ALEPH500 System Administration Guide

Release 14.2

Last update: April 29, 2001
# 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 DIRECTORY STRUCTURE ................................................................. 8
  2.1 The aleph Root .................................................................................................. 8
  2.2 The Main Directories ......................................................................................... 9
      2.2.1 aleph ........................................................................................................... 9
      2.2.2 aleph ........................................................................................................... 9
      2.2.3 aleph ........................................................................................................... 10
      2.2.4 Library Root Directories ........................................................................... 10
      2.2.5 ALEPH Unix Logins: ................................................................................ 10

## 3 ALEPH / ORACLE ........................................................................................ 11
  3.1 Structure ........................................................................................................... 11
      3.1.1 SQL Access to the Oracle Tables: ............................................................... 11
      3.1.2 ALEPH Access to the Oracle Tables ............................................................ 12
      3.1.3 Passwords .................................................................................................. 13
  3.2 Oracle Concepts ................................................................................................ 13
      3.2.1 Storage ......................................................................................................... 13
      3.2.2 Users ............................................................................................................ 14
      3.2.3 Tables .......................................................................................................... 14
      3.2.4 Indexes .......................................................................................................... 15
      3.2.5 Triggers ........................................................................................................ 15

## 4 ALEPH LOGICAL ENVIRONMENT ............................................................ 16
  4.1 aleph start 505 ................................................................................................. 16
      4.1.1 # aleph start: ............................................................................................. 17
  4.2 Logical Names Per Unix User .......................................................................... 20
      4.2.1 For Unix user m505 .................................................................................. 20
      4.2.2 ALEPH Libraries ......................................................................................... 21
  4.3 A Library Tree .................................................................................................. 21
  4.4 Oracle Tables Management ............................................................................. 22
      4.4.1 file list ......................................................................................................... 22

## 5 LIBRARY UTILITIES ...................................................................................... 24
  5.1 UTIL A - File administration and building ................................................... 24
      5.1.1 A.10 Drop result set table (z05, z110) ......................................................... 25
      5.1.2 A.11 Drop session tables (z63/z64/z65) ...................................................... 25
      5.1.3 A.12 Drop lock table (z60/z50) ................................................................. 25
      5.1.4 A.13 Drop statistics table (z34) ................................................................. 26
      5.1.5 A.14 Drop "update doc" table (z07) ............................................................ 26
      5.1.6 A.15 Drop web "basket" table (z109) .......................................................... 27
      5.1.7 A.17 Manage Oracle tables ..................................................................... 28
      5.1.8 A.19 Export Library .................................................................................. 30
      5.1.9 A.20 Import Library .................................................................................. 31
  5.2 UTIL C - Monitor batch jobs ......................................................................... 31
      5.2.1 C.1 Check library running process [10-12] ............................................. 31
      5.2.2 C.2 Start library batch queue [10-10] ....................................................... 31
      5.2.3 C.3 Stop library batch queue [10-10] ....................................................... 32
      5.2.4 C.4 Check library lock status [10-1] .......................................................... 32
      5.2.5 C.5 Lock library [18] ............................................................................... 33
      5.2.6 C.6 Unlock Library [19] .......................................................................... 33
      5.2.7 C.7 Show queue of jobs waiting execution [10-3] .................................... 34
6 ALEPH SERVERS, BATCH JOBS, AND PROBLEM DIAGNOSIS

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

6.2 Server Logs

6.3 Starting Your Own Server for Testing

6.4 PC client/PC server

6.4.1 Connecting

6.4.2 Service Requests

6.5 Analyzing PC Server Problems

6.5.1 PC Server time-out

6.6 ue daemons

6.7 Batch jobs

6.7.1 Web Services

6.7.2 Chron Jobs

6.8 www_server (Public)

6.9 www_server (Staff)

6.10 Z39 Servers

6.11 OCLC Servers

6.12 sc_server (Self-Check server)

6.13 SQL...
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 5 interrelated, yet separate, units: Authority, Bibliographic, Holdings, Administrative 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 scalability potential.

ALEPH runs under a wide range of platforms:

- IBM/AIX
- ALPHA UNIX
- SUN SOLARIS
- HP-UX
- LINUX REDHAT

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. 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. 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), which ensures both maintainability and extensibility of the system and integrateability with new technologies and concepts. ALEPH's modularity plays a crucial role in the system's scaleability potential.
1.1 ALEPH Server Architecture

**Multitier Architecture - OPAC**

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)** - In the core 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 once 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 is based on Oracle RDBMS. Due to the textual, non-formatted nature of a library catalog, ALEPH contains SQL “enrichment” to ensure high performance.

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 distinguished 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. See Figure 2 below.
- **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 Directory Structure

2.1 The aleph Root

An aleph root is composed of three types of directories:

1. ALEPH software: alephm, aleph, tmp, usm01, usm10, ...
2. Node management: alephe
3. Libraries: rug01, rug02, ...

After initial installation, it is highly recommended to separate the software directories (type 1) from the site-specific directories (types 2 and 3).

A sample division is given in Figure 4:
With the above configuration, when the system is upgraded and a newer server version is installed, the \texttt{a50\_5} root is replaced, whereas the \texttt{u50\_5} root remains as is (unless data conversions or updates are required).

2.2 The Main Directories

2.2.1 alephm

The \texttt{alephm} directory (alias "am") contains software installation, development and maintenance programs and procedures.

The \texttt{alephm} tree:

\begin{itemize}
\item \texttt{proc} compilation scripts (alias "apm")
\item \texttt{scratch} scratch directory (alias "ac")
\item \texttt{source\_lib} customized and private programs and scripts maintained by Ex Libris
\item \texttt{source\_user} source libraries available for the customer
\item \texttt{sql\_tab} sql scripts for creation of tables and table dumps
\end{itemize}

2.2.2 aleph

The \texttt{aleph} directory contains the ALEPH run time modules
The **aleph** tree:

-- **exe**
    executable images (alias "ae")

-- **pc_exe.n**
    ALEPH clients executables:
    -- **win95**
        Windows95 and WindowsNT

-- **proc**
    aleph scripts (alias "ap")

### 2.2.3 alephe

The **alephe** directory includes Node management tables and parameter files pertaining to the server.

The **alephe** tree:

-- **tab**
    node management tables

-- **www**
    WWWtables and icons:
    -- **icons**
        icons per supported language:
        eng
        fre
        ....

-- **scratch**
    intermediate files, batch log files

-- **apache**
    conf/htdocs/logs

-- **backup**
    scripts and config files for backup

### 2.2.4 Library Root Directories

Every ALEPH library (rug01, rug50, ndu01, etc.) has a separate root directory which is also part of this tree. See sections 3.1 "ALEPH /Oracle: Structure" and 5.1 "ALEPH Libraries: A Library Tree" for further information.

### 2.2.5 ALEPH Unix Logins:

The ALEPH server requires the following Unix Users:

<table>
<thead>
<tr>
<th>Login</th>
<th>Home Directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>m505</td>
<td>a50_5/alephm</td>
</tr>
<tr>
<td>oracle</td>
<td>SORACLE_HOME</td>
</tr>
</tbody>
</table>

The person who is in charge of all the libraries, the System Administrator, should login as **m505**, and thus will be able to modify the parameters and data of all libraries.
3 ALEPH / Oracle

3.1 Structure

ALEPH500 is based on Oracle8 RDBMS.

In ALEPH, every library has a separate root directory. Each ALEPH library directory contains information relevant to administrating the library. Further details are given in Chapter 5: "ALEPH Libraries".

Each library’s database is implemented within Oracle. The general structure of the Oracle database implementation vis-a-vis ALEPH libraries is given below:

Each ALEPH Library is implemented as a separate Oracle user. Each Oracle user owns a set of tables which contain the Library’s data.

![Diagram of ALEPH and Oracle structures]

**Figure 5**

3.1.1 SQL Access to the Oracle Tables:

Any Oracle tool can be used to access the ALEPH/Oracle tables.

For example, SQL*Plus:

```
USM01-M505>sqlplus usm60/usm60
SQL*Plus: Release 3.2.3.0.0 - Production on Tue Nov 12 12:08:53 1996
Copyright (c) Oracle Corporation 1979, 1994. All rights reserved.
Connected to:
```

Oracle8 Server Release 8.0.6.0.0 - Production Release
With the distributed option
PL/SQL Release 8.0.6.0.0 - Production

SQL-USM60> desc z81
Name Null Type
---------- ----- -------
Z81_ID   NOT NULL CHAR(12)
Z81_SUB_LIBRARY CHAR(5)
Z81_OPEN_DATE   NUMBER(8)
Z81_UPDATE_DATE NUMBER(8)
Z81_BOR_TYPE CHAR(2)
Z81_BOR_STATUS CHAR(2)
Z81_EXPIRY_DATE NUMBER(8)
Z81_NOTE   VARCHAR2(80)
Z81_LOAN_PERMISSION CHAR(1)
Z81_PHOTO_PERMISSION CHAR(1)
Z81_OVER_PERMISSION CHAR(1)
Z81_MULTI_HOLD CHAR(1)
Z81_LOAN_CHECK   CHAR(1)
Z81_DELIVERY_TYPE CHAR(1)
Z81_HOLD_PERMISSION CHAR(1)
Z81_RENEW_PERMISSION CHAR(1)
Z81_IGNORE_LATE_RETURN CHAR(1)
Z81_NO_LOAN   NUMBER(4)
Z81_NO_HOLD   NUMBER(4)
Z81_NO_PHOTO   NUMBER(4)
Z81_NO_CASH   NUMBER(4)
Z81_CREDIT_DEBIT CHAR(1)
Z81_SUM   CHAR(10)

SQL-USM60> select z81_id from z81;

Z81_ID
--------
ALEPH
ORACLE

2 rows selected.

SQL-USM60> select Z81_ID,Z81_NO_LOAN,Z81_NO_PHOTO from z81 where 2  z81_id = 'ALEPH';

Z81_ID   Z81_NO_LOAN   Z81_NO_PHOTO
-------- ----------- ----------
ALEPH    5            7

1 row selected.

3.1.2 ALEPH Access to the Oracle Tables

The ALEPH server connects to the Oracle databases through a dedicated Oracle user named ALEPH (default password: ALEPH). The dedicated ALEPH user can select, insert, update and delete data from the tables of all Oracle users (e.g. rug01, rug50, rug60, etc.), but is not the owner of any table.

The second administrative user in the Oracle database is named ALEPH_ADMIN. This is a more privileged user, who in addition to the privileges of ALEPH user can create, drop and alter Oracle tables, indexes, users, triggers etc. The ALEPH_ADMIN Oracle user is used for these purposes in all ALEPH procedures.
The third and last administrative Oracle user for ALEPH is **ALEPH_DBA**. This user should be used only if the customer decides to administrate the Oracle database using ALEPH tools (UTIL O). This is the most privileged Oracle administrative user. It is used by ALEPH utilities to start up, shut down, delete all or part of the database, and so on.

The connection between ALEPH servers and procedures and these Oracle users is transparent to the ALEPH end user (using the WWW or PC or UTIL interfaces).

### 3.1.3 Passwords

ALEPH contains an encrypted file with the passwords of the Oracle administrative users used by ALEPH and described in the previous section, namely ALEPH, ALEPH_ADMIN and ALEPH_DBA. This means in effect that whenever you decide to change the password of these three Oracle administrative users, you must update the password both in the Oracle database and in the ALEPH password file. You update the ALEPH password file by using UTIL Y/8. You update passwords in the Oracle database by using one of the Oracle interfaces: SQL*Plus or Oracle Security Manager. In SQL*Plus the command is:

```
sqlplus system/manager
alter user <username> identified by <new password>;
exit;
```

Password updates of all other Oracle users do not require updating of the ALEPH password file.

### 3.2 Oracle Concepts

#### 3.2.1 Storage

An Oracle database consists of several logical units named tablespaces. Each tablespace consists of one or more physical datafiles which can be stored on one or more disks. For example:

<table>
<thead>
<tr>
<th>Tablespace name</th>
<th>Usage</th>
<th>Physical File</th>
</tr>
</thead>
<tbody>
<tr>
<td>System</td>
<td>Oracle system tables</td>
<td>/aleph/oradata/aleph0/system01.dbf</td>
</tr>
<tr>
<td>temp</td>
<td>Temporary space (for sorting, etc...)</td>
<td>/aleph2/oradata/aleph0/temp01.dbf</td>
</tr>
<tr>
<td>ts0</td>
<td>ALEPH tables</td>
<td>/aleph2/oradata/aleph0/ts0_0.dbf</td>
</tr>
<tr>
<td>ts1</td>
<td>ALEPH indexes</td>
<td>aleph/oradata/aleph0/ts1_0.dbf</td>
</tr>
</tbody>
</table>
Each Oracle table and index has to be mapped to a tablespace. In ALEPH, this mapping is done via a configuration file named file_list. The file_list file is located under the root of each library.

Example: Tablespace Creation:

```
SQL-ALEPH> CREATE TABLESPACE ts0
  2   DATAFILE '/aleph/oradata/aleph0/ts0_0.dbf' SIZE 50M
  4   DEFAULT STORAGE (INITIAL 10K NEXT 10K MINEXTENTS 1
  5   PCTINCREASE 0)
  6   ONLINE;
```

### 3.2.2 Users

In an Oracle database, users can be defined and identified by usernames. A user has:

- Default tablespace - which specifies where objects (tables and indexes) are built by default (unless explicitly specified otherwise).
- Temporary tablespace - Provides storage for SQL statements that require disk space to sort or summarize data.
- Tablespace quotas - the maximum space a user allocate for each tablespace.

Example - Create user rug01:

```
SQL-ALEPH> CREATE USER rug01
  2   IDENTIFIED BY rug01
  3   DEFAULT TABLESPACE ts0
  4   TEMPORARY TABLESPACE temp
  5   QUOTA UNLIMITED ON ts0
  6   QUOTA UNLIMITED ON temp;
```

### 3.2.3 Tables

A table is an Oracle object which contains rows of data. A row is composed of columns. Each table is mapped to a tablespace. For each table, Oracle allocates initial space and extended space, according to the specification in its CREATE TABLE command. Tablespace mapping, and initial and extended allocation for each table are controlled in ALEPH in a special configuration file named file_list. The file_list file is located under the root of each library.

Example of CREATE TABLE:

```
CREATE TABLE RUG01.Z01 ( 
  Z01_REC_KEY CHAR(44) NOT NULL,
  Z01_REC_KEY_BACK RAW(44) NOT NULL,
  Z01_ACC_SEQUENCE CHAR(8) NOT NULL,
  Z01_AUT_LIBRARY CHAR(5),
  Z01_AUT_DOC_NUMBER NUMBER(7),
  Z01_ACC_SEQUENCE_SEE NUMBER(8),
```
### 3.2.4 Indexes

An Oracle index is an Oracle object (B-tree) which contains pointers (rowid) to a specific row in a table. Each index is mapped to a tablespace. For each index, Oracle allocates initial space and extended space, according to the specification in its CREATE INDEX command. Tablespace mapping, and initial and extended allocation for each index are controlled in ALEPH in a special configuration named `file_list`. The `file_list` file is located under the root of each library.

Example of CREATE INDEX:

```sql
CREATE UNIQUE INDEX RUG01.Z01_ID ON RUG01.Z01 (Z01_REC_KEY ASC)
STORAGE (INITIAL 500K NEXT 300K MINEXTENTS 1 PCTINCREASE 0)
PCTUSED 80
PCTFREE 10
TABLESPACE ts1;
```

### 3.2.5 Triggers

A database trigger is a stored PL/SQL block that is associated with a table. Oracle automatically executes a trigger when a specified SQL statement is issued against the table. The trigger can be executed before or after the SQL statement is issued against the table. Once the trigger has been created it can be disabled or enabled, which means that Oracle either will not fire the trigger (disabled) or will fire it (enabled).

Example of CREATE TRIGGER:

```sql
CREATE OR REPLACE TRIGGER RUG50.Z34_TRIGGER
AFTER UPDATE OF Z30_ITEM_STATUS,Z30_SUB_LIBRARY,Z30_COLLECTION
ON RUG50.Z30
FOR EACH ROW
WHEN ( NEW.Z30_ITEM_STATUS != OLD.Z30_ITEM_STATUS OR
       NEW.Z30_SUB_LIBRARY != OLD.Z30_SUB_LIBRARY OR
       NEW.Z30_COLLECTION != OLD.Z30_COLLECTION )
BEGIN
    INSERT INTO RUG50.Z34
    VALUES ( SUBSTR(OLD.Z30_REC_KEY,1,7),
             SUBSTR(OLD.Z30_REC_KEY,8,6),
             OLD.Z30_BARCODE,
             OLD.Z30_COPY,
             OLD.Z30_SUB_LIBRARY,
             OLD.Z30_ITEM_STATUS,
             SYSDATE,
             OLD.Z30_COLLECTION );
END;
/
4 ALEPH Logical Environment

4.1 aleph_start_505

aleph_start_505 is a file that contains definitions of ALEPH libraries (rug01, usm01, etc.) and directories. It also contains logical assignments (the only place with physical references to ALEPH directories). It is invoked by every ALEPH process.

aleph_start_505 is placed in alephe/aleph_start_505, and /etc/aleph_start_505 is a symbolic link to alephe/aleph_start_505.

An example of aleph_start_505 is shown on the next page. Following is a key to each section of the file:

1. General information about the version
2. Libraries’ path assignments
3. Set and Password libraries
4. Oracle environment variables
5. Z39.50 and I/O Servers’ configurations
6. ALEPH directories’ path assignments
7. Automatic startup configuration
8. Logical names for ALEPH directories
4.1.1 # aleph_start:
#
### Version and/or platform dependent setting

# Hostnames
#
setenv ORA_HOST ram25
setenv IO_HOST ram25
setenv Z39_HOST ram25

# Oracle Application Server & Port
setenv OAS_HOST ram25
setenv OAS_PORT 8899

# Directories: ALEPH_MOUNT - software + demo libraries
# USER_MOUNT - user libraries
#
setenv ALEPH_MOUNT /aleph
setenv USER_MOUNT /aleph

setenv ALEPH_VERSION 505
setenv PATCH_NUMBER 14
setenv SUB_PATCH_NUMBER 2
setenv VERSION_NUMBER 0
setenv HTTPD_PORT 81

setenv usm01_dev ${ALEPH_MOUNT}/a5${VERSION_NUMBER}_5
setenv usm09_dev ${ALEPH_MOUNT}/a5${VERSION_NUMBER}_5
setenv usm10_dev ${ALEPH_MOUNT}/a5${VERSION_NUMBER}_5
setenv usm11_dev ${ALEPH_MOUNT}/a5${VERSION_NUMBER}_5
setenv usm12_dev ${ALEPH_MOUNT}/a5${VERSION_NUMBER}_5
setenv usm19_dev ${ALEPH_MOUNT}/a5${VERSION_NUMBER}_5
setenv usm20_dev ${ALEPH_MOUNT}/a5${VERSION_NUMBER}_5
setenv usm30_dev ${ALEPH_MOUNT}/a5${VERSION_NUMBER}_5
setenv usm50_dev ${ALEPH_MOUNT}/a5${VERSION_NUMBER}_5
setenv usm51_dev ${ALEPH_MOUNT}/a5${VERSION_NUMBER}_5
setenv usm60_dev ${ALEPH_MOUNT}/a5${VERSION_NUMBER}_5
setenv uni01_dev ${ALEPH_MOUNT}/a5${VERSION_NUMBER}_5
setenv uni10_dev ${ALEPH_MOUNT}/a5${VERSION_NUMBER}_5
setenv uni20_dev ${ALEPH_MOUNT}/a5${VERSION_NUMBER}_5
setenv uni30_dev ${ALEPH_MOUNT}/a5${VERSION_NUMBER}_5
setenv uni50_dev ${ALEPH_MOUNT}/a5${VERSION_NUMBER}_5
setenv uni60_dev ${ALEPH_MOUNT}/a5${VERSION_NUMBER}_5
setenv vir00_dev ${ALEPH_MOUNT}/a5${VERSION_NUMBER}_5
setenv vir01_dev ${ALEPH_MOUNT}/a5${VERSION_NUMBER}_5
setenv swt01_dev ${ALEPH_MOUNT}/a5${VERSION_NUMBER}_5
setenv ext01_dev ${ALEPH_MOUNT}/a5${VERSION_NUMBER}_5
setenv ext02_dev \${ALEPH_MOUNT}/a5\${VERSION_NUMBER}_5
setenv ext03_dev \${ALEPH_MOUNT}/a5\${VERSION_NUMBER}_5
setenv ext04_dev \${ALEPH_MOUNT}/a5\${VERSION_NUMBER}_5

# In the next section make sure that only ONE space exists between libraries
# and NO space between "" and the libraries.
# (for example " SDEMO_LIBS aut01 aut02 " is not correct).
#
setenv DEMO_LIBS "usm01"
setenv DEMO_LIBS "$DEMO_LIBS usm10 usm11 usm12 usm19"
setenv DEMO_LIBS "$DEMO_LIBS usm20 usm30"
setenv DEMO_LIBS "$DEMO_LIBS usm50 usm51 usm60"
setenv DEMO_LIBS "$DEMO_LIBS uni01 uni10 uni20 uni30"
setenv DEMO_LIBS "$DEMO_LIBS uni50 uni60"
setenv DEMO_LIBS "$DEMO_LIBS vir00 vir01"
setenv DEMO_LIBS "$DEMO_LIBS ext01 ext02 ext03 ext04"
setenv DEMO_LIBS "$DEMO_LIBS swt01"

setenv ALEPH_LIBS "$DEMO_LIBS"

setenv pw_library USM50
setenv v_library VIR01
setenv control_lng ENG
setenv default_character_conversion 8859_1_TO_UTF

setenv QUE_STARTUP_LIBS "usm01 usm50 usm20 usm30"
setenv WORD_STARTUP_LIBS "usm01 usm10 usm20 usm30 usm50"
setenv REQUEST_STARTUP_LIBS "usm50"
setenv ACC_AUT_STARTUP_LIBS "usm01"
setenv MESSAGE_STARTUP_LIBS "usm01 usm10 usm50"
setenv RLIN_STARTUP_LIBS "usm01"
setenv SC_STARTUP_LIBS "usm50"

setenv Z39_SERVER_STARTUP Y
setenv Z39_GATE_STARTUP Y
setenv OCLC_SERVER_STARTUP Y

# If you want Aleph to connect to a remote Oracle DataBase
# setenv BOTH 'aleph_db' and 'TWO_TASK' to the SAME service
#
setenv ORACLE_SID aleph2
setenv ORACLE_OWNER oracle
setenv ORACLE_VERSION 80
setenv ORACLE_CONF \${ALEPH_MOUNT}/ora_aleph
setenv NLS_LANG American_America.US7ASCII
setenv ORACLE_HOME /aleph/app/oracle/product/806

# setenv aleph_db \${ORA_HOST}.\${ORACLE_SID}
setenv backup_dev /dev/rmt0
#
setenv server_max_port 20000
setenv io_server_address ${IO_HOST}:4${VERSION_NUMBER}23
setenv z39_gate_address ${Z39_HOST}:99${VERSION_NUMBER}7
#
setenv aleph_dev ${ALEPH_MOUNT}/a5${VERSION_NUMBER}_5
setenv alephm_dev ${ALEPH_MOUNT}/a5${VERSION_NUMBER}_5
#
setenv alephe_dev ${USER_MOUNT}/u5${VERSION_NUMBER}_5
#
setenv alephe_dev ${ALEPH_MOUNT}/a5${VERSION_NUMBER}_5
alephe_synch_dev
#
setenv aleph_product /aleph/product
setenv TMPDIR ${ALEPH_MOUNT}/a5${VERSION_NUMBER}_5/tmp
setenv TMPMEDIA ${ALEPH_MOUNT}/a5${VERSION_NUMBER}_5/tmpmed
ia
setenv LOGDIR ${ALEPH_MOUNT}/a5${VERSION_NUMBER}_5/log
setenv PROCESS_NUMBER $$
setenv FILE_TMPDIR $TMPDIR
setenv httpd_bin /aleph/product/apache_1.3.12/bin
setenv httpd_root ${ALEPH_MOUNT}/a5${VERSION_NUMBER}_5/apache
setenv aleph_11_5_support Y
#
#
# Use aleph_start_505.private to define customer libraries
#
if (-f $alephe_dev/alephe/aleph_start_505.private) then
    source $alephe_dev/alephe/aleph_start_505.private
endif
#
#
# Set compiler_type to 0 (zero) for rte (runtime only) or
# current machine type for full compiler
#
setenv platform_type 1
#
switch ($platform_type)
case 1:
    setenv unix_name ALPHA
#    setenv COBDIR /aleph/product/cobol4.1
    setenv COBDIR /aleph/product/cobol-3.2

setenv LANG en_US.ISO8859-1
setenv JAVA_HOME /usr
breaksw
case 2:
  setenv unix_name AIX4
  setenv COBDIR /aleph/product/cobolv4.0
  # setenv COBDIR /aleph/product/cobolv4.1
  setenv LANG en_US
  setenv JAVA_HOME /aleph/product/jdk/J1.1.6
  breaksw
case 3:
  setenv unix_name SUN_O.S_2
  setenv COBDIR /aleph/product/cobolv4.0

# --> NOTE: for proper compilation set the version number
# --> to the correct value
if ($?LD_LIBRARY_PATH) then
  setenv LD_LIBRARY_PATH "${LD_LIBRARY_PATH}:${ORACLE_HOME}/lib"
else
  setenv LD_LIBRARY_PATH "${ORACLE_HOME}/lib"
endif
#
setenv EPC_DISABLED TRUE
exit
~

4.2 Logical Names Per Unix User
Logical assignments and aliases are defined in the following files:

4.2.1 For Unix user m505:

/etc/aleph_start_505
$alephm_dev/prof_master
$aleph_proc/def_local_env
4.2.2 ALEPH Libraries

As described in section 3, every library in ALEPH has a separate root directory. Each ALEPH library directory contains information relevant to administrating the library. Each library’s database is implemented within Oracle.

4.3 A Library Tree

Here is an example of a standard library tree:

rug01

file_list  Configuration file for Oracle tables and indexes.

prof_library  Logical assignments pertaining to the library.

files/  Contains:
Library export files (rug01.export)
Dump files (znn.seq)
Batch queue (que_batch)
Batch log (batch_log)

Alias "df1" takes you to /alephn/u50_5/aaann (example: /aleph1/u50_5/rug01 ).

tab/  Library parameter and configuration tables. Accessed via UTIL (alias "dt").

scratch/  Scratch directory. Used for intermediate and log files of UTIL scripts and programs (alias "ds").

pc_tab/  Clients parameter and configuration tables.

print/  Print files (alias "dp").

source/  Source data for conversion.

www_a_lng/  HTML files, per language, for the ALEPH/WWW server. Accessed via UTIL.

The tab, pc_tab and www_a_lng directories and library parameter and configuration tables are described in the ALEPH500 Database Management Guide.

Usually the customer's files will be part of a different file system ("aleph1", "aleph2", etc., instead of "aleph"). Thus, the path will be /alephn/u50_5/aaann .

The "dr" (data root) alias takes you to /alephn/u50_5/aaann.
4.4 Oracle Tables Management

4.4.1 file_list

The file_list is located under each library root directory. It consists of parameters used to manage all Oracle objects (table, index, synonym) of the particular library.

Here is an example of the different objects listed in file_list:

<table>
<thead>
<tr>
<th>TAB</th>
<th>table name</th>
<th>initial allocation</th>
<th>next allocation</th>
<th>tablespace name</th>
</tr>
</thead>
<tbody>
<tr>
<td>z01</td>
<td>2M</td>
<td>1M</td>
<td>ts0</td>
<td></td>
</tr>
<tr>
<td>z01_id</td>
<td>1M</td>
<td>1M</td>
<td>tsl</td>
<td></td>
</tr>
<tr>
<td>IND</td>
<td>index_name</td>
<td>initial allocation</td>
<td>next allocation</td>
<td>tablespace name</td>
</tr>
<tr>
<td>z01_id1</td>
<td>1M</td>
<td>1M</td>
<td>ts1</td>
<td></td>
</tr>
<tr>
<td>TRIGGER</td>
<td>trigger name</td>
<td>z34_trigger</td>
<td></td>
<td>usm50</td>
</tr>
<tr>
<td>LS</td>
<td>table name</td>
<td>library name (to link to)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>z30</td>
<td>30</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SEQ</td>
<td>sequence name</td>
<td>last_result_set</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TAB</td>
<td>table name</td>
<td>initial allocation</td>
<td>next allocation</td>
<td>tablespace name</td>
</tr>
<tr>
<td>RAM36.aleph0</td>
<td>2M</td>
<td>1M</td>
<td>ts0</td>
<td></td>
</tr>
<tr>
<td>IND</td>
<td>index_name</td>
<td>initial allocation</td>
<td>next allocation</td>
<td>tablespace name</td>
</tr>
<tr>
<td>RAM36.aleph0</td>
<td>1M</td>
<td>1M</td>
<td>ram36.aleph0</td>
<td></td>
</tr>
<tr>
<td>IND</td>
<td>index_name</td>
<td>initial allocation</td>
<td>next allocation</td>
<td>tablespace name</td>
</tr>
<tr>
<td>RAM36.aleph0</td>
<td>1M</td>
<td>1M</td>
<td>ram36.aleph0</td>
<td></td>
</tr>
<tr>
<td>RS</td>
<td>link to a remote table</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>z98</td>
<td>usm01</td>
<td>ram36.aleph0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Notes:

In the case of table z02, the table and indexes are physically located on a different node (ram36.aleph0), but can be referenced and maintained on the local node as well.

In the case of table Z98, the table and indexes are physically located on a different node (ram36.aleph0), but can be referenced on the local node as well.
An alias to the node referred to in notes 1 and 2 (such as ram36.aleph0) must be defined in the Oracle aliases file: SORACLE_HOME/network/admin/tnsnames.ora

For example:

```latex
ram36.aleph0=(description=
    (address=
      (protocol=ipc)
      (key=aleph0))
    (address=
      (protocol=tcp)
      (host=ram36)
      (port=1521))
    (connect_data=(sid=aleph0)))
```

alias is always host-name.database-name

For more information see also two sections in Chapter 6, "Library Utilities": "Working With Remote Tables" and "UTIL O/16".
5 Library Utilities

Each library’s database is implemented within Oracle as a separate Oracle user. Each Oracle user owns a set of tables which contain the library’s data. For more details see Chapter 3, "ALEPH / Oracle".

Each library has access to the ALEPH utilities by activating:

   RUG01>> util

The UTIL main screen will appear:

Library Utilities
------------------
A. File administration and building
C. Monitor batch jobs
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. Server configuration
K. ILL tables
L. GUI tables
M. GUI CATALOGING tables
N. Z39.50 management
O. ORACLE management
P. Unicode tables
S. Statistics
W. Server management (Monitor, Stop, Start, Log files)
X. Clean up
Y. Node management

Please select [exit]:

Note: All the UTIL functions not included here in the System Administration Guide are covered in the ALEPH500 Database Management Guide.

5.1 UTIL A - File administration and building

A. File administration and building for USM01
   -----------------------------------------------
   0. Exit procedure
   10. Drop result set table (z05,z110)
   11. Drop session tables (z63/z64/z65)
   12. Drop lock table (z60/z50)
   13. Drop statistics table (z34)
   14. Drop update doc table (z07)
   15. Drop web basket table (z109)
   16. Drop event table (z35)
   17. Manage Oracle tables
   18.
   19. Export library
   20. Import library

Please select [exit]:
5.1.1 A.10 Drop result set table (z05, z110)

Function: Deletes the search result sets from Oracle tables z05 and z110
Activation: UTIL A/10
Parameters: Run time (or q to exit)

The z05 and z110 tables contain the search result sets of FIND requests. These tables should be deleted weekly with the Drop result set table (z05, z110) utility.

When you select this utility, the system will notify you that you will be deleting tables z05 and z110 in their defined library. For example:

```
delete z05,z110 in USM01
```

Note: This utility may be run from ANY library, even though z05 and z110 are only defined in ONE of the libraries through the environment variable set_library in the /etc/aleph_start_505 file.

5.1.2 A.11 Drop session tables (z63/z64/z65)

Function: Deletes session records from Oracle tables z63, z64 and z65
Activation: UTIL A/11
Parameters: Run time (or q to exit)

Each time a user enters ALEPH via the web a session is started. The session definition is written in tables z63, z64 and z65. These tables should be deleted weekly with the Drop session tables (z63/z64/z65) utility.

When you select this utility, the system will notify you that you will be deleting tables z63, z64 and z65 in their defined library. For example:

```
delete z63/z64/z65 in VIR01
```

Note: This utility may be run from ANY library, even though z63, z64 and z65 are only defined in the VIR01 library.

5.1.3 A.12 Drop lock table (z60,z50)

Function: Deletes record locks from Oracle table z60,z50
Activation: UTIL A/12
Parameters: Run time (or q to exit)

When a record is locked, a line is written in table z60. This table should be deleted weekly with the Drop lock table (z60,z50) utility.

When you select this utility, the system will notify you that you will be deleting table z60 in its defined library. For example:

```
delete z60/z50 in USM50/USM01
```
Note: This utility may be run from ANY library, even though z60 is only defined in ONE of the libraries through the environment variable pw_library in the /etc/aleph_start_505 file.

5.1.4 A.13 Drop statistics table (z34)

Function: Deletes statistics from Oracle table z34
Activation: UTIL A/13
Parameters: Run time (or q to exit)

The z34 table contains statistics about transactions between the client and the server. If the create_statistics flag is set to Y in the aleph/proc/www_server file and the aleph/proc/pc_server file, then a z34 record is generated each time a transaction takes place between the client and the server. The z34 table should be deleted weekly with the Drop statistics table (z34) utility.

When you select this utility, the system will notify you that you will be deleting table z34 in its defined library. For example:

    delete z34 in USM50

Note: This utility may be run from ANY library, even though z34 is only defined in ONE of the libraries through the environment variable pw_library in the /etc/aleph_start_505 file.

5.1.5 A.14 Drop "update doc" table (z07)

Function: Deletes records from Oracle table z07
Activation: UTIL A/14
Parameters: Run time (or q to exit)

When a cataloging (BIB, AUT, HOL) record is created or updated, its system number is placed in the z07 table. The z07 table controls the updating of index files. A smoothly running system should not have many records in the z07 table.

Under very rare circumstances, the cataloger might decide not to run UE_01, and therefore the z07 records will not be deleted automatically by the system. They can be deleted using the Drop "update doc" table (z07) utility.

When you select this utility, the system will notify you that you will be deleting table z07 in its active library. For example:

    delete z07 in USM01
5.1.6  A.15  Drop web "basket" table (z109)

Function: Deletes web "basket" records from Oracle table z109
Activation: UTIL A/15
Parameters: Run time (or q to exit)

Each time a user puts records in a web "basket", information is stored in the z109 table. This table should be deleted weekly with the Drop web "basket" table (z109) utility.

When you select this utility, the system will notify you that you will be deleting table z109 in its defined library. For example:

    delete z109 in VIR01

☞ Note: This utility may be run from ANY library, even though z109 is only defined in the library VIR01.
5.1.7 A.17 Manage Oracle tables

A. Manage Oracle tables of USM01

0. Exit procedure
1. Drop & Create table and index
2. Create index
3. Rebuild index
4. Drop index
5. Create/Recreate all library synonyms
6. Create/Recreate all library triggers
7. Enable library trigger
8. Disable library trigger
9. Create/Recreate library sequences
10. Edit file_list of USM01 tables
11. Check space utilization of Oracle tables
12. Check space utilization of a table/index
13. Check space utilization of dynamic tables
14. List existing indexes for a table
15. Multimedia
16. Show trigger status
17. 

Please select [exit]:

Each option is explained below:

1. **Drop & Create table and index**

   A script to drop and create an Oracle table from the particular library. This script will also drop and create the corresponding indexes.

   Options:

   **Enter table name to Create/Recreate (all/table-name)**

   **all** will create / recreate all of the tables in the library and their indexes.

   **table-name** will create / recreate the specified table and all of its indexes.

2. **Create index**

   A script to create Oracle indexes in a particular library. This script is used to refresh the index.

   Options:

   **Enter table name to Create (all/table-name/index-name):**
all will create / recreate the indexes in all of the tables in the library.

Table-name will create all the indexes for the specified table.

Using a specific index name (e.g. z01_id1) will only create that index.

3. Rebuild index

A script to rebuild an Oracle index in a particular library.

This option is used to refresh the index, and is much faster than dropping an index and recreating it.

Options:

Enter table name to Rebuild (all/table-name/index-name):

all will rebuild all the library’s indexes.

table-name will rebuild all the table’s indexes.

Using a specific index name (e.g. z01_id1) will only rebuild that index.

4. Drop index

A script to drop an Oracle Index.

5. Create/Recreate all library synonyms

A script to create all the library’s synonyms as defined in the file_list.

6. Create/Recreate all library triggers

A script to create all the library’s triggers as defined in the file_list.
7. Enable library trigger

A script to enable a library trigger after it was disabled. After a trigger is created, it is automatically enabled.

8. Disable library trigger

A script to disable a library trigger after it has been enabled.

9. Create/Recreate library sequences

A script to create all the library’s sequences as defined in the file_list.

10. Edit file list of USM01

Edit the file_list of the particular library.

11. Check space utilization of Oracle files

This option generates a report. For each Oracle object in the library it reports the size in Kilobytes of the first allocation and the whole object.

12. Check space utilization of Oracle files

13. Check space utilization of table/index

14. List existing indexes for a table

15. Multimedia

16. Show trigger status

5.1.8 A.19 Export Library

Function: Export Library
Activation: UTIL A/19
Parameters: Enter run time (or q to exit):

Exports all of the library’s Oracle tables into $data_files/<library-name>.export.Z.
The utility uses Oracle’s Export.

Note: Depending on the volume of data, this process might take considerable time and disk space. Make sure that $data_files is free enough.
5.1.9  A.20  Import Library

**Function:** Import Library  
**Activation:** UTIL A/20  
**Parameters:** Enter run time (or q to exit):

Imports all of the library’s data from `$data_files/library-name.export.Z` into Oracle’s tables. The utility uses Oracle’s Import.

**Note:** Depending on data volume, this process might take considerable time.

5.2  UTIL C - Monitor batch jobs

<table>
<thead>
<tr>
<th>C. Monitor batch jobs</th>
</tr>
</thead>
<tbody>
<tr>
<td>0. Exit procedure</td>
</tr>
<tr>
<td>1. Check library running process [10-12]</td>
</tr>
<tr>
<td>2. Start library batch queue [10-10]</td>
</tr>
<tr>
<td>3. Stop library batch queue [10-11]</td>
</tr>
<tr>
<td>4. Check library lock status [10-1]</td>
</tr>
<tr>
<td>5. Lock library [18]</td>
</tr>
<tr>
<td>6. Unlock library [19]</td>
</tr>
<tr>
<td>7. Show queue of jobs waiting execution [10-3]</td>
</tr>
<tr>
<td>8. Delete entry from queue of jobs waiting execution [10-7]</td>
</tr>
<tr>
<td>9. Alter run-time of job waiting execution [10-8]</td>
</tr>
<tr>
<td>11. Display log list of batch jobs [10-5]</td>
</tr>
</tbody>
</table>

Please select [exit]:

5.2.1  C.1  Check library running process [10-12]

**Function:** Checks the library's running processes  
**Activation:** UTIL C/1

Displays all the presently executing jobs of the library, including:

UTIL batch jobs presently executing  
lib_batch (queue manager, see UTIL C/2 below)  
word update (UTIL E/1)  
update BIB-AUT (UTIL E-8)

5.2.2  C.2  Start library batch queue [10-10]

**Function:** Starts the library's batch queue  
**Activation:** UTIL C/2

The library's batch queue is managed through the `que_batch` file in the library's `data_files` directory. All batch job requests are written to this file.
UTIL C/2 starts the **lib_batch** process (e.g. **USM01_batch**), a process which checks the **que_batch** file and sends the earliest waiting job for execution.

After UTIL C/2 is invoked, the process remains running until system shutdown, or until UTIL C/3 is invoked.

UTIL C/2 must be invoked:

When the system is brought up. This can be set automatically in the system's startup procedures.

When the queue has been stopped using UTIL C/3.

**WARNING**: You will find that if jobs are submitted from a terminal to the **batch_que** when the queue is NOT running, the jobs will be **holding**, and will execute as soon as the queue is started.

A log file of the **lib_batch** process, called **run_b.nnnn**, is written in the library's **data_scratch** directory. The number appended to the name indicates the process number, and a new log file is opened each time that the **lib_batch** process is started. This file lists each batch job that was run from **que_batch**.

The **data_scratch** also has log files for the word process, named **run_e_01.<process number>** (e.g. run_e_01.2490).

The **run_e_01** log file displays the number of the last updated document, the number of the last updated document for which ALEPH indexes were updated, and the time.

5.2.3 C.3 Stop library batch queue [10-10]

**Function:** Stops the queue manager

**Activation:** UTIL C/3

Stops the queue manager. This is done by stopping the process that checks the **que_batch** file and sends the earliest waiting job for execution.

5.2.4 C.4 Check library lock status [10-1]

**Function:** Checks the library lock status

**Activation:** UTIL C/4

Users can check via UTIL C/4 whether a library is locked. The system will display a message if the database is locked.

A library can be locked by invoking UTIL C/5, and unlocked by invoking UTIL C/6 (see below).
5.2.5  C.5  Lock library [18]

Function:  Locks the library  
Activation:  UTIL C/5

ALEPH batch procedures that update data files usually lock the library, i.e. make it inaccessible to users. The purpose of the locking is to insure data integrity.

The procedures that lock the library start by checking if the library is locked. If not, the procedure locks the library: it closes files and prevents access to library files.

If the procedures end successfully, the library is unlocked at that time, and access to the library files is enabled. If the procedures are not completed successfully the library remains locked.

5.2.6  C.6  Unlock Library [19]

Function:  Unlocks the library  
Activation:  UTIL C/6

Libraries can be unlocked by invoking UTIL C/6.

Operators should never unlock a library before checking why it was locked:

1. Check the batch log queue via UTIL C/11 for the last job that began execution.

2. Check the queue of batch jobs via UTIL C/7 to determine which jobs are currently running and and what will be executed once the library is unlocked.

3. Check the backup log via UTIL C/12.

Operators should never unlock the library if a batch job is still running, because any job that will re-lock the library will start once the library is unlocked. Therefore before activating UTIL C/6 you should:

? First delete pending jobs (through UTIL C/8)

or

change their projected run time (through UTIL C/9).

1. Then stop the executing job (through UTIL C/8).

Activating UTIL C/6 also activates jobs that run in the background.
5.2.7  C.7  Show queue of jobs waiting execution [10-3]

Function:  Shows the queue of jobs waiting execution
Activation:  UTIL C/7

This procedure displays a list of batch jobs in the library's batch queue (e.g. USM01_batch). The list is chronologically arranged, and provides information about executing jobs and jobs holding (waiting queue). This includes the job name, entry number and status. The time the job should start is also displayed.

Example of batch queue:

<table>
<thead>
<tr>
<th>Jobname</th>
<th>Entry</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>util_a_10_b</td>
<td>00096</td>
<td>Holding until 960914 18</td>
</tr>
</tbody>
</table>

5.2.8  C.8  Delete entry from queue of jobs waiting execution [10-7]

Function:  Deletes the job entry from queue of jobs waiting execution
Activation:  UTIL C/8

This procedure enables operators to delete any executing, or holding job.

After the user enters the job entry number, the system prompts for confirmation of deletion request.

Select the Q ption to return to the UTIL C menu.

Select the S option to display the execution queue.

5.2.9  C.9  Alter run-time of job waiting execution [10-8]

Function:  Alters the run-time of jobs waiting execution
Activation:  UTIL C/9

This procedure can be used to change the run time of a job which is pending.

When UTIL C/9 is invoked, the system displays the execution queue, and a request to select an option. If the operator enters the job entry number, the system will prompt for the requested run time.

Jobs are executed according to the order of their submission into the queue, but not before the hour indicated for their start. If the indicated hour is less than the current hour, the job will enter the queue immediately. The valid start hour range is between 01 to 99:

- 01-24: clock hour of the current day.
- 25-48: clock hour of the next day (subtracting 24).
- 49-72: clock hour after 2 days (subtracting 48).
- 73-96: clock hour after 3 days (subtracting 72).
Select the Q option to return to the main menu.

5.2.10 C.10 Display logfile of a batch job [10-4]
Function: Displays the logfile of batch jobs
Activation: UTIL C/10

This procedure displays a log file of a batch job procedure.

Q returns the user to the UTIL C menu.

Enter the name of the procedure (e.g. 96) in order to display the file. If you are uncertain as to the name of the procedure, enter D to display a list with names of jobs and times they were sent, sequenced by name and cycle. These log files are located in the alephe_scratch directory.

5.2.11 C.11 Display log list of batch jobs [10-5]
Function: Displays the list of batch jobs in edit mode
Activation: UTIL C/11

This procedure displays a list of jobs executed for the library, in edit mode. The list can be shortened, using editing procedures. The name of the log file of a batch procedure is written immediately preceding the start line of the procedure.

Exiting from the file returns the user to the UTIL C menu.
5.3 UTIL O - Oracle Management

O. Managing ORACLE

<table>
<thead>
<tr>
<th>0. Exit procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Activate Oracle Server</td>
</tr>
<tr>
<td>2. Close Oracle Server</td>
</tr>
<tr>
<td>3. Show running Oracle Server</td>
</tr>
<tr>
<td>4. Activate Oracle Listener</td>
</tr>
<tr>
<td>5. Close Oracle Listener</td>
</tr>
<tr>
<td>6. Show running Oracle Listener</td>
</tr>
<tr>
<td>7. Turning Archiving On</td>
</tr>
<tr>
<td>8. Turning Archiving Off</td>
</tr>
<tr>
<td>9. Show Archiving status</td>
</tr>
<tr>
<td>10. Start Server Manager session</td>
</tr>
<tr>
<td>11. Add file to Tablespace</td>
</tr>
<tr>
<td>12. Database Verification Utility</td>
</tr>
<tr>
<td>13. List of Database files</td>
</tr>
<tr>
<td>14. List Database Free/Used Space</td>
</tr>
<tr>
<td>15. Coalesce all contiguous free extents</td>
</tr>
<tr>
<td>16. Manage database links for distributed database</td>
</tr>
<tr>
<td>17. Resize Oracle data file</td>
</tr>
<tr>
<td>18. Oracle statistics</td>
</tr>
<tr>
<td>19. Manage Shared Pool</td>
</tr>
</tbody>
</table>

Please select [exit]:

General:

ALEPH500 is based on Oracle8 RDBMS.

In ALEPH every library has a separate root directory. Each ALEPH library directory contains information relevant for administrating the library.

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

Note: In order to work with ALEPH, the Oracle server and the Oracle listener must be active.

Note: ALEPH enables you to place Oracle tables on a separate server from the ALEPH application, and even to distribute Oracle tables across two or more servers. For more details see Chapter 6, "Library Utilities: Working With Remote Oracle Tables".
5.3.1 O.1 Activate Oracle Server

Function: Activates the Oracle server
Activation: UTIL O/1
Note: Requires ALEPH DBA username/password

In order for ALEPH to interact with Oracle, both the Oracle server and the Oracle listener should be running. They may be automatically started at boot time (this is determined during installation) or controlled by the ALEPH Oracle Management utilities.

If you chose to manage the Oracle database through ALEPH, the user ALEPH_DBA was created during installation. This user has the Database Administrator privileges, and can start up or shut down the database.

When you select **Activate Oracle Server** you will be prompted:

To continue you will need to enter ALEPH DBA username/password.
Username/password:

Enter the ALEPH DBA username and password (e.g. aleph_dba/aleph_dba).

5.3.2 O.2 Close Oracle Server

Function: De-activates the Oracle server
Activation: UTIL O/2
Note: Requires ALEPH DBA username/password

If you chose to manage the Oracle database through ALEPH, the user ALEPH_DBA was created during installation. This user has the Database Administrator privileges, and can start up or shut down the database.

This utility will shut down the Oracle server immediately by activating the Oracle shutdown immediate option. All the clients connected to the server will be logged out immediately.

When you select **Close Oracle Server** a question will appear:

Do you want to restart Oracle server after closing? yes/[no]

If you enter yes, the server will be shut down and restarted.
If you enter no, the server will be shut down and will not be restarted. In order to restart it you will need to select utility **O/1 Activate Oracle Server**.

After you enter yes or no and press <Enter> you will be prompted:

To close Oracle server enter ALEPH DBA username/password.
username/password:
Enter the ALEPH DBA username and password (e.g. aleph_dba/aleph_dba).

5.3.3 O.3 Show running Oracle Server

Function: Shows running Oracle server
Activation: UTIL O/3

This utility displays the Oracle server. The following lines will be displayed if Oracle is working properly:

<table>
<thead>
<tr>
<th>Process Name</th>
<th>User ID</th>
<th>Time</th>
<th>Date and Time</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>oracle 13454</td>
<td>1</td>
<td>0</td>
<td>May 21</td>
<td>0:01 ora_pmon_aleph0</td>
</tr>
<tr>
<td>oracle 13461</td>
<td>1</td>
<td>0</td>
<td>May 21</td>
<td>23:41 ora_dbwr_aleph0</td>
</tr>
<tr>
<td>oracle 13477</td>
<td>1</td>
<td>0</td>
<td>May 21</td>
<td>1:06 ora_smon_aleph0</td>
</tr>
<tr>
<td>oracle 13469</td>
<td>1</td>
<td>0</td>
<td>May 21</td>
<td>23:21 ora_lgwr_aleph0</td>
</tr>
<tr>
<td>oracle 13483</td>
<td>1</td>
<td>0</td>
<td>May 21</td>
<td>0:03 ora_reco_aleph0</td>
</tr>
</tbody>
</table>

Note: This utility is only relevant if you are running the Oracle server on the same node as the ALEPH server.

If these lines do not appear, the Oracle server may be activated using utility O/1.

5.3.4 O.4 Activate Oracle Listener

Function: Activates the Oracle listener
Activation: UTIL O/4
Note: Requires Oracle software owner password

When a user process makes a connection request, the Oracle listener process examines the request and connects it to a dedicated server process. In order for ALEPH to interact with Oracle, both the Oracle server and the Oracle listener should be running. They may be started automatically at boot time (this is determined during installation) or controlled by the ALEPH Oracle Management utilities.

When you select Activate Oracle Listener you will be prompted:

To continue you will need to enter Oracle's password.
Password:

Enter the Oracle password.
5.3.5 O.5 Close Oracle Listener

Function: De-activates the Oracle listener
 Activation: UTIL O/5
 Note: Requires Oracle software owner password

When you select Close Oracle listener you will be prompted:

To continue you will need to enter Oracle's password.
Password:

Enter the Oracle password and the listener will be closed.

5.3.6 O.6 Show running Oracle Listener

Function: Shows the running Oracle listener
 Activation: UTIL O/6

When you select Show running Oracle Listener, a line similar to the following, will be displayed:

803 /aleph/app/oracle/product/806/bin/tncslsnr LISTENER -inherit

If this line does not appear, the Oracle listener may be activated using utility O/4.

5.3.7 O.7 Turning Archiving On

Function: Turns Oracle archiving on
 Activation: UTIL O/7
 Note: Requires ALEPH DBA username/password

ALEPH backup and recovery procedures are based on Oracle. In order to have the complete ability to recover data up to the time of failure, the Oracle database mode should be ARCHIVELOG. This will ensure full recovery up to the time of failure. Hot backup cannot be done without ARCHIVELOG mode. This is why it is highly recommended to work in archiving mode.

There are some preliminary actions that need to be done before UTIL O/7 can be used. Please refer to the Backup manual for more info.

Changing the archiving mode shuts down the database. ALEPH processes (servers and batch procedures) will be stopped (using the aleph_shutdown_505 script in Salephe_root) and then re-started (using the aleph_startup_505 script in Salephe_root).

Note: When running utilities to rebuild the word or headings indexes, it is recommended to stop Oracle archiving as it will slow down the process, and fill up
the disk. After the process is finished you should turn archiving back on and make a full database backup.

5.3.8 O.8 Turning Archiving Off

Function: Turns Oracle archiving off  
Activation: UTIL O/8  
Note: Requires ALEPH DBA username/password  

This utility turns archiving off.

Note: Changing Archiving mode shuts down the database. ALEPH processes (servers and batch procedures) will be stopped (using the aleph_shutdown_505 script in $alephe_root) and then re-started (using the aleph_startup_505 script in $alephe_root).

5.3.9 O.9 Show Archiving status

Function: Displays archiving status  
Activation: UTIL O/9  
Note: Requires ALEPH DBA username/password  

This utility displays the archiving status. After entering your username/password you will see the following if archiving is off:

Oracle Server Manager Release 2.3.3.0.0 - Production  
Copyright (c) Oracle Corporation 1994, 1995. All rights reserved.

Oracle7 Server Release 8.0.6.0.0 - Production Release  
With the distributed option  
PL/SQL Release 2.3.3.6.0 - Production  
SVRMGR> Connected.  
SVRMGR> Database log mode No Archive Mode  
Automatic archival Disabled  
Archive destination /aleph1/oradata/aleph1/arch  
Oldest online log sequence 1380  
Current log sequence 1382  
SVRMGR>  
Server Manager complete.  
Enter CR to continue...

When archiving is on you will see that Database log mode is set to Archive Mode and Automatic archival is enabled.
5.3.10 O.10 Start Server Manager session
Function: Starts the server manager session
Activation: UTIL O/10
Note: Requires ALEPH DBA username/password

This utility starts an Oracle Server Manager session as sysdba. Enter the following:

SVRMGR> connect aleph_dba as sysdba;
Password: aleph_dba
Connected
SVRMGR>

5.3.11 O.11 Add file to Tablespace
Function: Adds a file to an existing tablespace
Activation: UTIL O/11
Note: Requires ALEPH DBA username/password

Tablespaces are composed of one or more data files. When a tablesapce does not have enough contiguous free space it needs to be enlarged. This may be done by adding new files or by resizing existing files (See UTIL O/17).

When you select Add file to Tablespace you will be prompted for the ALEPH DBA username/password. After you enter the username/password you will be prompted for the tablespace name. Enter the tablespace name and a list of the existent files will appear. For example:

Tablespace TS1 consists of the following files:

/aleph1/oradata/aleph1/ts1_0.dbf
/aleph1/oradata/aleph1/ts1_1.dbf
/aleph/oradata/aleph1/ts1_2.dbf

You will be prompted for the following parameters:

The new file name. Enter the complete path.
The file's size (in Megabytes).

Note: Different operating systems restrict the size of data files to 1000 or 2000 Megabytes.

Do you want the file to be auto extendable? (y or n). If you answer y the file will automatically grow when more space is needed.

Note: It is recommended to answer n. More space can be added as necessary using utilities O/11 and O/17.

The utility will list all of your choices and ask for confirmation. For example:
New file: /aleph/oradata/aleph1/ts1_3.dbf
Size: 1000 MB
Auto Extend: OFF
confirm (y/[n]):

Enter y and the file will be created and added to the tablespace.

5.3.12 O.12 Database Verification Utility
Function: Verifies the consistency of an Oracle data file
Activation: UTIL O/12
Note: Requires ALEPH DBA username/password

This procedure verifies that all the data files are fully readable and accessible. It is advisable to run it once a week for all database files.

In order to run this Oracle tool the database must first be shut down. See UTIL O/2.

When you select Database Verification Utility you will be prompted:

To continue you will need to enter ALEPH DBA username/password.
Username/password:

Enter the ALEPH DBA username and password (e.g. aleph_dba/aleph_dba).

5.3.13 O.13 List of Database files
Function: Lists Oracle data files
Activation: UTIL O/13

This utility lists the Oracle data files and their sizes. For example:

<table>
<thead>
<tr>
<th>NAME</th>
<th>SIZE K</th>
</tr>
</thead>
<tbody>
<tr>
<td>/aleph/oradata/aleph2/rbs01.dbf</td>
<td>51200</td>
</tr>
<tr>
<td>/aleph/oradata/aleph2/system01.dbf</td>
<td>81920</td>
</tr>
<tr>
<td>/aleph/oradata/aleph2/temp01.dbf</td>
<td>204800</td>
</tr>
<tr>
<td>/aleph/oradata/aleph2/tools01.dbf</td>
<td>25600</td>
</tr>
<tr>
<td>/aleph/oradata/aleph2/ts0_0.dbf</td>
<td>2048000</td>
</tr>
<tr>
<td>/aleph/oradata/aleph2/ts0_2.dbf</td>
<td>2048000</td>
</tr>
<tr>
<td>/aleph/oradata/aleph2/ts0_3.dbf</td>
<td>1024000</td>
</tr>
<tr>
<td>/aleph/oradata/aleph2/ts0_1.dbf</td>
<td>2048000</td>
</tr>
<tr>
<td>/aleph/oradata/aleph2/ts1_0.dbf</td>
<td>2048000</td>
</tr>
<tr>
<td>/aleph/oradata/aleph2/ts1_1.dbf</td>
<td>2048000</td>
</tr>
<tr>
<td>/aleph/oradata/aleph2/users01.dbf</td>
<td>5120</td>
</tr>
</tbody>
</table>
5.3.14 O.14 List Database free space

Function: Lists the free space in a database
Activation: UTIL O/14

This utility provides information about the tablespaces' free space. When you select List Database free space a submenu displays:

14. DataBase Free/Used Space Statistics
----------------------------------------
0. Exit procedure
1. All Tablespaces free space summary
2. Number of Free extents by size in a Tablespace
3. All Free extents of min size in a Tablespace
4. Space Used by a Library/Libraries in each Tablespace
5. Space Used by a Group of Libraries in each Tablespace

Please select [exit]:

Select the desired option:

All Tablespaces free space summary

This option provides details about the database free space in the Oracle DBA_FREE_SPACE table. There are four columns in the report:

1. TABLESPACE_NAME: The tablespace’s name.
2. TOTAL_FREE_SPACE: The total free space in the tablesapce (in Megabytes).
3. MAX_EXTENT: The largest contiguous extent of the tablespace (in Megabytes).
4. NUM_FREE_EXTENTS : The number of free extents the tablespace.

<table>
<thead>
<tr>
<th>TABLESPACE_NAME</th>
<th>TOTAL_FREE_SPACE</th>
<th>MAX_EXTENT</th>
<th>NUM_FREE_EXTENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSTEM</td>
<td>28.273343</td>
<td>28.273345</td>
<td>1</td>
</tr>
<tr>
<td>TEMP</td>
<td>200.09429</td>
<td>40.0141679</td>
<td>33</td>
</tr>
<tr>
<td>TOOLS</td>
<td>17.7753893</td>
<td>17.6972214</td>
<td>2</td>
</tr>
<tr>
<td>TS0</td>
<td>425.632244</td>
<td>118.549496</td>
<td>754</td>
</tr>
<tr>
<td>TS1</td>
<td>444.236214</td>
<td>147.565435</td>
<td>118</td>
</tr>
<tr>
<td>USERS</td>
<td>4.9949313</td>
<td>4.9949313</td>
<td>1</td>
</tr>
</tbody>
</table>

It is important to review this report from time to time in order to prepare additional resources for the database.

Note: If a tablespace has no free space left it will not appear in this report.

Note: If the largest contiguous extent is less than 20MB, you will probably have trouble enlarging Oracle tables in the database. This is because the Oracle server will
not be able to allocate any additional extents for the tables. In this case you should consider adding more space to the tablespace or reorganizing it. See utilities O/11 Add file to Tablespace, O/15 Coalesce all contiguous free extents and O/17 Resize Oracle data file.

**Number of Free extents by size in a Tablespace**

This option lists the number of extents of a certain size (truncated in Megabytes) in the tablespace. You will be prompted for the tablespace name.

For example:

Enter tablespace name: *ts0*

<table>
<thead>
<tr>
<th>SIZE IN MB</th>
<th>NUM OF EXTENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>879</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>4</td>
</tr>
</tbody>
</table>

There are five free extents in tablespace TS0. Four of them are less than a Megabyte and one of them is 879 megabytes.

**All Free extents of min size in a Tablespace**

This option lists exact size (in Megabytes) of all free extents that are larger than a given size. You will be prompted for the tablespace name and the minimum size (in Megabytes) desired.

For example:

Enter tablespace name: *ts0*

Enter Min size (MB) of free extent [0=ALL]: 0

<table>
<thead>
<tr>
<th>EXTENT_SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>879.428397</td>
</tr>
<tr>
<td>.156335878</td>
</tr>
<tr>
<td>.117251908</td>
</tr>
<tr>
<td>.117251908</td>
</tr>
</tbody>
</table>

Since the minimum size entered was zero, this example lists the exact sizes of all the free extents in TableSpace TS0.

**Space Used by a Library/Libraries in each Tablespace**

**Space Used by a Group of Libraries in each Tablespace**
5.3.15 O.15 Coalesce all contiguous free extents

Function: Coalesces all contiguous free extents
Activation: UTIL O/15
Note: Requires ALEPH DBA username/password

When all the database free space is composed of many small extents it is worthwhile to use this procedure to coalesce all the contiguous free extents into one large extent. It is highly recommended to perform this procedure at least on a weekly basis.

You will be prompted for the ALEPH DBA username/password.
Enter the ALEPH DBA username and password (e.g. aleph_dba/aleph_dba).

Note: The procedure only coalesces extents for tablespaces TS0 and TS1.

5.3.16 O.16 Manage database links for distributed database

Function: Manages database links for distributed database
Activation: UTIL O/16

The O/16 utilities are used to manage one logical database over the network from more than one physical database. They are used when Oracle tables are on one or more separate servers from the ALEPH application. When you select Manage database links for distributed database, the following sub-menu displays:

16. Manage Database Links
   --------------------------
   0. Exit procedure
   1. List database links
   2. Create database link
   3. Drop database link

Please select [exit]:

List database links

This option lists the existent database links. The list will be empty if there are no existing links.

Create database link

This option creates a new database link. You will be prompted:

Enter oracle TNS service name for remote database:

Enter the name of the network service <hostname>.<SID> as defined in the Oracle network configuration file.
SORACLE_HOME/network/admin/tnsnames.ora

**Note:** If the network service is not defined in the configuration file, an error message will appear and you will not be able to create the new link.

If the network service is defined, you will be prompted:

Enter username to remote system [aleph]
Enter password to remote system [aleph passwd]

The database link will be created and the following messages will appear (in our example the TNS service name is ram40.aleph0 and the username is aleph):

Now creating a private database link to remote user aleph,
If the remote database's aleph password is changed
in the remote location, then
this database link should be recreated!

drop database link ram40.aleph0

*  
ERROR at line 1:  
ORA-02024: database link not found

Database link created.

Enter CR to continue...

**Note:** This utility drops the link and then creates it. Therefore, if this is the first time a link is created the error message:

ERROR at line 1:  
ORA-02024: database link not found

will appear. This message should be ignored.

In ALEPH new database links are used to:

Create an empty Oracle table on a remote database (see Chapter 6: "Library Utilities: Working With Remote Oracle Tables", section 4: "Managing a Remote Table").

Move an Oracle table from a remote database to a local database (see Chapter 6: "Library Utilities: Working With Remote Oracle Tables", section 5: "Changing a Table from Remote to Local").

Move an Oracle table from a local database to a remote (see Chapter 6: "Library Utilities: Working With Remote Oracle Tables", section 4: "Managing a Remote Table").
Access an existing Oracle table on a remote database from a local database (see Chapter 6: "Library Utilities: Working With Remote Oracle Tables", section 2: "Accessing a Remote Table (Without Managing It)".

**Drop database link**

This option is used to drop a database link when it is not needed anymore. You will be prompted for the database link name to drop.

### 5.3.17 O.17 Resize Oracle data file

**Function:** Resize an existing Oracle data file  
**Activation:** UTIL O/17  
**Note:** Requires ALEPH DBA username/password  

Tablespaces are composed of one or more data files. When a tablespace does not have enough free space it needs to be enlarged. This may be done by resizing existing files or by adding new files (See UTIL O/11).

When you select **Resize Oracle data file** you will be prompted for the ALEPH DBA username/password. After you enter it you will be prompted for the tablespace name.

Enter the tablespace name and a list of the existent files will appear. For example:

Tablespace TS1 consists of the following files:

/aleph1/oradata/aleph1/ts1_0.dbf  
/aleph1/oradata/aleph1/ts1_1.dbf  
/aleph/oradata/aleph1/ts1_2.dbf

You will be prompted for the following parameters:

1. Enter file name to resize  
2. Enter new file size (in Megabytes)

The utility will list all of your choices and ask for confirmation. For example:

**Tablespace:** TS0  
**File:** /aleph/oradata/aleph1/ts0_4.dbf  
**New size:** 2000MB

**confirm** (y/[n]):

Enter y and the file will be resized.
5.3.18 O.18  Oracle statistics
Function: Displays Oracle statistics
Activation: UTIL O/18

This utility provides the following Oracle Statistics:

1. Performance Statistics
2. Rollback Segments definitions
3. Rollback Segments dynamic allocation

Enter the number of the desired option and the statistics will be displayed.

5.3.19 O.19  Manage Shared Pool
Function: Manage Shared Pool
Activation: UTIL O/19

When you select Manage Shared Pool, the following submenu displays:

19. Shared Pool
--------------------------------
0. Exit procedure
1. Show SGA buffers
2. Flush Shared Pool
Please select [exit]:


5.4 Working With Remote Oracle Tables

The most common network configuration is to have one server where both the ALEPH application and the Oracle tables are located. However, ALEPH also enables you to place Oracle tables on a separate server from the ALEPH application, and even to distribute Oracle tables across two or more servers.

This flexibility accommodates changing needs. For example, if you originally set up the system with both the ALEPH application and the Oracle data tables on the same server, then later, as the amount of data grows, you can add another server to the system and transfer some Oracle tables to the new (remote) server. You will be able to manage the remote tables from the local server.

If your library system has many libraries, one group of libraries may be located on one server and another group of libraries may be located on a second server. In this situation, you have two choices:

Keep the tables of each group separate. Place all of the tables of Group A on one server and all of the tables of Group B on another server. In this case, you can allow each group to access the Oracle tables of the other group, but to manage only the tables located on their own respective servers.

Place some tables of one group (e.g., Group A) on the server of the other group. In this case, you would allow Group A to manage tables that are located on the server of Group B.

This section covers the following topics:
1. Establishing Separate Servers for ALEPH Application and Oracle Tables
2. Accessing a Remote Table (Without Managing It)
3. Stopping Access to a Remote Table
4. Managing a Remote Table
5. Changing a Table from Remote to Local

5.4.1 Separate Servers for ALEPH Application and Oracle Tables

You can set up your system from the very beginning with the ALEPH application on one server and the Oracle tables on another server. To support this configuration, go to the /aleph/a50_5/alephe directory and open the alephe_start_505 file. Make sure that the following variables appear in that file:

setenv aleph_db <hostname>.<SID>
setenv TWO_TASK <hostname>.<SID>

The hostname is the name of the remote server and the SID is the name of the Oracle database on that server.
5.4.2 Accessing a Remote Table (Without Managing It)

You can access a remote table from the local server. The management of the table, including creation and load, will be accomplished from the remote server.

To support this arrangement, the table will be defined in the file_list of the remote server and will be removed from the file_list of the local server (to prevent managing it from the local server). A Remote Synonym for the table will be defined in the file_list of the local server.

1. Edit the file $ORACLE_HOME/network/admin/tnsnames.ora to add the service name you want to use.
   
   `<hostname>,<sid> = (description=
   (address=
   (protocol=tcp)
   (host=<hostname>)
   (port=<portnumber>))
   (connect_data=(sid=<sid>)))`

   There is no need to restart Oracle.

2. Use Util O/16 to create a database link for the remote service:
   
   Enter oracle TNS service name for remote database:
   `<hostname>,<SID>`
   Enter username to remote system [aleph] `<remote user name>`
   Enter password to remote system [aleph passwd]
   `<remote user password>`

   **Note:** "aleph password" will be the password of user aleph in the local system!

3. Define the table in the local file_list with the keyword RS (Remote Synonym):
   
   RS `<table name>` `<hostname>,<SID>`
   For example:
   RS Z31 ram40.aleph0

4. Use UTIL A/17/5 to recreate all library synonyms.

5. Check in your local host using sqlplus aleph/aleph that you can desc `<lib>,<table>` which is in the remote instance.

6. Check in your local host using UTIL F/4 that you can see the table from the remote instance.

5.4.3 Stopping Access to a Remote Table

The remote table was accessed from the local server but now we want to stop accessing it. The table is not defined in the file_list of our local server.

1. **Perform this step only if the table is not being used in the remote instance.** If the table does not appear in the remote file_list, then drop the table from the remote instance:
   
   `s+ <lib>`
drop table <table>;

2. Remove the 6th column (<hostname>,<sid>) from the local file_list.

3. Use UTIL A/17/5 to recreate all library synonyms.

4. Change TAB/RS to NA and delete all columns from col3 and on.
   Leave only NA <table_name> or totally remove the line from local
   file_list.

5.4.4 Managing a Remote Table

The table is located on a remote server. All management, including creation and load,
will be performed from the local server. Therefore, the table will be defined in the
file_list of the local server, but not defined in the file_list of the remote server (to
prevent managing it from there).

1. Optional. Save table data. Use UTIL B/4 to build a flat file from an
   existing table.

2. Edit the file $ORACLE_HOME/network/admin/tnsnames.ora to add
   the service name you want to use.
   <hostname>,<sid> = (description =
   (address =
      (protocol=tcp)
      (host=<hostname>)
      (port=<portnumber>))
   (connect_data=(sid=<sid>)))
   There is no need to restart Oracle.

3. Use UTIL O/16 to create a database link for the remote service:
   Enter oracle TNS service name for remote database:
   <hostname>,<SID>
   Enter username to remote system [aleph]: <remote user name>
   Enter password to remote system [aleph passwd]:
   <remote user password>
   Note: "aleph passwd" will be the password of user aleph in the local
   system!

4. Define the table in the local file_list using all columns as usual
   but adding a 6th column <hostname>,<SID>
   For example:
   TAB Z31 100K 100K TS0 ram40.aleph0

5. Use UTIL A/17/1 in your local host to create the table in the remote
   instance using your local file_list.

6. Optional. Upload table data. Use UTIL B/5 to upload a flat data file
   into a newly created table.

7. Optional (checks):

   Check that the table was created in the remote instance and not in the
   local:

   Enter to s+ <lib> and "desc <table>;", you will see the table when
doing
this in the remote host and will not see it doing it in the local host.
At this point you can NOT see the table doing

    sqlplus aleph/aleph
desc <lib>.<table>

in your local host.

8. Use UTIL A/17/5 to recreate all library synonyms.

9. Check in your local host using sqlplus aleph/aleph that you can
desc <lib>.<table> which is in the remote instance.

10. Check in your local host using UTIL F/4 that you can see the table
from the remote instance.

5.4.5 Changing a Table from Remote to Local
The table was used as a remote table, but now we want to create the table in our local instance.

1. Optional. Save table data. Use UTIL B/4 to build a flat file from an
existing table.

2. Perform this step only if the table is not being used in the remote
instance.
   If the table does not appear in the remote file_list, then drop the table
   from the remote instance:

       s+ <lib>
       Drop table <table>;

3. Remove the column <host>.<sid> from the local file_list, if RS
change to TAB and add needed columns for TAB option.
4. Use UTIL A/17/5 to recreate all library synonyms.
5. Use UTIL A/17.1 in your local host to create the table in your local
host.
6. Optional. Upload table data. Use UTIL B/5 to upload a flat data file
into a newly created table.
7. Check in your local host using sqlplus aleph/aleph that you can
desc <lib>.<table>.
8. Check in your local host using Util F/4 that you can see the table.
These utilities delete scratch files from a number of directories in the Unix server. Before running the utilities be certain to close all ALEPH servers, batch queues and daemons.
6  ALEPH Servers, Batch Jobs, and Problem Diagnosis

There are nine forms of ALEPH activity, nine ways in which work is accomplished in ALEPH:

1. PC client/PC server [section 6.4]
2. ue_daemons [6.5]
3. Batch jobs [6.6]
4. www_server (public) [6.7]
5. www_server (staff) [6.8]
6. Z39 servers [6.9]
7. OCLC server [6.10]
8. sc_server (self-check server) [6.11]
9. SQL [6.12]

Each is discussed below. First we have three sections with information about servers in general: the ALEPH UTIL W menu, Server logs, and Starting your own server.

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

Function: Server Management
Activation: UTIL W

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

<table>
<thead>
<tr>
<th>W. Server Management (WWW,PC,Z39)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0. Exit procedure</td>
</tr>
<tr>
<td>1. Monitor servers</td>
</tr>
<tr>
<td>2. Stop servers</td>
</tr>
<tr>
<td>3. Start servers</td>
</tr>
<tr>
<td>4. View log file</td>
</tr>
</tbody>
</table>

Please select [exit]:

Conventional port numbers are:

- PC server (GUI) 6505
- WWW server (Web) 4506,4507..
- WWW front (Multi Web servers) 4505
- Z39 server 9909
- Z39 back end server (complementary to Z39 server) 9908
- Z39 gate 9907
- Generic server (OCLC server) 7505
- Self-check server 8505
Configuration of the www_server is discussed in the UTIL J/1, J/2, and J6 sections of the Database Management Guide; configuration of the pc_server in the UTIL J5 section.

### 6.2 Server Logs

The logs of the PC, Z39.50, OCLC, and self-check servers are written to the TEMPDIR directory.

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

The logs of the Apache server (see section 6.7, www_servers, below) are written to ../alephe/apache/logs.

To view the logs of the most recent or currently running versions of the servers, do “ls –lr”. The “tail” command gives you the most recent activity for that particular server.

The names are:

- `pc_server_<port>.log`
- `www_server_<port>.log`
- `www_front_<port>.log`
- `z39_server_<port>.log`
- `z39_back_end_server_<port>.log`
- `z39_gate_<port>.log`

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:

```
```

### 6.3 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 your transactions are not mixed in with those for other transactions.)

For example:

Start your own 4599 and point your Web browser to:

```
http://xxxx.xxxx.xxx:4599/ALEPH
```


In another example, for the GUI:

Start a 6599 and specify 6599 as the address in your PC’s
\ALEPHCOM\TAB library.ini file, instead of 6505.

To start a server in this fashion, enter (in our example):
>>pc_server 6599 stdout
**Note:** for the WWW server, you need to enter two numbers:
>>www_server 4599 4599 stdout

You are telling the system that you want the input server to be 4599 and the output server to be 4599 (unlike the usual situation where 4505 would be the input and 4507 might be the output). The “stdout” specifies that you want the server output (log) to be displayed on your display terminal.

### 6.4 PC client/PC server

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

Error messages which you see may have been generated by the PC client or they may have been passed 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 rep.change 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. (For specific errors and how to deal with them, consult the WebPRB OPAC of the Ex Libris office which supports your site.)

Here's how the PC and the PC-server interact:

#### 6.4.1 Connecting

When you start the first GUI module on your PC, if you are not already logged on, the software prompts you for a user name and password. It checks all the servers listed in the ..\ALEPHCOM\DAT\library.ini file. If this name/password isn't valid on any of the servers it can connect to, you get the message "Password not verified on connectable hosts". This indicates either that this name/password is not valid OR that the address specified in the ..\ALEPHCOM\DAT\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 database. (Click on "File" in the top bar and then "Connect"). When you click on "Connect" the system displays the libraries listed in the module's per_lib.ini file (for example, ..\CIRC\TAB\per_lib.ini). (The only exception is the Sear [OPAC] client which uses the "Options", "Database" values.)

When you click on a particular library (or base), the client software goes to the ..\ALEPHCOM\DAT\library.ini and tries to connect to the IP address specified for this library.
6.4.2 Service Requests

[Note: The ALEPH Web Guide, accessible at http://......45n5/S, documents each GUI module's functions.]

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, etc., 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 which looks like this:

```
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's 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 were 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’s no line with text for the 0021 error in the alephe/error_eng/pc_cat_c0204 file when there should be.)

Two general tables which are important to all GUI modules are the xxx01/tab/pc_tab_col.lng and the xxxxn/tab/pc_tab_exp_field.lng. The former is used to produce all columnar displays you see in the GUI; the latter, to specify the data which appears in dropdown choices in the GUI.

The bibliographic information which displays on different screens in the GUI is built by various methods. Please consult the document "How To Change Bib Info Displays" for details.

PC functions which print slips, labels, etc., will call a program in the ../asm/source/form_fill directory to read the form from the ../xxx01/form_lng directory. For more information on forms, see the document "How To Change Printed Products" and the "Template" sections in the ALEPH Web Guide.
6.5 Analyzing PC Server Problems

There is normally just a single PC server (6505). Multiple logs for this server (for example, “pc_server_6505.log.2142”) in a time period when the server was not being stopped intentionally (for example, for a backup) indicate a server failure. To get more information than what you see in the pc_server_\(nnnn\).log file do:

\[\text{>>pc_server view 6505 xx}\]

where “xx” is the number of transactions from the end that you want to include. (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:

\[\text{>>pc_server check 6505 55459}\]

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

\[\text{>>pc_server checkx 6505 55459}\]

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

An 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 catalog GUI I the error message “Failed to read reply” appears. This means that the server could not handle the abnormal situation.

Type: 
M505>> pc_server view 6505 10

And you will see:

.................
53 OUT 710 Mon 08-02-1999 10:10:44 192.115.235.67 192115235067
C0201 505121 USM01ENGYOHANAN 000000
Thu Jan 14 11:50:43 1999
/aleph/a50_5/usm01/pc_tab/catalog/pc_cat.pck
54 IN 710 Mon 08-02-1999 11:57:37 192.115.235.111 192115235111
C0201 USM01ENGYOHANAN 000000

This reveals that transaction number 54 (C0201 is Catalog connect) has no OUT response.

Therefore you can use:
M505>> pc_server check 6505 54

And you will see:
Now it is obvious that UTIL M/7 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 the Ex Libris support groups. This can be done with the following sequence of commands:

```
M505>> cd $TMPDIR
M505>> tail -10 pc_ser_6505 > my_transaction_file
```

### 6.5.1 PC Server Time-out

There may be cases where you will want to adjust the PC_SERVER_TIMEOUT. If you are frequently reaching the maximum number of Oracle connections and the log shows that the transactions using the Oracle connections are primarily PC transactions, you may want to decrease the PC_SERVER_TIMEOUT value to make them time-out more quickly. This timeout value is set in the `pc_server_defaults` in UTIL L/1.

**Note:** When a PC transaction times out, this does not mean that the user needs to reconnect by clicking on Connect, selecting a database, etc. The software will automatically reconnect him.

The distributed value for PC_SERVER_TIMEOUT is 400 (seconds). Some sites have found that decreasing the value to 200 (seconds) works better for them. Sites with large processors may want to increase the number of connections (semaphores) instead. This is done by increasing the number of processes in `initaleph0.ora` (in the `/aleph/ora_aleph` directory). For example:

```
processes = 200
open_cursors = 100
```

may be increased to:
```
processes = 250
open_cursors = 200
```
6.6 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
- `ue - 14/15` job daemon

Each is described in detail in the UTIL E section of the *Database Management Guide*.

6.7 Batch jobs

[Note: For detailed information on batch jobs which build indexes, please consult the document, "How To Run Index Jobs".]

Batch jobs execute either daily, weekly, etc., or on-request. They may be initiated via the Web Services, the UTIL E 14/15 job list, or a "chron job".

6.7.1 Web Services

The WWW Staff Menu (at http://........:45n5/S ) has a "Services" option. 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/www_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 screen, say, the b-cir-51 overdue summary job screen, the system calls the corresponding www_b program --in the case of cir-51 this is the ../www_b/www_b_cir_51 -- to process the screen. If errors are found in the parameters, it will display a message. Otherwise, it says, "Job submitted to batch queue."

The job will be then be placed in the UTIL C/7 queue (jobs waiting for execution) of the library for which it was submitted. If the library's batch queue (see the UTIL C/2, section 5.2.2, above) is running, then the job will be executed --and will show up under UTIL C/1 (library running processes) as being executed.

The job will appear in the UTIL C/1 with its parameters:

```
aleph 31696 59116  43 18:24:25 pts/12  0:00 csh -f /aleph/a52_5/aleph/proc/p_cir_51 EXU50,ovrdustafN,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,14,00,Y,Y,N,O,4,0,0,0,
```
The “p_cir_51” component is the procedure which is going to be executed. The procedures are found in the aleph/proc directory. The ../aleph/proc/p_cir_51 executes three different programs:

cobrun b_cir_51_a  
cobrun b_cir_51_b  
cobrun b_cir_51_e

All compiled, executable programs are in the ../aleph/exe directory. The source for the batch programs called by the aleph/proc procedures is in the ..alephm/source/butil directory. In this case we have butil/b_cir_51_a, b_cir_51_b, b_cir_51_e, and b_cir_51_get_param. The b_xxx_nn_a programs use the ...get_param program to parse the parameter string into the separate variables. (In this case that's b_cir_51_get_param.)

Batch jobs which print columnar reports will use the xxx01/tab/rep_col.lng to format the columns.

Batch jobs which print notices, slips, other non-columnar forms will call a program in the ../asm/source/form_fill directory to read the form from the ../xxx01/form_lng directory. In this case b_cir_51_b calls the form_fill/overdue_summary program which reads the xxx01/form_eng/overdue_summary-xx-yy form. (For more information on forms, see the document "How To Change Printed Products" and the "Template" sections in the ALEPH Web Guide.)

[Note: For detailed information on batch jobs which build indexes, please consult the document, "How To Run Index Jobs".]

Any printed output produced by the batch job will be in the library's print directory (for example, the ../usm50/print directory). 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 (for example, usm50_p_cir_51.00138).

Job List
Jobs which need to run on a daily, weekly, etc., basis can be placed in the UTIL E 14/15 Job List, where they are executed by the ALEPH job daemon. You would take the parameters as they appear in the UTIL C/1 display of library-running-processes and paste them into a job log entry, for example,

W1 08:15:00  N csh -f /aleph/a52_5/aleph/proc/p_cir_51  
EXU50 Orrustaf,N,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,,00,14,00,Y,Y,N,O ,4,0,00,

If the log name (position # 4) is blank, as it is above, then the log will appear in the alephe/scratch under the name "csh.nnnnn" (where "nnnnn" is the process number the job had as it was executing.)

For more information on job list entries, consult sections UTIL E/14 and UTIL E/15 of the Database Management Guide.
6.7.2 Chron Jobs

"Chron jobs" are similar in principle to the ALEPH job list described in the preceding section, except that their control and maintenance is outside of ALEPH.

6.8 www_server (Public)

[The www server for Web OPAC is accessed at http://......:45n5/ALEPH. For example: http://aleph02.exlibris-usa.com:4525/ALEPH. (In most cases the port will be “4505”).]

In 14.2 and higher the 45n5 front server is replaced by the Apache HTTP server, whose files are in the ./alephe/apache directory. Please consult the document, “How To Install and Configure the Apache Server”, found in the How-to’s under the “Documentation” option in the Web Staff menu (at http://......:45n5/S).

UTIL J is used for other www_server configuration, including:

1. Messages to user (www_heading) [text of OPAC messages]
2. Sort types for display in OPAC (www_sort_heading)
   ...
6. Definition of server defaults (www_server_defaults)

See the "UTIL J" chapter in the Data Management Guide for details.

The www public screens are in the ../alephe/www_a_lng directory (unless directed elsewhere in the ../alephe/tab/path_convert). The document “List of HTML Files for the Web OPAC Interface” has descriptions of each www_a_lng file and the document “Web Customization” (in the Documentation CD /Functional Training/OPAC folder) has info on how to customize the screens.

6.9 www_server (Staff)

[The www_server staff menu is accessed at the address http://......:45n5/S. For example, http://aleph02.exlibris-usa.com:4525/S. (In most cases the port will be “4505”).]

Step 2: Choose module:
Services - Produce reports & manage files
Custom Services - Privately produced reports & management utilities
Guide - Read help on how to use the ALEPH system
OPAC - On-line Public Access Catalog [No Frames Version]
Course Reading List - Reading lists for courses
Release Notes - Read about changes since the last release
Documentation - ALEPH Documentation
Utilities - Online utilities

The "Services" are the WWW Batch Services described above in section 6.6.1.
The "Guide" is the Web Guide. It contains detailed description of the ALEPH GUI functions. It also includes the WWW OPAC and staff Services and Course Reading functions.

The "Course Reading List" is 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 the ../alephe/tab/path_convert).]

"Release Notes" is notes on ALEPH releases.

"Documentation" has documents in PDF: the Database Management Guide, this System Administration Guide, all the How-To documents, etc.

"Utilities" has the User Profiles, etc.

6.10 Z39 Servers
UTIL N is used for Z39 (z39.50) gate and server configuration and is discussed in the Data Management Guide "UTIL N" chapter. A separate document "ALEPH and Z39.50" (Z39.doc) has additional information.

6.11 OCLC Servers
OCLC records can be dynamically loaded into ALEPH from a PC connected to OCLC with the OCLC Passport software. Please consult the document, “How To Load OCLC Records into ALEPH”.

6.12 sc_server (Self-Check server)
The self-check server receives commands from / sends commands to a Circulation self-check device (such as the 3M self-check devices). Patrons use these devices to check books out and renew them, without staff intervention.

The sc_server programs are in the ../alephm/source/sc_server and ../pc_selfserv directories.
6.13 SQL

ALEPH sites can use native SQL or such report-writing tools as Microsoft Access to query and update the ALEPH znn Oracle tables directly. Descriptions of the znn tables can be found in the Oracle_Tables.zip in the WWW Staff menu Documentation option (see section 6.8 above).

The "s+ xxxnn" command can be used initiate SQL. For example, "s+ usm50".

The document “ALEPH_SQL” has common SQL commands with examples for querying ALEPH Oracle tables.

Note: the jobs which appear in the 14.2 WWW Services xxx50 libraries’ Circulation Statistics menu execute SQL scripts. These scripts are in the aleph/proc directory with the names “p_util_s_01_01” through “p_util_s_01_26”. (The proc for the www_b_lng screen b-stat-01 is p_util_s_01_01; that for b-stat-02 is p_util_s_01_02; etc.)