${ALEPH_VERSION}_${ALEPH_COPY}
setenv aleph_utf /tmp

setenv aleph_product ${ALEPH_MOUNT}/a${ALEPH_VERSION}_${ALEPH_COPY}/product
setenv TMPDIR ${ALEPH_MOUNT}/a${ALEPH_VERSION}_${ALEPH_COPY}/tmp
setenv LOGDIR ${ALEPH_MOUNT}/a${ALEPH_VERSION}_${ALEPH_COPY}/log

setenv apache_version apache_1.3.27
setenv apache_dir ${aleph_product}/local/apache
setenv httpd_bin ${aleph_product}/local/apache/bin
setenv httpd_root ${aleph_dev}/alephe/apache

# Use aleph_start.private to define customer libraries
if (-f ${aleph_dev}/alephe/aleph_start.private) then
  source ${aleph_dev}/alephe/aleph_start.private
endif

Note: Only lines that are above the following line can be configured:

### End of version and/or platform dependent setting ###

Everything under this line is hardcoded.
3.3 Library Structure

Every ALEPH library (xxx01, xxx50, xxx60, etc.) has a separate root directory. The term “library” in ALEPH relates to three things:

- The physical library building with books and other materials
- An Oracle database user with tables of data
- A UNIX directory structure

Here is an example of a standard UNIX directory structure. Under the library root directory there are files and directories.

The primary files are:

- `file_list` Configuration file for Oracle tables, indexes and objects
- `prof_library` Library logical assignments

The primary directories are:

- `files/` Export files (XXX01.exportSEQ.tar.z)
  Dump files (znn.seqaa)
  Batch queue management
  (alias df1)
- `tab/` Parameter and configuration tables (alias dt).
- `scratch/` Scratch directory. Used for intermediate and log files (alias ds).
- `pc_tab/` Parameter and configuration tables.
- `print/` Print files (alias dp).

The tab and pc_tab directories and library parameter and configuration tables are described in the Database Management Guide, available on the Ex Libris Documentation Center under ALEPH 500 Documentation/16.
4 Library Utilities

Each library’s database is implemented within Oracle as a separate Oracle user. Each Oracle user owns a set of tables which contains the library’s data.

Each library has access to the ALEPH utilities by activating the `util` command from the prompt. The UTIL main screen will appear:

```
Library Utilities
=================
A. File Administration and Building
C. Monitor Batch Jobs
D. Online Store/Restore Administration
E. Monitor Background Jobs
F. View Procedures and Files
G. Tables for Defining Database Structure
H. Library and Installation Report (Files, Tables and Definitions)
I. Formatting Data (PC, WWW, Reports)
J. Web and Server Configuration
K. ILL Tables
L. GUI Tables
M. GUI CATALOGING Tables
N. Z39.50 Management
O. ORACLE Management
P. Unicode Tables
Q. Data Loading, Import and Export Tables
R. Multimedia
S. Statistics
U. Data Warehouse
W. Server Management (Monitor, Stop, Start, Log Files)
X. Clean Up
Y. Node Management
Please select [exit]:
```

Note: For UTIL functions that are not included in this document see the Database Management Guide, available on the Ex Libris Documentation Center under ALEPH 500 Documentation/16.
4.1 UTIL C - Monitor batch jobs

C. Monitor Batch Jobs

------------------

0. Exit Procedure

1. Check Library Running Process
2. Start Library Batch Queue
3. Stop Library Batch Queue
4. Check Library Lock Status
5. Lock Library
6. Unlock Library
7. Show Queue of Jobs Awaiting Execution
8. Delete Entry from Queue of Jobs Awaiting Execution
9. Alter Run Time of Job Awaiting Execution
10. Display Logfile of a Batch Job
11. Display Log List of Batch Jobs

Please select [exit]:

For more information refer to the Database Management Guide, available on the Ex Libris Documentation Center under ALEPH 500 Documentation/16.
4.2 UTIL X - Clean Up

These utilities delete scratch files from various directories on the Unix server.

Note: before running the utilities you might need to close ALEPH servers, batch queues and daemons.
5 Servers, Daemons, Batch Jobs and Problem Diagnosis

There are various ALEPH servers and daemons:

- www_server
- PC server
- Z39.50 servers
- OCLC server
- Self-check server (SIP2)
- ue_daemons
- Batch jobs
- NCIP Server
- ILL Server

Conventional servers’ port numbers are:

- WWW server (Web)   4991
- PC server (GUI)   6991
- Z39.50 server    9991
- Z39.50 gate    7991
- Generic server (OCLC server)  5771
- Self-check server (SIP2)  5331
- NCIP     5991
- ILL Server    5551
- Apache     80

5.1 UTIL W - Server management (Monitor, Stop, Start, Log files)

The server management utility shows you which servers are running and has dialogues for starting and stopping the servers.

```
W. Server Management (WWW,PC,Z39)
--------------------------------
0. Exit Procedure
1. Monitor Servers
2. Stop Servers
3. Start Servers
4. View Log File

Please select [exit]:
```

Configuration of the www_server is discussed in the UTIL J/6 Definition of Web Server Defaults (www_server.conf) section of the Database Management Guide, available on the Ex Libris Documentation Center under ALEPH 500 Documentation/16.
5.2 Server Logs

The logs of the various servers are written to the $LOGDIR directory.

The log names are prefixed with the server type. For example pc_ser_<port>.

The log files contain statistics and any other input from the servers. They are useful for debugging and analyzing.

When a new server is executed the old log files are renamed with a date/time extension, for example:

   Oct 20 12:10 www_server_4991.log.2010.1210

Apache server logs are written to ../alephe/apache/logs.

5.3 Starting Your Own Server for Testing

When testing or analyzing reproducible problems, it is frequently best to start your own instance of the server rather than use an existing one. (This way the log entries for your transactions are not mixed in with those for other transactions.)

For example:

Web

The syntax for starting your own Web server is:

   www_server <server-port> <apache-port> <num-servers> [stdout]

For example:

   www_server 4123 80 1 stdout

stdout specifies that you want the server output (log) to be displayed on your display terminal.

To use your own Web server enter the URL

   http://<URL>:<server-port>/<type>

In our example, to use the Web OPAC enter

   http://<URL>:4123/F

GUI

The syntax for starting your own GUI server is:

   pc_server <port number> stdout
For example:

```
pc_server 6123 stdout
```

`stdout` specifies that you want the server output (log) to be displayed on your display terminal.

Specify **6123** as the address in your PC’s ..\Alephcom\Tab\library.ini file, instead of **6991**.

### 5.4 PC Client / PC Server

Configuration of the `pc_server` is performed using **UTIL J/5 Definition of Defaults for PC Server Defaults** (`pc_server_defaults`).

When a problem occurs in the GUI it can be with either the client or the server (the PC server).

Error messages may have been generated by the PC client or they may have been sent from the server. Certain kinds of messages clearly indicate a problem on the PC side. These include: GDI failure; Dr. Watson's; and General Protection Faults (GPF). With such problems, make sure that the GUI client is at the same service pack level as the server you are trying to connect to and make sure that the PC has sufficient free memory and disk space.

Other error messages such as Remote file error (global-xx); Remote service error; or Failed to read reply are passed from the server and indicate a problem on the server.

Here is how the PC and the PC-server interact:

#### 5.5 Connecting

When you start the first GUI module on your PC, if you have not saved your logon identification on the client, the software prompts you for a user name and password. It checks all the servers listed in the ..\Alephcom\Tab\library.ini file. If this user name/password isn't valid on any of the servers it can connect to, the message Password not verified on connectable hosts is displayed. This indicates either that this user name/password is not valid; OR that the address specified in the ..\Alephcom\Tab\library.ini is wrong; OR that the pc_server specified in the library.ini entry is down.

Assuming that the password is verified, you then connect to a library. Select **File / Connect** from the main menu. The system displays the libraries listed in the module's `per_lib.ini` file, for example, ..\Circ\Tab\per_lib.ini. When you click on a particular library (or base), the client software goes to the ..\Alephcom\Tab\library.ini file and tries to connect to the IP address specified for this library.
Service Requests
(Note: See the ALEPH Web Guide, available from your Staff Menu for documentation of each GUI module.)

The functions you perform on the PC generate service requests for the PC server. The pc_com/pc_server program processes all incoming requests. It checks the license date, user limit, and so on, and passes control to the program specified in the service request. For example, if the service request is c0511 and the module is ACQ, it passes control to the pc_acq/pc_acq_c0511 program. Just before doing this it writes an entry to the pc_server log:

```
SERVICE: C0511
MODULE: Acquisition Services
DESCRIPTION: Invoice Header Information
ACTION: GET
PROGRAM: pc_acq_c0511
```

The text for the description comes from the ../alephm/proc/pc_service.dat file. This file shows all available services. Each service is self-contained. Though the program may call other non-service programs in the course of its processing, it is the PC which initiates each service call.

The pc_xxx_cnnnn programs call the com/service_error_message routine, passing a specific error number as a parameter, in order to generate text for error messages. The service_error_message reads the ../alephe/error_<lng> file to get the text for this program for the specific error number.

For example, if there was an error 0101 in pc_acq_c0511, the service_error_message would read the ../alephe/error_<lng>/ pc_acq_c0511 file and find this entry:

```
0101 0005 L Invoice Net Amount must be numeric.
```

(An error message such as Error 21 Not defined for service C0204 in pc_cat_c0204.eng indicates that there is no line with text for the 0021 error in the alephe/error_eng/pc_cat_c0204 file when there should be.)

5.6 Analyzing PC Server Problems

There is normally just a single PC server (6991). Multiple logs for this server (for example, pc_ser_6991) in a time period when the server was not stopped intentionally (for example, for a backup) indicate a server failure. To get more information than what you see in the pc_ser_nnnn file type:

```
pc_server view <port number> <number of lines>
```

For example

```
>>pc_server view 6991 25
```
(The fact that the PC server is stateless means that each transaction is self-contained. The transaction can be caught and simulated without a client.) When you get the number of the problem IN transaction, such as 55459 IN, type:

```
pc_server check[x] <port number> [<line number>]
```

For example

```
pc_server check 6991 55459
```

or (if the DATA line you see is truncated and you want to see the rest):

```
pc_server check[x] <port number> [<line number>]
```

For example

```
>>pc_server checkx 6991 55459
```

Note that in cases where the transaction is performing an update, execution of `pc_server check` will cause the update to be performed again.

**Example:**

The system does not have a catalog package for some reason, that is, `$data_root/pc_tab/catalog/pc_cat.pkc` is missing. When connecting to the Cataloging module the error message `Failed to read reply` appears. This means that the server could not handle the abnormal situation. Type:

```
M505>> pc_server view 6991 10
```

and you will see:

```
4078  IN  2987  Mon 12-01-2004 14:42:35  010.001.235.203
010001235203ShirlyM;;default;8846f602-44da-11d8-a7ee-0050dac686e0;ACQ;
C0513  GET  USM50ENGSMYTH  000000
```

This reveals that transaction number 4078 called function C0513 (Acquisition Services).

For more details use:

```
M505>> pc_server check 6991 4078
```

and as part of the output you will see:

```
SERVICE  : C0513
MODULE   : Acquisition Services
DESCRIPTION: Invoice Information
ACTION   : GET
PROGRAM  : pc_acq_c0513
```
Now it is obvious that **UTIL M/7 Update Tables Package** should be activated in order to create the catalog package.

**Note:** Even if you cannot understand the error, it is best to catch the transaction and put it aside for further handling by Ex Libris Support. This can be done with the following sequence of commands:

```
M505>> cd $LOGDIR
M505>> tail -10 pc_ser_6991 > my_transaction_file
```

### 5.7 PC Server Configuration (pc_server_defaults)

#### 5.7.1 Max Response Time

If you have a problem with transactions timing out, increasing the `alephe/pc_server_defaults PC_SERVER_MAX_RESPONSE_TIME` value might help.

The only downside is that problematic transactions (transactions which are looping, etc.) may run even longer. So before changing the `PC_SERVER_MAX_RESPONSE_TIME` value, make certain that the problematic transactions are not due to a lack of the appropriate Oracle indexes: compare your `xxxnn` library's `file_list` to the `usmnn`'s `file_list`.

#### 5.7.2 Backend Servers

The `alephe/pc_server_defaults PC_NUM_SERVER` parameter determines how many backend servers are running. The default value is 5.

### 5.8 ue daemons

Certain utilities can be initiated to execute as daemons, constantly checking to see if work needs to be done. These are:

- `ue_01` indexing daemon
- `ue_03` RLIN loader
- `ue_06` request handling
- `ue_08` update BIB ACC from AUT
- `ue_11` messaging daemon

### 5.9 Batch jobs

**Note:** For detailed information on batch jobs which build indexes, please consult the document, *How To Run Index Jobs* available on the [Ex Libris Documentation Center](https://www.exlibrisgroup.com/en/documentation) under *ALEPH 500 Documentation/How To/System Administration.*

Batch jobs are executed either periodically (daily, weekly, etc.) or on request. They may be initiated via the Services menu in the GUI or the utilities **UTIL E/15 Managing Job Daemon** and **UTIL E/16 Update Daemon Job List.**
Each utility is described in detail in the *UTIL E Monitor Background Jobs* section of the *Database Management Guide*, available on the [Ex Libris Documentation Center](http://www.exlibrisgroup.com) under *ALEPH 500 Documentation/16*.

### 5.10 Services

Each GUI module has a **Services** menu. This option lets you submit batch jobs from a menu, choosing the values you want for each parameter. The submission screens which you see are in the `../alephe/pc_b_<lng>` directory (unless directed elsewhere in *path_convert*) and may be modified. As is the case with the `www_a_<lng>` Web OPAC screens, the system checks first for a suffixed version and then looks for a non-suffixed version.

When you submit the job from a particular window, the system calls the corresponding program to process the service. If errors are found in the parameters, an error message will be displayed. Otherwise the message *Job submitted to queue* will be displayed.

The job will then be placed in the batch queue of the library for which it was submitted. If the library's batch queue is running, then the job will be executed and will show up under *UTIL C/1 Check Library Running Process* as being executed.

The job will appear in *UTIL C/1 Check Library Running Process* with its parameters. In this example the batch service is **cir-51 Overdue and Lost Billing Summary Letter**:

```
aleph 31696 59116 43 18:24:25 pts/12 0:00 csh -f
/exlibris/aleph/a50_5/aleph/proc/p_cir_51
EXU50,ovrdustaf,N,00,00,00,00,00,00,00,00,00,N,00,00,00,00,00,00,0
0,,00,14,00,Y,Y,N,O,4,0,00,
```

The `p_cir_51` component is the procedure which is being executed. The procedures are in the `aleph/proc` directory. `../aleph/proc/p_cir_51` executes three different programs:

```
cobrun b_cir_51_a
cobrun b_cir_51_b
cobrun b_cir_51_e
```

Print templates are stored in `usm01/form_<lng>`. All formats are taken from XSL files.

Any printed output produced by the batch job will be in the library's print directory (for example, `../usm50/print`). The name of the output file is controlled by the **Output File** value on the submission screen. The log of the job will appear in the `../alephe/scratch` directory under the name of the process (in our example, `usm50_p_cir_51.00138`).

For more information on forms, see the document *Customizing Printouts (XML and XSL) - 16.pdf* available on the [Ex Libris Documentation Center](http://www.exlibrisgroup.com) under *ALEPH 500*.
5.11 Job List

Jobs which need to be run on a periodical basis can be placed in the job_list file. Use UTIL E/15 Managing Job Daemon and UTIL E/16 Update Daemon Job List.


5.12 cron Jobs

"cron jobs" are similar in principle to the ALEPH job_list described in the preceding section, except that their control and maintenance are outside of ALEPH.

5.13 www_server (Public)

The WWW server for Web OPAC is accessed via http://<URL>/P. For example: http://ram11:8991/F.

The Web OPAC is an interface for accessing and searching an ALEPH 500 online catalog via the HTTP Internet standard. The Web OPAC allows a patron either to enter the system as a guest user, or to sign in, thereby activating his customized profile.

UTIL J - Web and Server Configuration is used for several www_server configurations. This includes messages to Web OPAC users (www_f_heading); sort types for display in Web OPAC (www_f_sort_heading); and definitions for server defaults (www_server.conf). See the UTIL J - Web and Server Configuration chapter in the Database Management Guide, available on the Ex Libris Documentation Center under ALEPH 500 Documentation/16.

The Web OPAC screens are in the ./alephe/www_f_eng directory. See List of Web OPAC HTML Files for a description of each file. Web OPAC Screens – Structure and Customization explains how to customize the screens. These documents are available on the Ex Libris Documentation Center under ALEPH 500 Documentation/16.

5.14 www_server (Staff)

The WWW server Staff Menu is accessed via the address http://<URL>/S. For example, http://ram11:8991/S.
Guide - Read help on how to use the ALEPH system
OPAC - Online Public Access Catalog
Course Reading Management - Reading lists for courses
Release Notes - Read about changes since the last release
Ex Libris Documentation Center
Utilities - Online utilities

Guide - the ALEPH 500 Web Guide, the first and foremost source of ALEPH documentation.

OPAC – Access to the ALEPH 500 online catalog.

Course Reading Management- the staff Course Reserve function where library staff and faculty create course records and connect documents to them. (The Course Reading screens are in the ../alephe/www_r_<lng> directory, unless directed elsewhere in ../<library>/tab/path_convert.)

Release Notes - changes made from revision to revision, organized according to functionality and tables.

Ex Libris Documentation Center - a link to the Ex Libris Documentation Center

Utilities – Utilities that are used to view/update the Chinese dictionary (the dictionary is used for translation for filing and for word breaking) and to view/update Synonym groups (Synonym groups are optional for word indexing).

5.15 Z39.50 Servers

UTIL N - Z39.50 Management is used for Z39.50 gate and server configuration. See the Z39.50 documentation available on the Ex Libris Documentation Center under ALEPH 500 Documentation/Z39.50.

5.16 OCLC Servers

OCLC records can be dynamically loaded into ALEPH from a PC connected to OCLC with the OCLC Passport, OCLC CatME or OCLC Connexion software. Please consult the document, How To Load OCLC Records into ALEPH available on the Ex Libris Documentation Center under ALEPH 500 Documentation/How To/Cataloging.

5.17 SIP2 Server (3M Standard Interchange Protocol)

This protocol provides a standard interface between a library's Automated Circulation System (ACS) and library automation devices.

Self-check systems enable library patrons to borrow and return materials without the assistance of library staff, usually by means of scanning the barcodes on the item and on the patron's library card.
See the document, *How To Set Up a SIP2 Server for Self-Check Systems* available on the [Ex Libris Documentation Center](https://www.exlibrisgroup.com/documentation) under **ALEPH 500 Documentation/How To/Circulation**.

### 5.18 NCIP Server (NISO Circulation Interchange Protocol)

This standard defines a protocol that is limited to the exchange of messages between and among computer-based applications to enable them to:

- Perform the functions necessary to lend and borrow items.
- Provide controlled access to electronic resources.
- Facilitate co-operative management of these functions.

See the document, *NISO Circulation Interchange Protocol Server NCIP Server* available on the [Ex Libris Documentation Center](https://www.exlibrisgroup.com/documentation) under **ALEPH 500 Documentation/How To/Circulation**.

### 5.19 Interlibrary Loan Server (ILL Server)

The Interlibrary Loan Server is a server that participates in handling ISO ILL transactions. It is essential for handling ILL processes that support ISO 10160 and 10161 standards via the ILL module.

The ILL Server performs three major tasks in the process of handling an ISO ILL transaction:

1. Receiving the ILL APDUs (Application Protocol Data Units) and decoding them.
2. Storing the packets received by the local server, as indexed documents in the ILLSV library.
3. Logging APDUs sent from the local server. This is also done in the ILLSV library.

The ILL Server is the link between the ILL module and the communications system by which the ILL APDUs are delivered, translating protocol packets into library documents.

#### 5.19.1 Running the ILL Server

The ILL Server is managed in the same manner as other servers, such as the PC Server and the WWW Server, using **UTIL W Server Management**. In each procedure, the ILL Server is listed under **Other Server**.

#### 5.19.2 The ILLSV Library

Once ISO ILL APDUs have been received and decoded they are written as documents into a library named **ILLSV**. This library stores the packets received by the local server as indexed documents.
There are two types of fields in the record:

1. Fields with a numeric tag – These are the bibliographic fields expanded from the APDU.

   **Example:** 245 L $$aSpiders of Britain

2. Fields with the ILL tag. These fields are the exact fields of the APDU. They always have a subfield a with the name of the tag in the APDU, and a subfield b with the value.

   **Example:** ILL L $$aIllRequest.ClientId.ClientName$$bABEP

Records logged in the ILL Server that have a 466 field with a subfield a value of LOG, are APDUs sent from the local server. For example:

   466 L $$aLOG$$bREQUEST$$c20031026

All other records are APDUs received from other servers.
6 Data Warehouse

The ALEPH Data Warehouse is a copy of the production data used by the ALEPH software. This copy, built automatically using Oracle triggers, collects all the required data in one place (under a single Oracle user), and enables the delivery of the data in a structure that is optimal for queries, reports and statistics. All the customization required to activate the ALEPH Data Warehouse (like Oracle triggers) is done within the ALEPH Data Warehouse. This means that the Data Warehouse’s existence and its implications are transparent to the production environment.

Refer to your Ex Libris representative for more details.