Aleph 23 System Administration Guide

Version 23

CONFIDENTIAL INFORMATION

The information herein is the property of Ex Libris Ltd. or its affiliates and any misuse or abuse will result in economic loss. DO NOT COPY UNLESS YOU HAVE BEEN GIVEN SPECIFIC WRITTEN AUTHORIZATION FROM EX LIBRIS LTD.

This document is provided for limited and restricted purposes in accordance with a binding contract with Ex Libris Ltd. or an affiliate. The information herein includes trade secrets and is confidential.

DISCLAIMER

The information in this document will be subject to periodic change and updating. Please confirm that you have the most current documentation. There are no warranties of any kind, express or implied, provided in this documentation, other than those expressly agreed upon in the applicable Ex Libris contract. This information is provided AS IS. Unless otherwise agreed, Ex Libris shall not be liable for any damages for use of this document, including, without limitation, consequential, punitive, indirect or direct damages.

Any references in this document to third-party material (including third-party Web sites) are provided for convenience only and do not in any manner serve as an endorsement of that third-party material or those Web sites. The third-party materials are not part of the materials for this Ex Libris product and Ex Libris has no liability for such materials.

TRADEMARKS

"Ex Libris," the Ex Libris bridge , Primo, Aleph, Alephino, Voyager, SFX, MetaLib, Verde, DigiTool, Preservation, URM, Voyager, ENCompass, Endeavor eZConnect, WebVoyage, Citation Server, LinkFinder and LinkFinder Plus, and other marks are trademarks or registered trademarks of Ex Libris Ltd. or its affiliates.

The absence of a name or logo in this list does not constitute a waiver of any and all intellectual property rights that Ex Libris Ltd. or its affiliates have established in any of its products, features, or service names or logos.

Trademarks of various third-party products, which may include the following, are referenced in this documentation. Ex Libris does not claim any rights in these trademarks. Use of these marks does not imply endorsement by Ex Libris of these third-party products, or endorsement by these third parties of Ex Libris products.

Oracle is a registered trademark of Oracle Corporation.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Ltd.

Microsoft, the Microsoft logo, MS, MS-DOS, Microsoft PowerPoint, Visual Basic, Visual C++, Win32,

Microsoft Windows, the Windows logo, Microsoft Notepad, Microsoft Windows Explorer, Microsoft Internet Explorer, and Windows NT are registered trademarks and ActiveX is a trademark of the Microsoft Corporation in the United States and/or other countries.

Unicode and the Unicode logo are registered trademarks of Unicode, Inc.

Google is a registered trademark of Google, Inc.

Document February 23, 2020

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

Table of Contents

1. ALEPH / Oracle

1.1. Introduction

ALEPH 500 is based on the **Oracle** RDBMS (relational database management system).

A typical ALEPH installation includes a number of ALEPH "libraries", where each library has its own set of data definitions (configuration tables) and data tables (Oracle tables). The different libraries represent either different types of information (bibliographic, administrative, holdings, authority), or separate libraries in the real world.

Typically, one real library will have at least four ALEPH libraries (bibliographic, administrative, holdings, authority). However, several real libraries might share one bibliographic library, and conversely, one bibliographic library might serve multiple administrative libraries.

1.2. Structure

Each ALEPH library is implemented by:

- A separate Oracle user; each Oracle user owns a set of tables which contain the ALEPH library's data.
- A separate directory tree, beginning from a root directory for the ALEPH library, which contains configuration tables, scratch files, print files, etc.

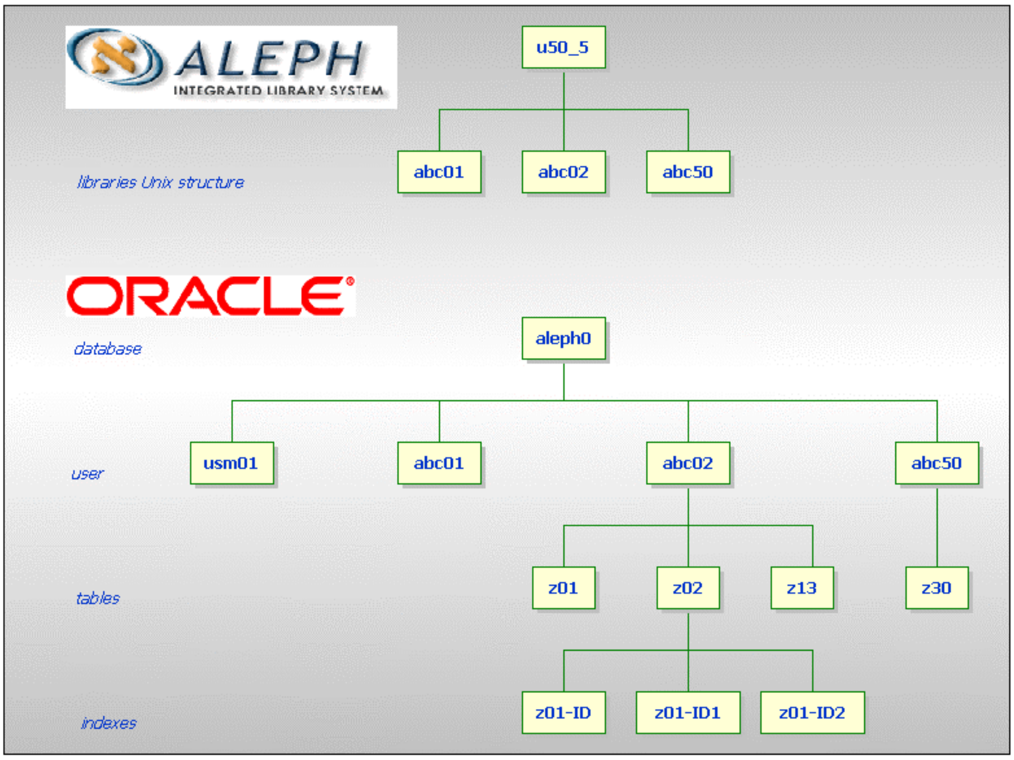

Figure 1

There are a number of types of libraries in ALEPH. Each library is identified by a code, made up of three characters followed by two digits. The digits identify the library type (following the Ex Libris scheme of digits is a naming convention, and not a system requirement).

- A Bibliographic (BIB) Library contains bibliographic records. It is identified by a number between 01-09 (for example, USM01).

  The Bibliographic Library is the search database. One site can have a single or multiple Bibliographic Libraries. If the Bibliographic Library includes multiple records for the same title, the ALEPH Union View can be used to present a single record to the OPAC user. A single Bibliographic Library database can be made to appear as if it consists of separate databases, using logical bases. Separate databases can solve problems of differing record structure (fields and tags), differing authority control, historically separate databases which cannot be merged, and so on. It is advisable to use one Bibliographic Library, unless there are good reasons to do otherwise.

- An Administrative (ADM) Library contains data about acquisitions, circulation, library staff, and patrons. It is identified by a number between 50-59 (for example, NDU50).

  Several Administrative Libraries can share a common Bibliographic Library. The basic patron record (name and address) is held in common. There can be separate patron privilege records, grouped at different levels.

- A Holdings (HOL) Library contains location and summary holdings information. It can also hold site-specific fields such as local subjects. It is identified by a number between 60-69 (for example, MAB60).

- An Authority (AUT) Library contains authority records of preferred forms of headings, relationships between headings (thesauri). It is identified by a number between 10-19 (for example, UNI11).

- An Interlibrary Loan (ILL) Library contains copies of the bibliographic records that are being requested by ILL. It is identified by a number between 20-29 (for example, UNI20).

- The Course Reading (CR) Library is a Bibliographic Library containing records of materials kept for the Course Reading (or Course Reserve) list. It is identified by a number between 30-39 (for example, TUD30).

1.2.1. SQL Access to the Oracle Tables:

In ALEPH, SQL *Plus can be used to access ALEPH’s Oracle tables.

```
il-aleph12.exlibris-int.il-a18(1) >>sqlplus usm50

SQL*Plus: Release 11.2.0.2.0 - Production on Thu Feb 5 11:28:44 2009

Copyright (c) 1982, 2008, Oracle.  All rights reserved.

Enter password:

Connected to:
Oracle Database 10g Enterprise Edition Release 11.2.0.2.0 -
Production
With the Partitioning, OLAP and Data Mining Scoring Engine options

usm50@ALEPH1> desc z72
 Name                          Null?    Type
 ----------------------------- -------- ----------------------------
 Z72_REC_KEY                   NOT NULL CHAR(21)
 Z72_ALPHA                              CHAR(1)
 Z72_VENDOR_ADDRESS                     VARCHAR2(500)
 Z72_VENDOR_TEL                         VARCHAR2(20)
 Z72_VENDOR_FAX                         VARCHAR2(20)
 Z72_VENDOR_EM                          VARCHAR2(60)
 Z72_VENDOR_IP                          VARCHAR2(60)
 Z72_VENDOR_CITY                        VARCHAR2(50)
 Z72_VENDOR_STATE                       VARCHAR2(50)
 Z72_VENDOR_POSTAL_CODE                 VARCHAR2(20)
 Z72_VENDOR_COUNTRY                     VARCHAR2(50)
 Z72_NOTE                               VARCHAR2(200)
```

1.2.2. Oracle Users in ALEPH

Each ALEPH library is implemented as an Oracle user. In addition, there are several Oracle users used by the ALEPH application, which are not related to a specific library.

- **ALEPH**
  The ALEPH server connects to the Oracle databases through a special Oracle user named **ALEPH** (default password: ALEPH). The ALEPH user can select, insert, update and delete data from the tables of all Oracle users (for example, ABC01, ABC50, ABC60, etc.), but is not the owner of any table.

- **ALEPH_ADMIN**
  This is an administrative user. This is a more privileged user, who, in addition to the privileges of the ALEPH user, can create, drop and alter Oracle tables, indexes, users, triggers, and so on. The ALEPH_ADMIN Oracle user is used for these purposes in all ALEPH procedures.

- **ALEPH_DBA**
  The third and last administrative Oracle user for ALEPH is **ALEPH_DBA**. This is the most privileged Oracle administrative user. It is used by ALEPH utilities to start up, shut down, and perform other DBA operations.

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).

1.2.3. Passwords

ALEPH contains an encrypted file with the passwords of the Oracle users used by ALEPH (for example ALEPH, ALEPH_ADMIN, ALEPH_DBA and ABC01). This means in effect that whenever you decide to change the password of an Oracle user, the password must be changed both in the Oracle database and in the ALEPH password file.

You can do this using **UTIL Y/8/1 Update Password for User**:

```
Enter Number [0] 1
Enter User Name:aleph
Enter New Password:aleph
Do you want to update this password in ALEPH Password file ([n]/y)? y
The password for ALEPH was saved in ALEPH password file
If you want to update this password in Oracle
Enter ALEPH_DBA user/passwd, or press [Enter] to exit
:ALEPH_DBA/<password>
Change passwd in Oracle

SQL*Plus: Release 11.2.0.2.0 - Production on Mon Feb 20 12:38:03 2006

Copyright (c) 1982, 2005, Oracle.  All rights reserved.
```

```
idle> Connected.
idle> idle>
User altered.

idle> Disconnected from Oracle Database 10g Enterprise Edition
Release 11.2.0.2.0 - Production
With the Partitioning, OLAP and Data Mining Scoring Engine options
```

1.3. Oracle Concepts

1.3.1. Storage

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

Tablespace Name	Usage	Physical File
System	Oracle system tables	/exlibris/oradata/aleph1/aleph1_system01.dbf
temp	Temporary space (for sorting, index creation, etc...)	/exlibris2/oradata/aleph1/aleph1_temp01.dbf
ts0	ALEPH DEMO tables	/exlibris2/oradata/aleph1/aleph1_ts0_0.dbf
ts1	ALEPH DEMO indexes	/exlibris/oradata/aleph1/aleph1_ts1_0.dbf
ts1d	ALEPH customer’s tables	/exlibris2/oradata/aleph1/aleph1_ts1d_01.dbf
ts1x	ALEPH customer’s indexes	/exlibris/oradata/aleph1/aleph1_ts1x_01.dbf

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**. See more information in Oracle Tables Management - file_list on page12.

1.3.2. 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.

1.3.3. 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 specifications in its CREATE TABLE command. The table mapping to a tablespace, and its initial space allocation are controlled by the **file_list**. The size of additional extent allocation also appears in the **file_list** for reasons of backward compatibility. See more information in Oracle Tables Management - file_list on page12.

1.3.4. 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. The index mapping to a tablespace, and its initial space allocation are controlled by the **file_list**. The size of additional extent allocation also appears in the **file_list** for reasons of backward compatibility. See more information in Oracle Tables Management - file_list on page12.

1.3.5. 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 on the table. The trigger can be executed before or after the SQL statement is issued on the table. Once the trigger has been created it can be enabled (will be executed automatically in case of a specific event) or disabled (is defined but will not be executed).

1.4. Oracle Tables Management - file_list

1.4.1. Introduction to Locally Managed Tablespaces

The data base for ALEPH 18 has Locally Managed Tablespaces.

Locally Managed Tablespaces
There are two types of extent allocation when using Locally Managed Tablespaces.

Auto Allocate
Oracle takes full control, automatically allocating extents as needed and taking into account the initial allocation of the table/index as supplied in the create table/index command.

Example: Initial allocation of the table/index as defined in the **file_list** is 1GB. Oracle might split the 1GB to 50 extents, 1 extent or any other combination.

In a standard ALEPH installation, tso and ts1 are created as Locally Managed Tablespaces with the auto allocate allocation type. Additional tablespaces, such as ts1d – ts4d (for tables) and ts1x – ts4x (for indexes) which are used for customer tables and indexes, are created as Locally Managed Tablespaces with the uniform size allocation type. These tablespaces use four different uniform sizes in order to accommodate small, medium, large and huge tables/indexes.

uniform
When creating the tablespace, the DBA determines the uniform extent size for all the extents in the tablespace. Each extent will be of that size. The DBA determines which table will be assigned to which tablespace depending on the table (Znn) size. All the extents of a table created in a Locally Managed Tablespace with uniform size will have the same size. This size is the uniform size defined for the tablespace, with no regard for the extents definition that may have been given in the Create Table command. In this way, there is no fragmentation and the utilization is optimal.

Example: When creating a tablespace with a uniform extent size of 10MB and a table that is 50MB, 5 extents will be used.

When working with Locally Managed Tablespaces, the word **LOCAL** appears in the **EXT-MGMT** (extent management) column in **UTIL O/17/4 Show Tablespace Definitions**. For example:

```
TS_NAME     EXT_MGMT    ALLOC_TYP    INIT_EXT    NEXT_EXT  TYPE  STAT
----------  ----------  ---------  ----------  ----------  ----  ----
LOG         LOCAL       SYSTEM          65536              PERM  ONL
SYSAUX      LOCAL       SYSTEM          65536              PERM  ONL
SYSTEM      LOCAL       SYSTEM          65536              PERM  ONL
TEMP        LOCAL       UNIFORM       1048576     1048576  TEMP  ONL
TS0         LOCAL       SYSTEM          65536              PERM  ONL
TS1         LOCAL       SYSTEM          65536              PERM  ONL
TSLOB       LOCAL       UNIFORM       8388608     8388608  PERM  ONL
UNDOTBS1    LOCAL       SYSTEM          65536              UNDO  ONL
USERS       LOCAL       UNIFORM         40960       40960  PERM  ONL
```

In the **ALLOC_TYP** column, you may see these values:

SYSTEM = auto allocate
UNIFORM = uniform

1.4.2. The role of file_list when working with Locally Managed Tablespaces

This is the mechanism for defining which table will sit in which tablespace. If a table has too many extents (**UTIL A/17/11 Space Utilization**), this means the table was assigned to the "wrong" tablespace. In this case, you can consider reorganizing the table. This entails performing a sequential dump, dropping the table, changing **file_list,** loading the table's sequential file and recreating the table's indexes.

In **file_list**, each table is mapped to a tablespace. When the table is created several extents will be allocated in order to match the initial allocation size specified in the **file_list**. The number of extents will vary.

- **Auto Allocate** - Oracle determines the extents' size and number.
- **Uniform** – each extent will be the size defined for the tablespace as the default extent size. Initially, the number of extents will be the number needed in order to get to the initial allocation size given in the **file_list** for that table/index.

By default, demo libraries use Auto Allocate and customer libraries use Uniform.

ALEPH comes with built-in `file_list` templates for each type of library. It consists of parameters used to manage all Oracle objects (**table**, **index**, **synonym**, and so on) of the particular library.

To override values from the template with your own values, you need to edit the `file_list` located in the root of each library. Use util a/17/10/1 for this purpose. To view the template `file_list` and/or to view the merged `file_list`, use util a/17/10/2. The definitions that determine the library type and size can be found in the `prof_library` file in the root of the library.

The first column is the type of object being defined. The content of the other columns depends on the type in column one.

- **TAB** table name initial allocation next allocation* tablespace name
- **IND** index_name initial allocation next allocation * tablespace name
- **TRI** trigger name
- **LS** table name library name (to link to)
- **NA** table name(not applicable in current library)
- **SEQ** sequence name
- **RS** table name library name alias name**
- **INX*** ** index_name initial allocation next allocation * tablespace name

* For locally managed tablespaces this column is not taken into account and can be defined as 0 KB. It appears for backward compatibility reasons only.

** The alias name that appears in the `tnsnames.ora` file. See Working With Remote Oracle Tables on page 59 for more details.

*** The INX is for the use of ADAM

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

```
TAB   z01             128K            0K              ts0
IND   z01_id          128K            0K              ts1
IND   z01_id2         128K            0K              ts1
TRI   z34_trigger
LS    z303              usm50
NA    z30
SEQ   last_result_set
```

In ALEPH, it is possible to work with some of the tables as remote tables, that is, tables that are in a different database on a different server. When working with remote tables, part of the **file_list** might look like this:

```
TAB   z02               1M          0K    ts0         ram36.aleph1
IND   z02_id            1M          0K    ts1         ram36.aleph1
IND   z02_id1           1M          0K    ts1         ram36.aleph1
RS    z98               usm01       ram36.aleph1
```

Notes:

In the Z02 table, the table and indexes are physically located on a different node (`ram36.aleph1`), but can be referenced and maintained on the local node as well.

In the Z98 table, the table and indexes are physically located on a different node (`ram36.aleph1`), but can be referenced (although they cannot be managed) on the local node as well.

When working with remote tables, an alias to the database on the different node (such as `ram36.aleph1`) must be defined in the file:

```
$ORACLE_HOME/network/admin/tnsnames.ora
```

For example:

```
      ram36.aleph1=(description=
                  (address=
                        (protocol=tcp)
                        (host=ram36)
                        (port=1521))
                  (connect_data=(sid=aleph1)))
```

For more information see also sections *Working With Remote Oracle Tables* on page 59 and *UTIL O - Oracle Management* on page 29.

2. Library Utilities

Each library's data is stored within Oracle as a separate Oracle user. Each Oracle user owns a set of tables which contain the library's data. There are various ALEPH online utilities that can be run in a library. Each utility deals with a different subject. For example, UTIL A File Administration and Building deals with the various library objects (tables, indexes, and so on) in the Oracle database.

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

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

2.1. UTIL A - File Administration and Building

```
   A. File Administration and Building for USM01
   ----------------------------------------------------
        0. Exit Procedure
        8. List Analyzed Tables / Indexes
        9. Delete Statistics for Analyzed Tables / Indexes
       12. Drop Cache Table (z50)
       13. Drop Statistics Table (z34)
       14. Drop Update Doc Table (z07)
       15.
       16. Drop Event Table (z35)
       17. Manage Oracle Tables
       18. Drop Web My e-Shelf guest Table (z129)
       19. Export Library
       20. Import Library
       21. List Objects
Please select [exit]:
```

UTIL A/8 List Analyzed Tables / Indexes

ALEPH works within Oracle in rule-based mode and not in cost-base mode. For this reason, ALEPH will not function properly if statistics for ALEPH tables/indexes are existing in the database. For these cases, use util a/8/1 to see if you have any statistics. Note: there is an exception to this rule. Z91 table and z91_id index must be present and should not be deleted.

UTIL A/9 Delete Statistics for Analyzed Tables / Indexes

Use this utility in order to delete unwanted statistics.

UTIL A/12 Drop Cache Table (Z50)

This table should be cleaned out periodically with the **Drop Cache Table (Z50)** utility. The utility drops the table and indexes and then creates them empty.

When you select this utility, the system will notify you that you will be deleting Z50 from the defined library. For example:

```
delete z50 in VIR01
```

UTIL A/13 Drop Statistics Table (Z34)

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_root/www_server.conf` file and the $`aleph_root/pc_server_defaults` file, then a Z34 record is generated each time a transaction takes place between the client and the server. If you do not want to save statistics, use this utility periodically to clean out the statistics table (Z34).

When you select this utility, the system will notify you that you will be deleting table Z34 from the `pw_library`. For example:

```
delete Z34 in USR00
```

☞ **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` file.

UTIL A/14 Drop "Update doc" Table (Z07)

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 cleaned out automatically by the system. They can be cleaned out 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
```

UTIL A/16 Drop Event Table (Z35)

When you select this utility, the system will notify you that you will be deleting table Z35 in the defined library. For example:

```
delete z35 in USM50
```

UTIL A/17 Manage Oracle Tables

```
A.17. Manage Oracle Tables of USM01
       0.  Exit Procedure
       1.  Drop & Create Table and Index
       2.  Create Index
       3.  Rebuild Index
       4.  Drop Index
       5.  Synonyms
       6.  Triggers
       7.
       8.
       9.  Sequences
      10.  Edit file_list of USM01 Tables
      11.  Space Utilization
      12.  Check existance of table space for <library name>
      13.
      14.  List Existing Indexes for a Table
      15.  Analyze Table/Index
      16.
      17.  Manage Context Indexes
      18.  Search for Duplicate Keys
```

UTIL A/17/1 Drop & Create Table and Index

Drops and creates an empty Oracle table and its corresponding indexes from the library. You will be prompted.

- **all** - Enter `all` to create/recreate all of the tables in the library and their indexes.
- **<table-name>** - Enter a table name to create/recreate the specified table and all of its indexes.

UTIL A/17/2 Create Index

Creates Oracle index(es) for one or more tables in the library. You will be prompted.

- **all** - Enter `all` to create/recreate all the indexes for all the tables.
- **<table-name>** - Enter a table name to create all of the given table’s indexes.
- **<index-name>** - Enter an index name (for example, Z01_id1) to create the given index.

UTIL A/17/3 Rebuild Index

Rebuilds an Oracle index in the library. This utility is used to reorganize the index, and is much faster than dropping an index and recreating it. You will be prompted.

- **all** - Enter `all` to rebuild all the indexes for all the tables.
- **<table-name>** - Enter a table name to rebuild all of the given table's indexes.
- **<index-name>** - Enter an index name (e.g Z01_id1) to rebuild the given index.

UTIL A/17/4 Drop Index

Drops an Oracle index. You will be prompted.

- **all** - Enter `all` to drop all the indexes for all the tables.
- **<table-name>** - Enter a table name to drop all of the given table's indexes.
- **<index-name>** - Enter an index name (for example, Z01_id1) to drop the given index.

UTIL A/17/5 Synonyms

```
     A.17.5 Manage Synonyms in USM01

       1.  List Library Synonyms
       2.  Create/Recreate All Library Synonyms
```

UTIL A/17/5/1   List Library Synonyms

Lists the synonyms defined in the current library.

UTIL A/17/5/2   Create/Recreate All Library Synonyms

Drops all existing synonyms in the library and then creates all the library's synonyms as defined in the library's **file_list**.

UTIL A/17/6 Triggers

```
     A.17.6 Manage Triggers in USM01

       1.  Create/Recreate All Library Triggers
       2.  Enable Trigger
       3.  Disable Trigger
       4.  Show Trigger Status
```

UTIL A/17/6/1 Create/Recreate All Library Triggers

Creates all the library’s triggers as defined in the **file_list**. After a trigger is created, it is automatically enabled.

UTIL A/17/6/2 Enable Trigger

Enables a library trigger. After a trigger is created, it is automatically enabled. This utility is only needed to re-enable a trigger that has been disabled.

UTIL A/17/6/3 Disable Trigger

Disables a library trigger. When a trigger is disabled, its definition still exists but it will not actually work until enabled again.

UTIL A/17/6/4 Show Trigger Status

Shows for each trigger in the library, the library name, the table which the trigger is defined on, and the current status of the trigger (enabled or disabled).

```
SQL*Plus: Release 11.2.0.2.0 - Production on Mon Feb 20 17:46:03 2006

Copyright (c) 1982, 2005, Oracle.  All rights reserved.


Connected to:
Oracle Database 10g Enterprise Edition Release 11.2.0.2.0- Production
With the Partitioning, OLAP and Data Mining Scoring Engine options

aleph_admin@ALEPH1> aleph_admin@ALEPH1> aleph_admin@ALEPH1>   2    3
4
TRIGGER_NAME                   TABLE_NAME                        STATUS
---------------------------  ------------------------------  --------
Z1003_TRIGGER                                                       Z36
ENABLED
aleph_admin@ALEPH1> Disconnected from Oracle Database 10g Enterprise
Edition Release 11.2.0.2.0- Production
With the Partitioning, OLAP and Data Mining Scoring Engine options
```

UTIL A/17/9 Sequences

```
     A.17.9 Manage Sequences in USM01

       1.
       2.  Create/Recreate Library Sequences
```

UTIL A/17/9/2 Create/Recreate Library Sequences

Creates or recreates library sequences. You will be prompted.

- **all** - Enter `all` to Drop and create all the library’s sequences as defined in the **file_list.**
- **<sequence-name>** - Enter a sequence name to drop and create the given sequence.

UTIL A/17/10 Edit file_list of USM01 Tables

UTIL A/17/10/1

Opens the library's **file_list** for editing.

UTIL A/17/10/2

Opens the merged file_list created from the template and the one in data_root.

UTIL A/17/11 Space Utilization

```
     A.17.11 Space Utilization in USM01

       1.  Check Space Utilization of Oracle Tables
       2.  Check Space Utilization of a Table/Index
       3.  Check Space Utilization of Dynamic Tables
```

UTIL A/17/11/1 Check Space Utilization Of Oracle Files

```
*****  Space utilization of USM01 files *****

The report is : count_rep.lst
Do you wish to edit the report now? [y/n]y


                   ***** USM01 Tables statistics *****



                 SEGMENT     TABLESPACE
SEGMENT NAME     TYPE        NAME       BYTES (K)      BLOCKS  EXTENTS
---------------  ----------  -------  ----------  ----------  -------
Z00              TABLE       TS0           23552        2944       23
Z00H             TABLE       TS0             128          16        2
Z00H_ID          INDEX       TS1             128          16        2
Z00R             TABLE       TS0            2048         256       17
Z00R_ID          INDEX       TS1             640          80       10
Z00R_ID1         INDEX       TS1             576          72        9
Z00_ID           INDEX       TS1             512          64        8
Z01              TABLE       TS0           27648        3456       27
Z0101            TABLE       TS0           13312        1664       28
Z0101_ID         INDEX       TS1            2048         256       17
Z0101_ID1        INDEX       TS1            8192        1024       23
Z0101_ID2        INDEX       TS1            2048         256       17
Z0102            TABLE       TS0           17408        2176       17
Z01_ID           INDEX       TS1           11264        1408       26
Z01_ID2          INDEX       TS1            3072         384       18
```

UTIL A/17/11/2 Check Space Utilization Of Table/Index

```
Enter Table/Index Name: Z01
TABLE_NAME = Z01
check = USM01
active_library = USM01

BYTES/1024      BLOCKS     EXTENTS  INITIAL_EXTENT  NEXT_EXTENT
----------  ----------  ----------  --------------  -----------
     27648        3456          27         2097152

Elapsed: 00:00:02.54

  COUNT(*)
----------
     98488

Elapsed: 00:00:00.41
```

UTIL A/17/11/3 Check Space Utilization Of Dynamic Tables

```
TABLE   BYTES/1024  BLOCKS  EXTENTS  INITIAL_EXTENT  NEXT_EXTENT

Z07
       128         16          2          106496
Z109
       128         16          2          106496
```

UTIL A/17/12 Check existance of table space for <library>

The utility checks existence of all table spaces mentioned in file_list of active library.

UTIL A/17/14 List Existing Indexes For A Table

Lists the indexes which should exist for a table according to the library **file_list**.

```
Enter Table Name : Z01

 Defined in file_list:
___________________________________________________________
IND z01_id           1M               1M               ts1
IND z01_id2          300K             100K             ts1
IND z01_id3          200K             100K             ts1
IND z01_id4          200K             100K             ts1
IND z01_id5          200K             100K             ts1
___________________________________________________________


 Exist in the Database:

INDEX_NAME          STATUS  INDEX_TYPE  UNIQUENESS  COLUMN_NAME
------------------  ------  ----------  ----------  -----------
Z01_ID              VALID    NORMAL     NONUNIQUE   Z01_REC_KEY
Z01_ID2             VALID    NORMAL     UNIQUE      Z01_ACC_SEQUENCE
Z01_ID3             VALID    NORMAL     NONUNIQUE   Z01_REC_KEY_4
Z01_ID4             VALID    NORMAL     NONUNIQUE   Z01_HASH
Z01_ID5             VALID    NORMAL     NONUNIQUE   Z01_UPDATE_Z0102
```

UTIL A/17/15 Analyze Table/Index

```
     A.17.15 Analyze Table/Index

       0.  Exit procedure
       1.
       2.
       3.
       4. Validate Index Structure
```

UTIL A/17/15/4 Validate Index Structure

```
Enter Index Name to Validate Structure : Z01_id
enter yes to Analyze Z01_id indexes for USM01 yes

Index analyzed.


PCT_DELETED  DISTINCTIVENESS
-----------  ---------------
          0                0

1 row selected.


 Enter to continue
```

UTIL A/17/17 Manage Context Indexes

```
     A.17.17 Manage Context Indexes

       0.  Exit Procedure
       1.
       2.
       3.
       4.
       5. Syncronize Context Index
       6. Optimize Context Index
```

UTIL A/17/17/5 Synchronize Context Index

Changes to the detail table do not trigger re-indexing when you synchronize the index. Only changes to the indexed column in the master table trigger a re-index when you synchronize the index.

ALEPH does this synchronization automatically, but this utility allows you to do it manually

UTIL A/17/17/6 Optimize Context Index

Use this procedure to optimize the index. You optimize your index after you synchronize it. Optimizing an index removes old data and minimizes index fragmentation, which can improve query response time.

ALEPH is doing this optimize automatically but this utility allows you to do it manually

UTIL A/17/18 Search for Duplicate Keys

This utility helps in locating and/or deleting duplicate keys in a column which must have a unique index on it. It is generally used when creating a unique index fails because of duplicate keys. Enter the table name, index number and column name. You will then be prompted to confirm the creation of a non-unique index in order to find the duplicate keys. In the option LIST, a list of the duplicate keys is generated into a file. In the option DELETE, the rows in the table with the duplicate keys are deleted (leaving only one row per key) then the unique key is created.

Notes:

1. Make sure you have a sufficient backup before using the DELETE option.
2. This utility is rarely needed. It is used mainly to troubleshoot during the conversion phase.

UTIL A/19 Export Library

Exports all Oracle tables in the library into `$data_files/<library-name>.exportSEQ.tar`.gz The utility uses ALEPH's Dump to Sequential procedures.

Note: Depending on the volume of data, this process might take considerable time and disk space. Make sure that `$data_files` is has enough free space. Note that this utility locks the library.

UTIL A/20 Import Library

Imports all of the library’s data from `$data_files/<library-name>. exportSEQ.tar.gz` into Oracle’s tables. The utility uses ALEPH’s Dump to Sequential procedures.

Note: Depending on data volume, this process might take considerable time. Note that this utility locks the library.

UTIL A/21 List Objects

```
     A.21 List Objects
   ----------------------------------------
         0.  Exit Procedure
         1.  Show Object Information
         2.  Show List of Objects by Prefix
         3.  Show List of Objects of Type
```

UTIL A/21/1 Show Object Information

This utility shows the object type, date of creation and current status. It is useful for finding the type of an object when you know the object’s name but are not sure of its type. For example, you might want to find out if Z50 is a table in the library or a synonym.

Example 1: TABLE

```
Enter Object Name : Z50

OBJECT_NAME           OBJECT_TYPE         CREATED          STATUS
--------------------  ------------------  ---------------  -------
Z50                   TABLE               05-OCT-03        VALID
```

Example 2: SYNONYM

```
Enter Object Name : Z50

OBJECT_NAME           OBJECT_TYPE         CREATED          STATUS
--------------------  ------------------  ---------------  -------
Z50                   SYNONYM             08-OCT-03
```

Example 3: INDEX

Enter Object Name : Z50_id

```
OBJECT_NAME           OBJECT_TYPE         CREATED          STATUS
```

```
--------------------  ------------------  ---------------  -------
Z50_ID                INDEX               05-OCT-03        VALID
```

Example 4: TRIGGER

```
Enter Object Name : Z1003_trigger

OBJECT_NAME          OBJECT_TYPE         CREATED          STATUS
-------------------  ------------------  ---------------  -------
Z1003_TRIGGER        TRIGGER             05-OCT-03        VALID
```

Example 5: SEQUENCE

```
Enter Object Name : last_result_set

OBJECT_NAME         OBJECT_TYPE         CREATED          STATUS
------------------  ------------------  ---------------  -------
LAST_RESULT_SET     SEQUENCE            05-OCT-03        VALID
```

UTIL A/21/2 Show List of Objects by Prefix

This utility is useful to list a group of objects that start with the same characters. For example:

```
Enter Object Prefix : Z01

OBJECT_NAM  OBJECT_TYPE         CREATED          STATUS
----------  ------------------  ---------------  -------
Z01         TABLE               05-OCT-03        VALID
Z0101       TABLE               05-OCT-03        VALID
Z0101_ID    INDEX               05-OCT-03        VALID
Z0101_ID1   INDEX               05-OCT-03        VALID
Z0101_ID2   INDEX               05-OCT-03        VALID
Z0102       TABLE               07-OCT-03        VALID
Z01_ID      INDEX               05-OCT-03        VALID
Z01_ID2     INDEX               05-OCT-03        VALID
Z01_ID3     INDEX               05-OCT-03        VALID
Z01_ID4     INDEX               05-OCT-03        VALID
Z01_ID5     INDEX               05-OCT-03        VALID
```

UTIL A/21/3 Show List of Objects of Type

This utility is useful to list a group of objects of the same type. You will be prompted:

```
Enter Object Type [TABLE/INDEX/SYNONYM/SEQUENCE ..] :
```

For example, after entering **`table`** you might see:

```
OBJECT_NAME                     CREATED          STATUS   OBJECT_TYPE
------------------------------  ---------------  -------  -----------
Z00                             05-OCT-03        VALID    TABLE
Z00H                            05-OCT-03        VALID    TABLE
Z00R                            05-OCT-03        VALID    TABLE
Z01                             05-OCT-03        VALID    TABLE
Z0101                           05-OCT-03        VALID    TABLE
Z0102                           07-OCT-03        VALID    TABLE
Z02                             05-OCT-03        VALID    TABLE
Z07                             05-OCT-03        VALID    TABLE
Z101                            05-OCT-03        VALID    TABLE
Z103                            05-OCT-03        VALID    TABLE
Z104                            05-OCT-03        VALID    TABLE
```

2.2. UTIL O - Oracle Management

```
O. Managing ORACLE
    ---------------
     0.  Exit Procedure
     1.  Oracle Server
     2.  Oracle Listener
     3.  Oracle Logs
     4.
     5.
     6.  Nls
     7.  Archiving
     8.
     9.  Database Users
    10.  SQL*Plus Session
    11.
    12.  Database Verification Utility
    13.  Database Files
    14.  Database Free/Used Space
    15.
    16.  Database Links
    17.  Database Tablespaces
    18.  Oracle Statistics
    19.  Shared Pool
    20.  Multi Threaded Server
```

General:

ALEPH 500 is based on **Oracle11*g* RDBMS**.

As explained in the introduction, in ALEPH every library has a separate root directory. Each ALEPH library's directory contains information relevant for administrating the library.

Each library's data is stored within Oracle in a separate Oracle user. Each Oracle user owns a set of tables which contain the library's data.

Note:
ALEPH enables you to place the Oracle database on a separate server from the ALEPH application, and even to distribute Oracle tables across two or more databases on different servers. See Working With Remote Oracle Tables on page 59.

Placing the Oracle database on a separate server is done by setting the `TWO_TASK` environment variable (in the `aleph_start` file in the "ALEPH Server") to the alias pointing to the "Oracle server", as defined in the `tnsname.ora` file in the "ALEPH Server". See Working in a TWO_TASK Environment on page 63.

Note:
Some UTIL O functionality does not work when the Oracle database is installed on a separate server. Ex Libris recommends using the Oracle Enterprise Manager Database console for database management in a TWO_TASK environment. For more information, see:
http://download.oracle.com/docs/cd/E11882_01/server.112/e10897/em_manage.htm#ADMQS031

2.2.1. About the Oracle Listener

When ALEPH is installed on the same server as the database, ALEPH can work without the Listener.

The Listener must run on the server if a third party product has to connect the database, or if there is a remote server that is connected to the database. For example, when ALEPH is installed on one server and the database is on a different server, the Listener must be running on the database server in order for ALEPH to work properly. See more about this option in Working With Remote Oracle Tables on page 59.

UTIL O/1 Oracle Server

```
O.1 Oracle Server
        ---------------
```

```
          0.  Exit Procedure
          1.  Activate Oracle Server
          2.  Close Oracle Server
          3.  Show Running Oracle Server
          4.  Show Oracle Server Status
```

Note:
UTIL O/1 does not work in a TWO_TASK environment.

UTIL O/1/1 Activate Oracle Server

Note: Requires ALEPH DBA username and password

In order for ALEPH to interact with Oracle, the Oracle server must be running. The Oracle Listener must be running in certain cases, as explained in section About the Oracle Listener above. They may be automatically started at boot time (this is determined during installation) and also controlled by the ALEPH Oracle Management utilities under **UTIL O**.

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: aleph_dba/<aleph_dba password>
```

Enter the ALEPH DBA username and password.

UTIL O/1/2 Close Oracle server

Note: Requires ALEPH DBA username and password

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 later on, you will need to select **UTIL O/1/1 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:aleph_dba/<aleph_dba password>
```

Enter the ALEPH DBA username and password.

UTIL O/1/3 Show Running Oracle Server

This utility displays the Oracle server. Here is an example of a running Oracle server:

```
  oracle  5200     1  0   Sep 25 ?        0:00 ora_arc1_aleph1
  oracle  5162     1  0   Sep 25 ?        0:02 ora_pmon_aleph1
  oracle  5194     1  0   Sep 25 ?        0:00 ora_d006_aleph1
  oracle  5166     1  0   Sep 25 ?        1:47 ora_lgwr_aleph1
  oracle  5186     1  0   Sep 25 ?        0:00 ora_d002_aleph1
  oracle  5164     1  0   Sep 25 ?        0:16 ora_dbw0_aleph1
  oracle  5184     1  0   Sep 25 ?        0:00 ora_d001_aleph1
  oracle  5188     1  0   Sep 25 ?        0:00 ora_d003_aleph1
  oracle  5170     1  0   Sep 25 ?        1:34 ora_smon_aleph1
  oracle  5172     1  0   Sep 25 ?        0:00 ora_reco_aleph1
  oracle  5180     1  0   Sep 25 ?        0:00 ora_s003_aleph1
  oracle  5176     1  0   Sep 25 ?        0:00 ora_s001_aleph1
  oracle  5168     1  0   Sep 25 ?        0:23 ora_ckpt_aleph1
  oracle  5178     1  0   Sep 25 ?        0:00 ora_s002_aleph1
  oracle  5196     1  0   Sep 25 ?        0:00 ora_d007_aleph1
  oracle  5190     1  0   Sep 25 ?        0:00 ora_d004_aleph1
  oracle  5198     1  0   Sep 25 ?        0:00 ora_arc0_aleph1
  oracle  5192     1  0   Sep 25 ?        0:00 ora_d005_aleph1
  oracle  5174     1  0   Sep 25 ?        0:23 ora_s000_aleph1
  oracle  5182     1  0   Sep 25 ?        0:00 ora_d000_aleph1
```

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

The lines that appear on your server may differ slightly from the lines presented here. The lines show the background processes and the dispatchers and shared servers used by your Oracle instance (database).

If these lines do not appear, the Oracle server may be activated using **UTIL O/1/1 Activate Oracle Server**.

UTIL O/1/4 Show Oracle Server Status

```
INSTANCE_N   HOST_NAME           VERSION          STARTUP_TI   STATUS
LOGINS
----------  ---------------  ------------  ----------  ----------  --
--------
aleph1           il-aleph02.exli    11.2.0.2.0      02-FEB-06      OPEN
ALLOWED
            bris-int.il


BANNER
----------------------------------------------------------------
Oracle Database 10g Enterprise Edition Release 11.2.0.2.0- Prod
PL/SQL Release 11.2.0.2.0- Production
CORE    11.2.0.2.0     Production
TNS for Linux: Version 11.2.0.2.0- Production
```

```
NLSRTL Version 11.2.0.2.0- Production
```

UTIL O/2 Oracle Listener

```
     O.2 Oracle Listener
        -----------------
         0.  Exit Procedure
         1.  Activate Oracle Listener
         2.  Close Oracle Listener
         3.  Show Running Oracle Listener
         4.  Show Listener Status
         5.  Show Listener Services
```

When a user process makes a connection request using a connect string, the Oracle Listener process examines the request and connects it to a server process. If Oracle and ALEPH are installed on the same server and no third party products have to connect to the database and no connections are being done from a remote server, ALEPH can work without the Listener. In any other case, both the Oracle server and the Oracle Listener must be running. They may be started automatically at boot time (this is determined during installation) and also controlled by the ALEPH Oracle Management utilities.

Note:
UTIL O/2 does not work in a TWO_TASK environment.

UTIL O/2/1 Activate Oracle Listener

Note: **Requires Oracle software owner password**

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.

UTIL O/2/2 Close Oracle Listener

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.

UTIL O/2/3 Show Running Oracle Listener

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

```
1513 ?    S    0:20 /exlibris/app/oracle/product/112112/bin/tnslsnr
LISTENER –inherit
```

UTIL O/2/4 Show Listener Status

When you select **Show Listener Status**, the following type of output will be displayed:

```
LSNRCTL  for  Linux:  Version  11.2.0.2.0-  Production  on  20-FEB-2006
17:55:52

Copyright (c) 1991, 2005, Oracle.  All rights reserved.

Connecting to (DESCRIPTION=(address=(protocol=ipc)(key=aleph1)))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                       TNSLSNR for Linux: Version 11.2.0.2.0-
Production
Start Date                02-FEB-2006 13:45:24
Uptime                    18 days 4 hr. 10 min. 28 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener                        Parameter                       File
/exlibris/app/oracle/product/112/network/admin/listene
r.ora
Listener                           Log                          File
/exlibris/app/oracle/product/112/network/log/listener.
log
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=aleph1)))
Services Summary...
Service "aleph1" has 1 instance(s).
  Instance  "aleph1",  status  UNKNOWN,  has  1  handler(s)  for  this
service...
Service "aleph1.il-aleph02.exlibris-int.il" has 1 instance(s).
The command completed successfully
```

UTIL O/2/5 Show Listener Services

When you select **Show Listener Services**, the following type of output will be displayed:

```
LSNRCTL  for  Linux:  Version  11.2.0.2.0-  Production  on  20-FEB-2006
17:57:23
```

```
Copyright (c) 1991, 2005, Oracle.  All rights reserved.

Connecting to (DESCRIPTION=(address=(protocol=ipc)(key=aleph1)))
Services Summary...
Service "aleph1" has 1 instance(s).
  Instance  "aleph1",  status  UNKNOWN,  has  1  handler(s)  for  this
service...
    Handler(s):
      "DEDICATED" established:0 refused:0
         LOCAL SERVER
Service "aleph1.il-aleph02.exlibris-int.il" has 1 instance(s).
  Instance  "aleph1",  status  READY,  has  9  handler(s)  for  this
service...
    Handler(s):
      "DEDICATED" established:2 refused:0 state:ready
         LOCAL SERVER
      "D007" established:2 refused:0 current:2 max:1022 state:ready
         DISPATCHER <machine: il-aleph02.exlibris-int.il, pid: 1654>
         (ADDRESS=(PROTOCOL=ipc)(KEY=#1654.1))
...
The command completed successfully
```

UTIL O/6 NLS

```
     O.6 NLS
        ---------------
         0.  Exit Procedure
         1.  Show NLS Parameters
```

UTIL O/6/1 Show NLS Parameters

ALEPH version 18 uses the UTF8 character set (earlier versions use the US7ASCII character set). This utility shows the NLS (National Language Support) definition of the database.

```
PARAMETER                               VALUE
===========================================================
NLS_LANGUAGE                            AMERICAN
NLS_TERRITORY                           AMERICA
NLS_CURRENCY                            $
NLS_ISO_CURRENCY                        AMERICA
NLS_NUMERIC_CHARACTERS                  .,
NLS_CALENDAR                            GREGORIAN
NLS_DATE_FORMAT                         DD-MON-RR
NLS_DATE_LANGUAGE                       AMERICAN
NLS_CHARACTERSET                        UTF8
NLS_SORT                                BINARY
NLS_TIME_FORMAT                         HH.MI.SSXFF AM
NLS_TIMESTAMP_FORMAT                    DD-MON-RR HH.MI.SSXFF AM
NLS_TIME_TZ_FORMAT                      HH.MI.SSXFF AM TZR
NLS_TIMESTAMP_TZ_FORMAT                 DD-MON-RR HH.MI.SSXFF AM TZR
NLS_DUAL_CURRENCY                       $
NLS_NCHAR_CHARACTERSET                  UTF8
NLS_COMP                                BINARY
```

```
NLS_LENGTH_SEMANTICS                    BYTE
NLS_NCHAR_CONV_EXCP                     FALSE
```

UTIL O/7 Archiving

```
O.7 Archiving
     ---------------
      0.  Exit Procedure
      1.  Turning Archiving On
      2.  Turning Archiving Off
      3.  Show Archiving Status
```

Introduction to Archiving

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.

If the database is in NOARCHIVELOG mode, only cold backups can be performed. In addition, when recovering using a cold backup, the data will be restored to the time the backup was performed and all the changes done afterwards until the time of the failure will be lost.

If the database is in ARCHIVELOG mode, both cold and hot backups can be used to recover the database until the time of the failure, providing that all the archive files that were generated from the time the backup (hot or cold) was performed until the time of failure are available. 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 Oracle backup manual for more information.

Note: Changing the archiving mode shuts down the database and restarts it again in ARCHIVELOG mode on.

The sequence of events is as follows:

1. ALEPH processes (servers and batch procedures) are stopped (using the **aleph_shutdown** script in `$alephe_root`).
2. Oracle database is shut down.
3. Oracle database is started up.
4. ALEPH is restarted (using the **aleph_startup** script in `$alephe_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 perform a full cold backup and then turn archiving back on.

Performing a full cold backup after switching to ARCHIVELOG mode is mandatory, because otherwise there will be a gap in ARCHIVELOG files which will prevent recovering.

Note:
UTIL O/7 does not work in a TWO_TASK environment.

UTIL O/7/1 Turning Archiving On

Note: Requires ALEPH DBA username and password

This utility turns Oracle archiving on.

Note: Changing the archiving mode shuts down the database and restarts it again in ARCHIVELOG mode on.

The sequence of events is as follows:

1. ALEPH processes (servers and batch procedures) are stopped (using the **aleph_shutdown** script in `$alephe_root`).
2. Oracle database is shut down.
3. Oracle database is started up.
4. ALEPH is restarted (using the **aleph_startup** script in `$alephe_root`).

UTIL O/7/2 Turning Archiving Off

Note: Requires ALEPH DBA username and password

This utility turns Oracle archiving off.

Note: Changing the archiving mode shuts down the database and restarts it again in ARCHIVELOG mode off.

The sequence of events is as follows:

1. ALEPH processes (servers and batch procedures) are stopped (using the **aleph_shutdown** script in `$alephe_root`).
2. Oracle database is shut down.
3. Oracle database is started up.
4. ALEPH is restarted (using the **aleph_startup** script in `$alephe_root`).

UTIL O/7/3 Show Archiving Status

Note: Requires ALEPH DBA username and password

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

```
SQL*Plus: Release 11.2.0.2.0- Production on Mon Feb 20 18:01:57
2006

Copyright (c) 1982, 2005, Oracle.  All rights reserved.

idle> Connected.
idle> idle> Database log mode          No Archive Mode
Automatic archival             Disabled
Archive destination            USE_DB_RECOVERY_FILE_DEST
Oldest online log sequence     384
Current log sequence           388
idle> Disconnected from Oracle Database 10g Enterprise Edition
Release 11.2.0.2.0- Production
With the Partitioning, OLAP and Data Mining Scoring Engine
options
```

When archiving is on you will see that **Database log mode** is set to Archive Mode and **Automatic archival** is enabled.

In a production database, the Database log mode should always be set to Archive Mode.

UTIL O/9 Database Users

```
O.9. Database Users
   ---------------
   0. Exit Procedure
   1. List Database Users
   2. Create a New User
   3.
```

UTIL O/9/1 List Database Users

This utility shows the list of all the users that exist in the database. Note that some of the users are ALEPH library users and others are administrative users.

```
The Database aleph1 Contains the Following Users:
======================================================
SYS
SYSTEM
OUTLN
DBSNMP
ALEPH
ALEPH_ADMIN
CTXSYS
PERFSTAT
ALEPH_BACKUP
ALEPH_DBA
USM01
USM10
USM11
USM12
```

```
USM14
USM15
USM19
USM20
USM23
```

UTIL O/9/2 Create a New User

This utility creates a new user and gives it a default password which is the same as the username. Note that if the name of the user that you gave already exists, all the tables belonging to that user with all the data will be dropped and the user will be created with all its tables empty. Afterwards, it is possible to use **UTIL Y/8 Update Oracle Passwords of ALEPH Users (ora_passwd)** to change the user’s password.

```
Enter User Name to Create New User: <new user name>
Enter User Name to Create New User: <new user name>
enter yes to create oracle user test <yes>
default password is test
if user test exists all data will be erased!!!
enter no to reconfirm no
source create_ora_user_b test
create_ora_user_b test

SQL*Plus: Release 11.1.0.7.0 - Production on Thu Feb 5 11:28:44 2009

Copyright (c) 1982, 2008, Oracle.  All rights reserved.

Enter user-name:
Connected to:
Oracle Database 10g Enterprise Edition Release 11.2.0.2.0- Production
With the Partitioning, OLAP and Data Mining Scoring Engine options

aleph_admin@ALEPH1> EXIT
Disconnected from Oracle Database 10g Enterprise Edition Release
11.2.0.2.0- Production
With the Partitioning, OLAP and Data Mining Scoring Engine options

SQL*Plus: Release 11.2.0.2.0- Production on Mon Feb 20 18:04:00 2006

Copyright (c) 1982, 2005, Oracle.  All rights reserved.

Enter user-name:
Connected to:
Oracle Database 10g Enterprise Edition Release 11.2.0.2.0- Production
With the Partitioning, OLAP and Data Mining Scoring Engine options

aleph_admin@ALEPH1> DROP USER test CASCADE
          *
ERROR at line 1:
ORA-01918: user 'TEST' does not exist



User created.
```

```
Grant succeeded.

Disconnected from Oracle Database 10g Enterprise Edition Release
11.2.0.2.0- Production
With the Partitioning, OLAP and Data Mining Scoring Engine options
```

UTIL O/10 SQL*Plus Session

In **Oracle 11g**, **svrmgrl** no longer exists. In **Oracle 11g SQL*Plus** is used in order to perform sysdba operations which were previously done via **svrmgrl**. This utility does `sqlplus / nolog`. You can then connect as sysdba using the sysdba user **aleph_dba**.

This utility starts an **Oracle SQL*Plus** session as sysdba. Enter the following:

```
***** DO: connect aleph_dba/aleph_dba_passwd as sysdba

SQL*Plus: Release 11.1.0.7.0 - Production on Thu Feb 5 11:28:44 2009

Copyright (c) 1982, 2008, Oracle.  All rights reserved.

idle>
```

UTIL O/12 Database Verification Utility

```
 O.12. Database Verification Utility
      --------------------------------
      0. Exit procedure
      1. Run Database Verification Utility
      2. Find Corrupted Object
```

UTIL O/12/1 Run Database Verification Utility

Note: **Requires Oracle password.**

This procedure verifies that all the Oracle datafiles are fully readable and accessible. It is advisable to run it periodically for all database files.

When you select **Database Verification Utility** you will be prompted:

```
Select one of the oracle files:
/exlibris/oradata/aleph1/aleph1_ts0_01.dbf
Enter database block size [8192]:

To continue you will need to enter oracle's password.
Password:
Execute oracle cshrc
Oracle 11.2.0.2.0
Oracle Home : /exlibris/app/oracle/product/112         Oracle SID :
aleph1
```

```
DBVERIFY: Release 11.2.0.1.0 - Production on Mon Feb 20 18:16:29 2006

Copyright (c) 1982, 2010, Oracle.  All rights reserved.

DBVERIFY - Verification starting : FILE =
/exlibris/oradata/aleph1/aleph1_ts0_01.dbf
.................................................................

DBVERIFY - Verification complete

Total Pages Examined         : 229376
Total Pages Processed (Data) : 202702
Total Pages Failing   (Data) : 0
Total Pages Processed (Index): 575
Total Pages Failing   (Index): 0
Total Pages Processed (Other): 5869
Total Pages Processed (Seg)  : 0
Total Pages Failing   (Seg)  : 0
Total Pages Empty            : 20230
Total Pages Marked Corrupt   : 0
Total Pages Influx           : 0
Highest block SCN            : 9336450 (0.9336450)
```

Note:
UTIL O/12/1 does not work in a TWO_TASK environment.

UTIL O/12/2 Find Corrupted Object

If **UTIL O/12/1 Run Database Verification Utility** indicates that corrupt blocks were found, use this utility to identify the Oracle objects that reside in the corrupted blocks(s).

UTIL O/13 Database Files

```
   O.13 Database Files
        ---------------
         0.  Exit Procedure
         1.  List of Database Files
         2.  Resize Oracle Datafile
         3.  Add File to Tablespace
         4.  Show Datafile Free Blocks by KBytes
         5.  Show Datafile Free Blocks by BlockID
```

Note:
UTIL O/13 does not work for databases created using ASM technology.

UTIL O/13/1 List of Database Files

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

```
The Database aleph1 Contains the Following Files:
======================================================

T        NAME                                          SIZE K  F
-------  -------------------------------------------  -------  ---
DRSYS    /exlibris3/oradata/aleph1/aleph1_drsys_01.dbf    86016  7
SYSTEM   /exlibris3/oradata/aleph1/aleph1_system_01.dbf  266240  1
TOOLS    /exlibris3/oradata/aleph1/aleph1_tools_01.dbf   204800  3
TS0      /exlibris3/oradata/aleph1/aleph1_ts0_01.dbf     525312  4
TS0      /exlibris2/oradata/aleph1/aleph1_ts0_02.dbf    2048000  8
TS1      /exlibris3/oradata/aleph1/aleph1_ts1_01.dbf     525312  5
TS1      /exlibris2/oradata/aleph1/aleph1_ts1_02.dbf    1228800  9
TSLOB    /exlibris3/oradata/aleph1/aleph1_tslob_01.dbf    51200  6
TSLOB    /exlibris3/oradata/aleph1/aleph1_tslob_02.dbf    30720  10
UNDOTS   /exlibris3/oradata/aleph1/aleph1_undots_01.dbf  524288  2

T        NAME                                          SIZE K  F
-------  -------------------------------------------  -------  ---
TEMP     /exlibris3/oradata/aleph1/aleph1_temp_01.dbf   1024000  1
TEMP     /exlibris3/oradata/aleph1/aleph1_temp_02.dbf     61440  2
```

UTIL O/13/2 Resize Oracle Datafile

This utility is used to enlarge or reduce the size of an Oracle datafile. You will be prompted for the datafile name and the new size.

Note:
Prior to enlarging the Oracle datafile, ensure that you have enough disk space on the file system where the datafile is to be placed. For databases created on a separate server (TWO_TASK mode), check the disk space on the database server and not on the application server, from where you activate UTIL O.

UTIL O/13/3 Add File to Tablespace

Note: Requires ALEPH DBA username and password

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

When you select **Add file to Tablespace** you will be prompted for the ALEPH DBA username and password. After you enter the username and password you will be prompted for the tablespace name. Enter the tablespace name and a list of the existing 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).

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

```
To add a file to a tablespace enter ALEPH_DBA username/password.
username/password:aleph_dba/<aleph_dba password>

Enter Tablespace name: ts0

Tablespace TS0 consist of the following files:

/exlibris3/oradata/aleph1/aleph1_ts0_01.dbf
/exlibris3/oradata/aleph1/aleph1_ts0_02.dbf

Enter new file name: /exlibris3/oradata/aleph1/aleph1_ts0_03.dbf
Enter file size (MB): 2000

Tablespace:  TS0
New file:    /exlibris3/oradata/aleph1/aleph1_ts0_03.dbf
Size:        2000MB

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

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

Note:
Prior to enlarging the Oracle datafile, ensure that you have enough disk space on the file system where the datafile is to be placed. For databases created on a separate server (TWO_TASK mode), check the disk space on the database server and not on the application server, from where you activate UTIL O.

UTIL O/13/4 Show Datafile Free Blocks by KBytes

```
Free Blocks Report by Kbytes
==========================================

TABLES  F    BLOCK_ID  KBYTES   NAME
------  ---  --------  ------   ----------------------------------
TS1     9    65161     707520   /exlibris2/oradata/aleph1/aleph1_ts
                                1_02.dbf

UNDOTS  2    2953      500672   /exlibris3/oradata/aleph1/aleph1_un
                                dots_01.dbf

TS0     8    227465    228288   /exlibris2/oradata/aleph1/aleph1_ts
                                0_02.dbf

TOOLS   3    2869      181856   /exlibris3/oradata/aleph1/aleph1_to
                                ols_01.dbf

TABLES  F    BLOCK_ID  KBYTES   NAME
```

```
------  ---  --------  ------   ---------------------------------
TS0     8    209673    110592   /exlibris2/oradata/aleph1/aleph1_ts
                                0_02.dbf

DRSYS   7    617       81088    /exlibris3/oradata/aleph1/aleph1_dr
```

UTIL O/13/5 Show Datafile Free Blocks by BlockID

You will be prompted. In order to determine the value of the Datafile Number, use **UTIL O/13/4 Show Datafile Free Blocks by Kbytes**.

```
Tablespace Name: ts0
Datafile Number: 4

  BLOCK_ID       BYTES
----------  ----------
    249865    39780352
    248729      917504
```

UTIL O/14 Database Free/Used Space

This utility provides information about the tablespaces' free space. The following submenu is displayed:

```
O.14.  Database Free/Used Space
     ------------------------
     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
     6. Coalesce Contiguous Free Extents
     7.
     8. Aleph Tablespaces Total/Free/Used Space Report
     9. Clean Temporary Tablespace Free Storage
```

UTIL O/14/1 All Tablespaces Free Space Summary

This utility 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 amount of free space in the tablespace (in megabytes).

3. MAX_EXTENT: The size of the largest contiguous extent of the tablespace (in megabytes).

4. NUM_FREE_EXTENTS: The number of free extents in the tablespace.

```
TABLESPACE_NAME  TOTAL_FREE_SPACE    MAX_EXTENT  NUM_FREE_EXTENTS
---------------  -----------------   ----------  ----------------
SYSTEM                   28.273343   28.2733435                 1
TEMP                     200.09429   40.0141679                33
TOOLS                   17.7753893   17.6972214                 2
TS0                     425.632244   118.549496               754
TS1                     444.236214   147.565435               118
USERS                    4.9949313    4.9949313                 1
```

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.

UTIL O/14/2 Number of Free Extents by Size in a Tablespace

This utility 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

SIZE IN MB            NUM OF EXTENTS
----------            --------------
       879                         1
         0                         4

Standard input: END
```

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

UTIL O/14/3 All Free Extents of Min Size in a Tablespace

This utility lists the 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

EXTENT_SIZE
-----------
 879.428397
 .156335878
 .117251908
 .117251908
 .117251908
Standard input: END
```

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

UTIL O/14/4 Space Used by a Library/Libraries in Each Tablespace

This utility shows for each library the amount of space that the library occupies in each tablespace. If a truncated library name is used, all the libraries starting with the given characters will be listed and the occupied space will be listed for each one of them.

```
Enter Library name (full, e.g. usm01, or truncated, e.g. usm): usm

OWNER                           TABLESPACE_NAME      SIZE_MB
---------------------------     ------------------   ----------
USM01                           TS0                  313.422168
USM01                           TS1                  209.615145
USM10                           TS0                  13.5074198
USM10                           TS1                  11.6313893
USM11                           TS0                  35.3319084
USM11                           TS1                  29.0784733
USM12                           TS0                  11.0685802
USM12                           TS1                  7.9418626
USM14                           TS0                  4.43993893
USM14                           TS1                  3.37685496
USM15                           TS0                  4.43993893


OWNER                          TABLESPACE_NAME  SIZE_MB
---------------------------    ---------------  ----------
USM15                          TS1              .062534351
USM19                          TS0              4.12726718
USM19                          TS1              1.68842748
USM20                          TS0              4.62754198
USM20                          TS1              3.56445802
```

UTIL O/14/5 Space Used by a Group of Libraries in Each Tablespace

This utility shows the total amount of space that all the libraries whose names start with the given characters occupy in each tablespace.

```
Enter first 3 characters of Library code (e.g. usm): usm
```

```
TABLESPACE_NAME                    SIZE_MB
------------------------------  ----------
TS0                             1514.76959
TS1                             848.466076
```

UTIL O/14/6 Coalesce Contiguous Free Extents

This utility is no longer needed when using Locally Managed Tablespace. It remains only for backward compatibility and will be removed in future versions.

Note: Requires ALEPH DBA username and password

Database free space may be composed of extents of various sizes. It is worthwhile to use this procedure to coalesce the contiguous free extents in an attempt to create larger free extents. Perform this procedure periodically.

You will be prompted for the ALEPH DBA username and password.

Enter the ALEPH DBA username and password (for example, `aleph_dba/<aleph_dba password>`).

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

UTIL O/14/8 Aleph Tablespaces Total/Free/Used Space Report

```
NAME                            TOTAL SIZE M
------------------------------  ------------
TS0                                     2513
TS1                                     1713
TSLOB                                     80

NAME                            TOTAL FREE M
------------------------------  ------------
TS0                                      414
TS1                                      723
TSLOB                                     72

NAME                            TOTAL USED M
------------------------------  ------------
TS0                                     2098
TS1                                      989
```

UTIL O/14/9 Clean Temporary Tablespace Free Storage

In rare cases, the temporary tablespace does not free non-used pages quickly enough. This utility is used to free those pages manually.

UTIL O/16 Manage Database Links

The **O/16 Manage Database Links** utilities are used to manage one logical database throughout the network from more than one physical database. They are used when there are some Oracle tables that are on one or more separate servers, and not on the ALEPH application's server.

For additional information about managing database links, see Working With Remote Oracle Tables on page 59.

Note: The use of database links is different from the use of TWO_TASK, where all the database is on a remote server.

```
O.16. Manage Database Links
   -------------------------------
   0. Exit Procedure
   1. List Database Links
   2. Create Database Link
   3. Drop Database Link
```

UTIL O/16/1 List Database Links

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

UTIL O/16/2 Create Database Links

This utility 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.

```
$ORACLE_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.aleph1` 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.aleph1
```

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

Database link created.
```

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, database links are used when working with tables on a remote database. See Working With Remote Oracle Tables on page 59.

UTIL O/16/3 Drop Database Link

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

UTIL O/17 Database Tablespaces

```
O.17. Manage Database Tablespaces
       ----------------------------
       0. Exit procedure
       1. Create a Tablespace
       2. List Tablespace Files
       3. Check Tablespace for a library based on file_list template
       4. Show Tablespaces Definition
       5. Show Tablespace Allocated/Free/Used Space
```

Note:
UTIL O/17 does not work for databases created using ASM technology.

UTIL O/17/1 Create a Tablespace

One rarely needs to create a tablespace, since all needeed tablespaces should have been created during system installation. This utility is used if there is a need for an additional tablespace. You can read about tablespaces and their types in *Introduction to Locally Managed Tablespaces* on page 12.

You will be prompted:

```
To Create a new Tablespace, Enter ALEPH_DBA username/password.
username/password:aleph_dba/<aleph_dba password>
Enter Tablespace name: test
Enter new file name (full path) :
/exlibris/oradata/aleph1/test_01.dbf
Enter new file size (MB): 1000
==============================================================
Tablespaces can be created with a UNIFORM size for all extents
or with allocation type AUTOALLOCATE which means
Oracle will decide how to define extents
Util o 17 4 can be used to see current definitions
for existing tablespaces
==============================================================
Tablespace Allocation Type : [AUTO/UNIFORM] UNIFORM
UNIFORM SIZE : [128K/1M/4M/128M/1920M]4m

Tablespace:  TEST
File:        /exlibris/oradata/aleph1/test_01.dbf
File size:    1000MB
Allocation : UNIFORM SIZE 4m
confirm (y/[n]):
```

Note:
Prior to enlarging the Oracle datafile, ensure that you have enough disk space on the file system where the datafile is to be placed. For databases created on a separate server (TWO_TASK mode), check the disk space on the database server and not on the application server, from where you activate UTIL O.

UTIL O/17/2 List Tablespace Files

```
Enter Tablespace name: ts1

Tablespace TS1 consist of the following files:

NAME                                                    SIZE K  F
--------------------------------------------------  ----------  ---
/exlibris3/oradata/aleph1/aleph1_ts1_01.dbf             525312  5
/exlibris2/oradata/aleph1/aleph1_ts1_02.dbf            1228800  9
```

UTIL O/17/3 Check Tablespace for a library based on file_list templates

This utility checks the existence of tablespaces for a library based on its' type and size. The check is done by comparing values in the file_list templates.

UTIL O/17/4 Show Tablespaces Definition

This utility shows for each tablespace: the types of extent management, segment allocation and tablespace (for permanent or temporary objects or for undo segments), and the tablespace status (online or offline).

```
TS_NAME     EXT_MGMT    ALLOC_TYP     INIT_EXT    NEXT_EXT  TYPE  STAT
------      ----------  ---------  ----------  ----------  ----  ----
DRSYS       LOCAL       SYSTEM          65536              PERM  ONL
```

```
SYSTEM       LOCAL       SYSTEM           65536               PERM  ONL
TEMP         LOCAL       UNIFORM        1048576     1048576   TEMP  ONL
TOOLS        LOCAL       UNIFORM          32768       32768   PERM  ONL
TS0          LOCAL       SYSTEM           65536               PERM  ONL
TS1          LOCAL       SYSTEM           65536               PERM  ONL
TSLOB        LOCAL       UNIFORM        8388608     8388608   PERM  ONL
UNDOTS       LOCAL       SYSTEM           65536               UNDO  ONL
```

UTIL O/17/5 Show Tablespace Allocated/Free/Used Space

This utility shows a given tablespace's total tablespace size, amount of free space and amount of used space.

```
Enter Tablespace name : ts1
Tablespace TS1 :
_____________________________
TOTAL SIZE M
------------
        1713

TOTAL FREE M
------------
         723

TOTAL USED M
------------
         989
```

UTIL O/18 Oracle Statistics

This utility provides the following Oracle Statistics:

```
    O.18. Oracle Statistics
       --------------------------------
       0. Exit Procedure
       1. Performance Statistics
       2. Rollback Segments Definitions
       3. Rollback Segments Dynamic Allocation
       4. View Long Operations
       5. IO Statistics
       6. Sort Operations
```

Enter the number of the utility and the statistics will be displayed.

UTIL O/18/1 Performance Statistics

```
opened cursors current
       254
db block gets
    281077
consistent gets
```

```
   3854704
NAME
     VALUE
physical reads
     36747
physical writes
     17228
DBWR checkpoints
         2
NAME
     VALUE
redo log space requests
         2
sorts (memory)
      7063
sorts (disk)
         4
=======================================================
DATA DICTIONARY CACHE (shared_pool_size)
      GETS      MISSES
RATIO
     79930        3224
95.97%
=======================================================
LIBRARY CACHE (shared_pool_size)
EXECUTIONS      MISSES
LIBCACHEPROZ
    265828         321
99.88%
```

UTIL O/18/2 Rollback Segments Definitions

In **Oracle 9*i***, there is a new feature named **Automatic Undo Management (AUM)**, which simplifies and automates the management of undo segments. When **AUM** is used, the rollback segments are defined and managed by Oracle and called **UNDO segments**. Utilities **O/18/2 Rollback Segments Definitions** and **O/18/3 Rollback Segments Dynamic Allocation** are still available for backward compatibility.

When you select **Rollback Segments Definitions**, the following type of output will be displayed:

```
All Rollback Segments
Segm Name        Ownr  In TabSpace
File containing header of rbs
SYSTEM           Priv  SYSTEM
/exlibris/oradata/aleph1/aleph1_system_01.dbf
_SYSSMU1$        Publ  UNDOTS
/exlibris/oradata/aleph1/aleph1_undots_01.dbf
_SYSSMU2$        Publ  UNDOTS
.
.
.

Online Rollback Segments:
Name     NrEx  Size         Init  Next  PctI  MinE  MaxE Opt   Stat
                                                         size
```

```
SYSTEM       6  376K      114,688                    1  ####       OnL
_SYSSMU      3  1144K     131,072                    2  ####       OnL
_SYSSMU      3  1144K     131,072                    2  ####       OnL
_SYSSMU      4  2168K     131,072                    2  ####       OnL
_SYSSMU      3  1144K     131,072                    2  ####       OnL
_SYSSMU      3  1144K     131,072                    2  ####       OnL
_SYSSMU      3  1144K     131,072                    2  ####       OnL
_SYSSMU      3  1144K     131,072                    2  ####       OnL
_SYSSMU      3  1144K     131,072                    2  ####       OnL
_SYSSMU      3  184K      131,072                    2  ####       OnL
_SYSSMU      3  1144K     131,072                    2  ####       OnL
```

UTIL O/18/3 Rollback Segments Dynamic Allocation

In **Oracle 9*i***, there is a new feature named **Automatic Undo Management (AUM)**, which simplifies and automates the management of undo segments. When **AUM** is used, the rollback segments are defined and managed by Oracle and called **UNDO segments**. Utilities **O/18/2 Rollback Segments Definitions** and **O/18/3 Rollback Segments Dynamic Allocation** are still available for backward compatibility.

```
NAME   EXT   RSSIZE      WRITES  SHRN    AVGSHR  WRAPS  CUREXT  WAITS
SYST     6   385024        7012     0         0      1       3      0
_SYS     4  2220032     1801856     1   1048576      8       2      0
_SYS     3  1171456     1500476     0         0      6       2      0
_SYS     4   253952     1624992     1    327680     28       2      0
_SYS     4   253952     1870178     2    294912     32       3      0
_SYS     4   253952     1642924     2    196608     29       2      0
_SYS     3  1171456     1426884     0         0      5       2      0
_SYS     3  1171456     1865062     1   1048576      5       2      0
_SYS     3  1171456     1462950     1   1048576      6       2      0
_SYS     3  1171456     1613910     0         0      6       2      0
_SYS     3  1171456     1851606     0         0      6       2      0
```

UTIL O/18/4 View Long Operations

This utility displays Oracle long operations, if they occur in the system at the time the utility is run. The following information is displayed:

SID - Session identifier
OPNAME - Operation name
TARGET - The object on which the operation is being performed
DONE SO FAR - Percentage of work already done

Use `<CTRL> + C` to stop the display.

UTIL O/18/5 IO Statistics

This utility displays the following information:

```
TO STOP DO ctrl C
```

```
BLOCK_GETS CONSISTENT_GETS PHYSICAL_READS BLOCK_CHANGES CONSISTENT_CHANGES

5360720    16859217       153081         4975011        15065
5364774    16869224       153118         4978618        15065
5368592    16877862       153138         4981732        15065
5375440    16891538       153180         4986583        15065
5378493    16898409       153196         4988801        15065
```

BLOCK_GETS - Block gets for this session
CONSISTENT_GETS - Consistent gets for this session
PHYSICAL_READS - Physical reads for this session
BLOCK_CHANGES - Block changes for this session
CONSISTENT_CHANGES – Consistent changes for this session

UTIL O/18/6 Sort Operations

This utility displays sort operations if they occur in the system when the utility is running.

Use <CTRL> + C to stop the display.

UTIL O/19 Shared Pool

```
O.19. Shared Pool
       --------------------------------
       0. Exit procedure
       1. Show SGA Buffers
       2. Flush Shared Pool
```

UTIL O/19/1 Show SGA Buffers

This utility shows the various SGA buffers.

```
NAME                            BYTES
--------------------------  ----------
fixed_sga                       730424
buffer_cache                 167772160
log_buffer                      787456
errors                           13864
enqueue                         408944
KGK heap                          7000
KQR L PO                       1228856
KQR M PO                        514224
KQR S SO                          3880
sessions                        608400
sql area                      10699040

NAME                            BYTES
--------------------------  ----------
1M buffer                      2098176
KGLS heap                      2456504
PX subheap                      177176
parameters                        7448
free memory                  117650816
transaction                     399696
PL/SQL DIANA                    688568
FileOpenBlock                  1563248
PL/SQL MPCODE                   204712
library cache                  8620560
miscellaneous                  8793184

NAME                            BYTES
--------------------------  ----------
KGSK scheduler                  108552
KGSKI schedule                   21008
MTTR advisory                    55312
PLS non-lib hp                    2088
joxs heap init                    4240
kgl simulator                  1315752
sim memory hea                  166272
table definiti                   15384
trigger defini                   55360
trigger inform                    1040
trigger source                    1216
```

```
NAME                               BYTES
MTTR advisory                      55312
PLS non-lib hp                      2088
joxs heap init                      4240
kgl simulator                    1315752
sim memory hea                    166272
table definiti                     15384
trigger defini                     55360
trigger inform                      1040
trigger source                      1216

NAME                               BYTES
--------------------------  ----------
Checkpoint queue                 1026560
VIRTUAL CIRCUITS                  470080
dictionary cache                 3229952
KSXR receive buffers             1034000
message pool freequeue            940944
KSXR pending messages que         853952
event statistics per sess        2325600
fixed allocation callback            552
PX msg pool                      1966080
free memory                     65142784
```

UTIL O/19/2 Flush Shared Pool

You will be prompted for the ALEPH_DBA user/password.

UTIL O/20 Multi Threaded Server

```
  O.20. Multi Threaded Server
        ----------------------
         0.  Exit Procedure
         1.
         2.  Show Listener Services
```

In a standard Oracle configuration a separate server process is created on behalf of each user process. It is called a **dedicated server process** (or **shadow** process), because it acts only on behalf of the associated user process.

Oracle also supports the Shared Server Architecture (or Multi Threaded Server Architecture - MTS) in which there are several server processes, each serving several user processes.

In ALEPH, the MTS infrastructure exists in the database but is only implemented in special cases in coordination with Ex Libris.

UTIL O/20/2 Show Listener Services

```
           MTS Dispatchers by Instance
           ===========================



Service "aleph1" has 1 instance(s).
  Instance "aleph1", status UNKNOWN, has 1 handler(s) for this
service...
    Handler(s):
Service "aleph1.il-aleph02.exlibris-int.il" has 1 instance(s).
  Instance "aleph1", status READY, has 9 handler(s) for this
service...
    Handler(s):
      "D007" established:2 refused:0 current:2 max:1022 state:ready
         DISPATCHER <machine: il-aleph02.exlibris-int.il, pid: 1654>
         (ADDRESS=(PROTOCOL=ipc)(KEY=#1654.1))
      "D006" established:7 refused:0 current:0 max:1022 state:ready
         DISPATCHER <machine: il-aleph02.exlibris-int.il, pid: 1652>
         (ADDRESS=(PROTOCOL=ipc)(KEY=#1652.1))
      "D005" established:6 refused:0 current:3 max:1022 state:ready
         DISPATCHER <machine: il-aleph02.exlibris-int.il, pid: 1650>
         (ADDRESS=(PROTOCOL=ipc)(KEY=#1650.1))
      "D004" established:3 refused:0 current:0 max:1022 state:ready
         DISPATCHER <machine: il-aleph02.exlibris-int.il, pid: 1648>
         (ADDRESS=(PROTOCOL=ipc)(KEY=#1648.1))
      "D003" established:5 refused:0 current:3 max:1022 state:ready
         DISPATCHER <machine: il-aleph02.exlibris-int.il, pid: 1646>
         (ADDRESS=(PROTOCOL=tcp)(HOST=il-aleph02.exlibris-
int.il)(PORT=32796))
      "D002" established:3 refused:0 current:0 max:1022 state:ready
         DISPATCHER <machine: il-aleph02.exlibris-int.il, pid: 1644>
         (ADDRESS=(PROTOCOL=tcp)(HOST=il-aleph02.exlibris-
int.il)(PORT=32795))
      "D001" established:13 refused:0 current:0 max:1022 state:ready
         DISPATCHER <machine: il-aleph02.exlibris-int.il, pid: 1642>
         (ADDRESS=(PROTOCOL=tcp)(HOST=il-aleph02.exlibris-
int.il)(PORT=32794))
      "D000" established:3 refused:0 current:3 max:1022 state:ready
         DISPATCHER <machine: il-aleph02.exlibris-int.il, pid: 1640>
         (ADDRESS=(PROTOCOL=tcp)(HOST=il-aleph02.exlibris-
int.il)(PORT=32792))
```

2.3. 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 (for example, 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

2.3.1. 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 `$alephe_root` directory and open the `aleph_start` 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.

2.3.2. 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 1: If the local and remote aleph password are identical, type `<Enter>` as a default. Otherwise enter the remote aleph password.

   Note 2: If the remote aleph password is changed, the link must be dropped and recreated with the new password.

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.aleph1
   ```

4. Use **UTIL A/17/5/2 Create/Recreate All Library Synonyms** 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 Display Records from Datafiles** that you can see the table from the remote instance.

2.3.3. 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. Remove the 6th column (`<hostname>.<sid>`) from the local **file_list**.

2. Use **UTIL A/17/5/2 Create/Recreate All Library Synonyms** to recreate all library synonyms.

3. Change **TAB/RS** to **NA** and delete all columns from column 3 onwards.
   Leave only `NA <table_name>` or totally remove the line from local **file_list**.

2.3.4. 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 the service **p_file_03 Export Database Tables** 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/16/2 Create Database Link** 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 1: If the local and remote aleph password are identical, type `<Enter>` as a default. Otherwise enter the remote aleph password.

Note 2: If the remote aleph password is changed, the link must be dropped and recreated with the new password.

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.aleph1
   ```

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

6. Optional. Upload table data. Use the service **p_file_04 Import Database Tables - Without Checks** to upload a flat datafile 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/17/5/2 Create/Recreate All Library Synonyms** 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 Display Records from Datafiles** that you can see the table from the remote instance.

2.3.5. 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 the service **p_file_03 Export Database Tables** to build a flat file from an existing table.

2. Remove the column <*host*>.<*sid*> from the local **file_list**, if RS change to TAB and add needed columns for TAB option.

3. Use **UTIL A/17/5/2 Create/Recreate All Library Synonyms** to recreate all library synonyms.

4. Use **UTIL A/17/1 Drop & Create Table and Index** in your local host to create the table in your local host.

5. Optional. Upload table data. Use the service **p_file_04 Import Database Tables - Without Checks** to upload a flat datafile into a newly created table.

6. Check in your local host using **sqlplus aleph/aleph** that you can desc <*lib*>.<*table*>.

7. Check in your local host using **UTIL F/4 Display Records from Datafiles** that you can see the table.

2.4. Working in a TWO_TASK Environment

The TWO_TASK functionality enables working with the ALEPH application on one server and the Oracle database on a different server. The server on which the database is located is referred to as the “remote host” and the server on which ALEPH is installed is referred to as the “local host”.

The Oracle server software has to be installed on the database server.

The Oracle client software has to be installed both on the database server, and on the ALEPH server.

ALEPH software has to be installed only on the ALEPH server.

Note that using TWO_TASK entails having some network overhead, which is significant in batch jobs but not when working online (for example with the WEB server).

The option of running ALEPH batch jobs on the database server should be taken into consideration. If this option is chosen, ALEPH software must also be installed on the database server.

To support TWO_TASK configuration, make sure that the following variables appear in the `$alephe_root/aleph_start` file:

```
setenv    aleph_db      <remote host>.<oracle_SID>
```

```
      setenv    TWO_TASK      <remote host>.<oracle_SID>
```

The `<remote host>` is the name of the remote server and `<oracle_SID>` is the name of the Oracle database on that server.

On the local host the following lines should appear in
`$ORACLE_HOME/network/admin/tnsnames.ora`

```
<remote host>.<oracle_SID>=(description=
                (address=
                        (protocol=tcp)
                        (host=<full remote host name>)
                        (port=1521))
                (connect_data=(service_name=<oracle_sid>)))
```

When all definitions are complete, shut down Oracle and ALEPH, reboot the server and test. During the test you should ensure that the only ALEPH activity on the local server will be the WEB server activation and use, and no other ALEPH and/or Oracle activity.

2.4.1. An Example of TWO_TASK Definitions

ram01.exlibris.co.il	**local host**
ram02.exlibris.co.il	**remote host**
aleph12	**DB on remote host**

1. On `ram02` - install oracle and create DB `aleph12`.

2. On `ram01` perform the following:

2.1. Log in to ALEPH.
In `$alephe_root/aleph_start` perform the following modifications:

```
#         setenv   aleph_db          ${ORA_HOST}.${ORACLE_SID}
          setenv   aleph_db          ram02.aleph12
          setenv   TWO_TASK          ram02.aleph12
```

2.2. After completion of all modifications log out and log in again in order for the changes in `aleph_start` to take effect.

2.3. As user **oracle** - add the following lines in
`$ORACLE_HOME/network/admin/tnsnames.ora`:

```
ram02.aleph12=(description=
                (address=
                        (protocol=ipc)
                        (key=aleph12))
                (address=
                        (protocol=tcp)
                        (host=ram02.exlibris.co.il)
                        (port=8003))
                (connect_data=(service_name=aleph12)))
```

2.4. Shutdown Oracle and ALEPH, reboot the server and test ALEPH on `ram01`.