DigiTool
DIGITAL ASSET MANAGEMENT

DIGITOOL VERSION 3.0

System Administrator's Guide

Ex Libris

Last Update: November 16, 2006
Document Version 1.4
Code: D-ver3-SAG-1.4

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.

Any references in this document to non-Ex Libris Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this Ex Libris product and Ex Libris has no liability for materials on those Web sites.

Documentation produced November 2006
Document version 1.4

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

DIGITOOL VERSION 3.0

System Administrator's Guide – General Overview

© Ex Libris Ltd., 2006
Last Update: April 26, 2006
Document Version 1.1
Code: D-ver3-SAGOver-1.1

CONFIDENTIAL INFORMATION

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

This document is provided for limited and restricted purposes in accordance with a binding contract with Ex Libris. The information herein includes trade secrets and should all be considered 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, errors and omissions excepted.

Any references in this document to non-Ex Libris Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this Ex Libris product and use of those Web sites is at your own risk.

Documentation produced April 2006

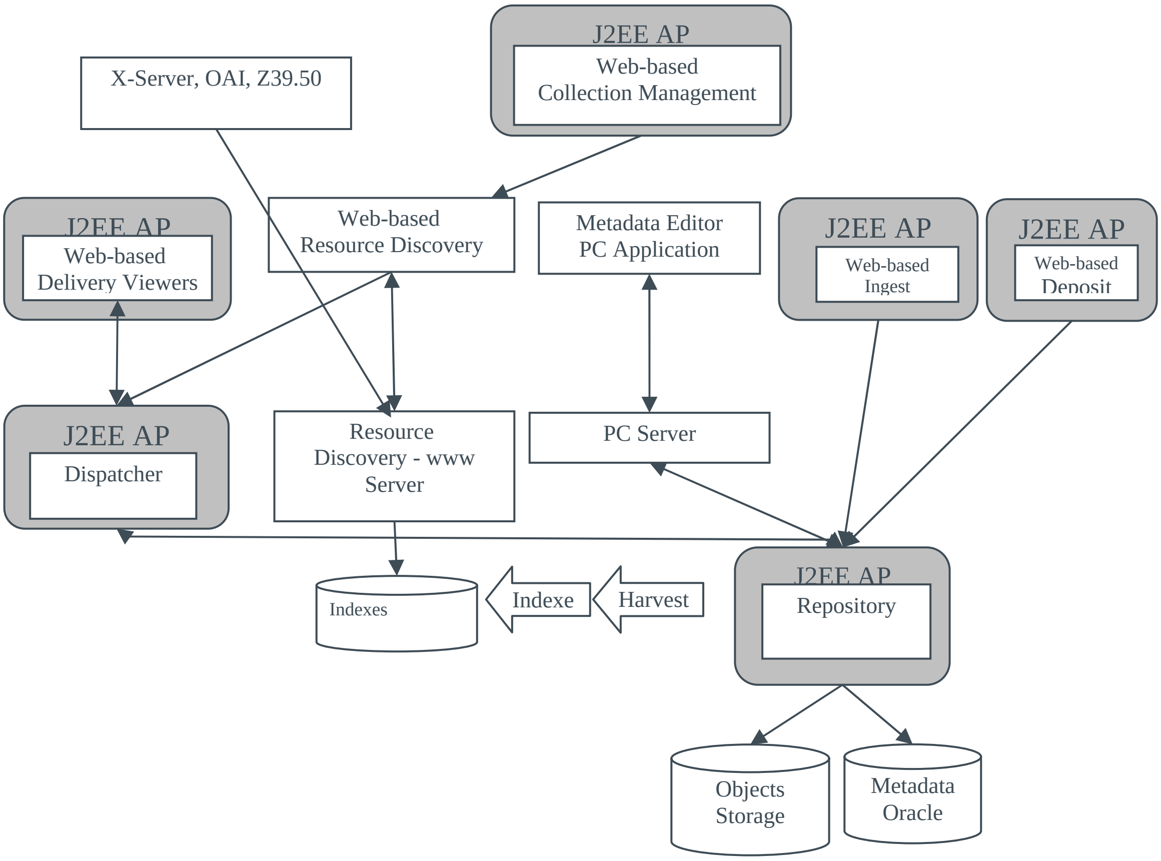
Document version 1.1

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

Table of Contents

1 DigiTool general architecture

System Components – Technology



The DigiTool product can be divided into two main application groups – the Repository group and the Resource discovery and Metadata Editing group.

1.1 Repository group

The repository group is implemented by a group of J2EE applications, all making use of the JBoss server.

All applications have access to the repository database implemented on Oracle, making use of tables in the REP00 schema. See *System Administration Guide – Database* Section.

1.2 Resource Discovery and Metadata Editing group

The Resource Discovery and Metadata Editing group (RD) is implemented by:

- Servers – WWW server, PC server, Z39.50 server
- Client application
- Harvesting workflow
- Batch queue processes
- Job daemon process
- Indexing process

1.2.1 Servers – WWW server, PC server, Z39.50 server

A group of applications accessing the Resource Discovery database implemented on Oracle, making use of tables in following schemas: vir01, ext01, gen01, dat01 and a schema for each Admin. Unit.

The Resource Discovery and Metadata Editing servers communicate with the JBoss applications using SOAP XML, activating Web Services.

1.2.2 Client application

A C++ application installed on a client's PC, communicating with the PC server using an internal protocol. The user of this application can find objects in the repository and edit their metadata as well as change the stream file attached to them. New objects can be added, duplication of objects is possible as well as deletion. All actions are on specific objects or metadata records, not on groups of them.

1.2.3 Harvesting workflow

Harvesting - a periodically run job that harvests metadata along with Full text extractions and thumbnails of objects from the Repository system into the Resource Discovery system (into a Silo named GEN01).

New objects, created since the last harvest, are pulled into the RD, and so are objects that have been modified since the last harvest.

Harvesting is done according to configuration files (harvesting_schema.xml, q1 q2 q3 matching harvest version from the data_tab of gen01) and does not necessarily include all metadata or all objects in the repository.

Any object that is harvested into the Resource Discovery system is indexed every time it enters the system.

Two types of harvesting are currently available - the customer can decide which type is more appropriate for usage and set it to be used as the default in the relevant configuration table. For more details, please consult the *Staff User's Guide – Meditor*.

After harvesting, due to possible change in the Silo content, some refreshing of the collections is done using the Update Logical collections and collection item count - (media-36) batch service.

1.2.4 Batch queue processes

Each Admin Unit and each Silo in the system needs a process called batch_queue. This process is in charge of running jobs that entered the queue by running a service from the Meditor application’s Services menu.

1.2.5 Job daemon process

The job daemon (jobd) is a scheduler for submitting jobs into the batch queue of an Admin Unit/Silo repeatedly.

To configure the run times of jobs, edit $dtle_root/tab/job_list.conf.

You can set lines for weekly and daily runs.

Entering specific runs of jobs is done by adding them to $dtle_root/tab/job_list

Use UTIL E/15 and UTIL E/16 to manage the jobd.

1.2.6 Indexing process

The metadata and full text of objects in the Resource Discovery silo are indexed. This is done by a process named ue_01. This process will index any object shortly after it was harvested.

2 DigiTool system startup and shutdown

Any server with Ex Libris software installed on it will have a file named /etc/initd.d/exlibris. This script runs with a start/stop parameter and it controls the run of the exlibris/startup/start_stop script. This script will start Ex Libris applications according to the configuration in the /exlibris/startup/init.dat file.

In the case of a server with DigiTool software installed, this might contain up to four lines for each DigiTool instance:

- Oracle – Starts up/shuts down the Oracle instance, serving all DigiTool processes, and maybe other Ex Libris components.
- Apache – Starts/stops the Apache server redirecting all WWW requests.
- dtl – Starts up all the DigiTool specific processes, by running the script /exlibris/dtl/d3_<**copy**>/dtle/dtl_startup_all.
- The DigiTool-specific processes:
  - Jboss process - $jdtlh_bin/jboss_startup.sh
  - Resource discovery processes – Web (www_server), PC(pc_server) and Z39.50 (z39_gate) servers, jobd daemon, batch queues (lib_batch, runs for each Admin Unit and Silo), indexing process (rts32 ue_01_a) , all started by $dtle_root/dtl_startup
  - dtl – Shuts down all the DigiTool-specific processes, by running the script /exlibris/dtl/d3_<**copy**>/dtle/dtl_shutdown_all
  - Jboss process - $jdtlh_bin/jboss_shutdown.sh

  - Resource discovery processes - Web, PC and Z39.50 servers, jobd daemon, batch queues, indexing process, all stopped by $dtle_root/dtl_shutdown

3 Rapidly growing storage areas

The following directories will rapidly grow (at different rates) when objects are added to the DigiTool system:

- Repository streams storage area
- Admin units ingest and deposit areas
- Repository indexes area
- Resource Discovery thumbnails and full text extractions area

It is important to check when these areas should be moved to separate disks, and mounted to those directories, to prevent the /exlibris disk from getting filled up and interfering with the normal work of the system.

3.1 Repository streams storage area

You can configure DigiTool storage as follows:

From the Management interface, click the Repository tab, then the Storage tab. This is the environment for configuring DigiTool storage. You can set a maximum storage limit for each Storage group (directory where streams are saved in the Unix tree).

The default root of the default storage directory that any system comes with is /exlibris/dtl/j3_1/digitool/home/profile/storage/.

This value can be changed. The directory can become a mount point of another disk.

Whenever a new storage directory is added, the root directory should be a mount point of a different disk.

3.2 Admin units ingest and deposit areas

Each Admin unit has an area that holds any information related to the deposit and ingest jobs, including all metadata and streams. If old ingest and deposit jobs are never deleted, this area will cause the disk to overflow.

For an Admin unit named ADM01, directory is:

/exlibris/dtl/j3_<**copy**>/digitool/home/profile/units/ADM01

This cannot be changed; rather it should become a mount point to another disk if you consider saving old ingest and deposit jobs for a long time before deleting them.

3.3 Repository indexes area

The index files reside in a directory pointed at by $jdtlh_conf/repository_configuration.xml :

```
<modules>
    <module name=”repository”>
        <parameters>
            <parameter>
                <p_key>index_directory></p_key>
                <p_val>Index_files_root</p_val>
                 <p_key>oracle_index_directory</p_key>
                 <p_val>Oracle_index_files_root</p_val>
                 ……
             </parameter>
             ……
        </parameters>
    </module>
```

The default value of this tag is:

```
/exlibris/dtl/j3_<copy>/digitool/home/profile/work/indexes/d31_rep00
```

This value can be changed. The directory can become a mount point of another disk.

3.4 Resource Discovery thumbnails and full text extractions area

The Resource Discovery system needs the thumbnails of all image type streams (for displaying) and the Full Text Extraction of text type streams (for full text search). Therefore these are copied from the repository system into the RD by the harvesting process, into the following directories:

```
/exlibris/dtl/d3_<copy>/gen01/object_index
```

```
/exlibris/dtl/d3_<copy>/gen01/object_thumbnail
```

The index files of all objects harvested is put in:

```
/exlibris/dtl/d3_<copy>/gen01/object_ctx
```

The directory names cannot be changed but they can be a mount point to another disk.

4 Ongoing maintenance

4.1 Processes running as part of the DigiTool server

This section explains for each process how to start, stop monitor and check logs.

4.1.1 Jboss server

Startup:

```
cd $jdtlh_bin

./jboss_startup.sh
```

Monitor:
Unix ps command will show lines like:

```
dtl      27936 27853  0 14:27 pts/15   00:00:00
/exlibris/dtl/d3_1/product/local/java/bin/java -Xms256m -Xmx512m -
Djava.awt.headless=true -Djava.awt.headless=true -
Ddigitool.home=/exlibris/dtl/j3_1/digitool/home/system/bin/.. -
Djava.library.path=/exlibris/dtl/d3_1/product/ImageMagick-
6.1.9/lib:/exlibris/dtl/d3_1/produc
```

Each line represents a thread of the Java Virtual Machine and not a process.

Note: The ps command might give you less information in each line (less characters).

Note : In some Unix versions you cannot use the –m option, in some you do not need it for thread display.

How many threads ?
By default, Jboss will start up with 25 threads and increase the number, if needed up to a limit of 150. These numbers can be changed in the file :

```
/exlibris/dtl/j3_1/digitool/home/profile/overwrites/thirdparty/opense
rver/server/default/deploy/jbossweb-tomcat.sar/server.xml.tmpl
```

Shutdown:

```
cd $jdtlh_bin
./jboss_shutdown.sh
```

Log file:

```
$jdtl_jb_def_log/server.log
```

4.1.2 Apache process

Startup:

```
cd $dtle_root/apache/bin
./apachectl start
```

Monitor:
Usually there are **8 instances**. Unix ps command will show lines like :

```
dtl      11478     1  0 10:12 ?        00:00:01
/exlibris/dtl/d3_1/product/bin/httpd -d
/exlibris/dtl/d3_1/dtle/apache
```

Shutdown:

```
cd $dtle_root/apache/bin
./apachectl stop
```

Log files:

```
$dtle_root/apache/logs/access_log
$dtle_root/apache/logs/error_log
$dtle_root/apache/logs/mod_jk.log
```

4.1.3 WWW server

Startup:

```
start_w
```

(this also stops WWW server if it is currently running)

Or by UTIL W/3/1

Monitor:

Usually there is one parent WWW server. In Unix ps command output it looks like:

```
dtl      11418     1  0 10:11 ?        00:00:00 /bin/csh –f
/exlibris/dtl/d3_1/dtl/proc/www_server 4881 8881 2
```

And **3 children.** In Unix ps command output, it look likes:

```
dtl      11426 11418  0 10:11 ?        00:00:00
/exlibris/dtl/d3_1/dtl/exe/www_server_main -p 4881 –v
```

You can also check they are running by the command:

```
server_monitor
```

If they are running output will include lines like:

```
| Port    | Pid     | Server Type       | Started At            |
Status
|--------|---------|--------------------|---------------------|----
----------------
| 4881  | 11426 | WWW Server    | Mar 13 10:11:29 | Free
| 4881  | 11431 | WWW Server c | Mar 13 10:11:29 | Free
| 4881  | 20037 | WWW Server c | Mar 13 12:09:03 | Free
```

Shutdown:

```
util w/2/2.
```

Log file:

```
$LOGDIR/www_server_488<copy>.log
```

4.1.4 PC server

Startup:

```
start_pc
```

(this also stops PC server if it is currently running)

Or by UTIL W/3/3

Usually there is one parent PC server. In Unix ps command output it looks like:

```
dtl      20379 22056  0 12:15 pts/8    00:00:00 /bin/csh -f
/exlibris/dtl/d3_1/dtl/proc/pc_server 6881
```

Monitor:

And **5 children.** In Unix ps command output it looks like:

```
dtl      20504 20379  0 12:15 pts/8    00:00:00
/exlibris/dtl/d3_1/dtl/exe/pc_server_main 0 6881
```

You can also check they are running by the command

```
server_monitor
```

If they are running output will include lines like:

```
| Port      | Pid       | Server Type        | Started At           |
Status

|----------|----------|--------------------|---------------------|-
-------------------

| 6881    | 20504   | PC Server           | Mar 13 12:15:58 | Free

| 6881    | 20515   | PC Server c0      | Mar 13 12:15:58 | Free

| 6881    | 20516   | PC Server c0      | Mar 13 12:15:59 | Free

| 6881    | 20517   | PC Server c0      | Mar 13 12:15:59 | Free

| 6881    | 20518   | PC Server c0      | Mar 13 12:15:59 | Free

| 6881    | 20519   | PC Server c0      | Mar 13 12:15:59 | Free
```

Shutdown:

```
util w/2/4.
```

Log file:

```
$LOGDIR/pc_server_688<copy>.log
```

4.1.5 Z39 Gate server

Startup:

```
Util w 3/5
```

Monitor:

There is one Z39.50 gate server. In Unix ps command output it looks like:

```
dtl           13233         1    0  Mar11  ?               00:00:00
/exlibris/dtl/d3_1/dtl/exe/z39_gate  -a-  -t /exlibris/dtl/d3_1/tmp  -
vlog   -r60   -c/exlibris/dtl/d3_1/dtle/tab/z39_gate/z39_gate.conf   -
m/exlibris/dtl/d3_1/dtle/tab/z39_gate/z39_gate_error_list           -
n/exlibris/dtl/d3_1/dtle/tab/z39_gate/z39_target_error_list         -
o/exlibris/dtl/d3_1/d
```

You can also check it is running by the command

```
server_monitor
```

If it is running output will include lines such as:

```
| Port      | Pid       | Server Type        | Started At
| Status

|----------|----------|--------------------|---------------------|-
-------------------

| 7881     | 13233  | Z39 Gate            | Mar 11 21:35:04 | Free
```

Shutdown:

```
util w/2/5
```

Log file:

```
z39_gate_788<copy>.log
```

4.1.6 Batch queue processes

Startup:

How to stop a batch queue of an Admin unit/silo named ABC01?

```
dlib ABC01
util c/3
```

Shutdown:

How to start a batch queue of an Admin unit/silo named ABC01?

```
dlib ABC01
util c/2
```

Monitor:

Assuming we have two Admin Units called ADM01 and ADM01, and one Silo GEN01, the output of the Unix ps command is as follows:

```
dtl     12483     1  0 Mar11 ?        00:00:00
/exlibris/dtl/d3_1/dtl/exe/lib_batch GEN01

dtl     13272     1  0 Mar11 ?        00:00:00
/exlibris/dtl/d3_1/dtl/exe/lib_batch ADM01

dtl     13299     1  0 Mar11 ?        00:00:00
/exlibris/dtl/d3_1/dtl/exe/lib_batch ADM02
```

Check a batch queue of Admin unit/silo ABC01 is running?

```
dlib ABC01
util c/1
```

Logfile:

```
dlib ABC01
$data_scratch/run_b.<pid of batch queue>
```

4.1.7 job daemon (jobd)

Startup:

```
$dtl_exe/jobd
```

Or

```
util e/16/9 – restart, read job_list and job_list.conf
util e/15/1 – start
```

Shutdown:

```
util e/15/2
```

Monitor:

```
util e/15/3
```

```
util e/16/1, e/16/5 – editing job_list and job_list.conf
```

Log file:

```
$dtle_root/ jobd.log
```

or

```
util e/15/4
```

4.1.8 Indexing in silo – ue_01

Startup:

```
util e/1
```

Monitor:

```
util c/1
```

In the output of the Unix ps command :

```
dtl       5423     1  0 Mar13 pts/8    00:00:25
/exlibris/dtl/d3_1/dtl/exe/rts32 ue_01_a GEN01.d3_1
```

Shutdown:

```
util e/2
```

Log file:

```
dlib gen01
```

```
$data_scratch/ run_e_01.<pid> (look for the newest file starting with
run_e_01 ).
```

4.2 What should be run by the jobd job_list and job_list.conf?

To handle the job_list and job_list.conf file you can either:

```
cd $dtle_root/tab
```

And vi the files

Or use the utility:

util e/16/1 and util e/16/5 and then restart by util e/16/9

By default, we have set in the job_list.conf:

```
W1    W YYYYYYY

D1    D 00:00 22:00 02:00

D2    D 00:20 22:00 02:00

W1 – Run every day of the week.
```

```
D1 – Run at 00:00, every 2 hours, until 22:00
D1 – Run at 00:20, every 2 hours, until 22:00
```

By default we have set in the job_list :

```
W1 00:00:00 Y                          GEN01 optimize_z91_context_index     GEN01
W1 00:00:00 Y                          GEN02 optimize_z91_context_index     GEN02
W1 00:00:00 Y                          REP00 optimize_object_index          REP00
W1 00:00:00 Y                          REP00 optimize_md_index              REP00
```

Once a day, at 00:00, run processes to optimize the indexing on the RD system (Silos gen01, gen02) and on the Repository system (for metadata as well as digital entities)

```
00 01:00:00 Y                          DTL02 util_x_06                      no_questions
```

Clean up apache_media used by RD system for displaying thumbnails.

```
00 D1       Y                          GEN01 p_harvest_02                   GEN01,A
```

Every Sunday, according to D1 (00:00-22:00 every 2 hours), run harvesting of type 2, in update mode on gen01. For each day of the week we have a line like this.

```
00 D2       Y                          GEN01 p_media_36                     GEN01,D
```

Every Sunday, according to D2 (00:20-22:00 every 2 hours), run p_media_36 in update mode on gen01.

```
00 23:00:00 Y                          DAT01 util_a_13_b                    DAT01
```

On Sunday at 23:00 drop the statistics table (Z34).

```
00 01:00:00 Y                          VIR01 clear_vir01                    VIR01
```

On Sunday, at 01:00, remove all tables and sequences of VIR01 schema (holding temporary data like session info, result sets for searches, and so on).

4.3 Which Repository maintenance jobs should be run periodically?

4.3.1 Maintenance jobs

In the DigiTool management interface, choose the “Maintenance” tab, “Jobs” sub tab, Filter by “Maintenance Jobs”.

Any of these jobs can be run on the whole repository, on one Admin unit, or many subsets of the Admin Unit, according to the staff permissions and arguments they give the job.

To view the report created by the job you ran, move to “Reports” sub tab and choose the HTML or XML version of the appropriate report.

Here are our recommendations for running these jobs:

Remove old Ingest activities:

This job should be run about once a month, or more often when a massive upload to the system is done. This is done for reasons of disk space and might be run less frequently if the deposit and ingest areas are mounted on a separate disk. See “Admin Units Ingest and Deposit areas”.

Remove old Deposit activities:

This job should be run about once a month, or more often when a massive upload to the system is done. This is done for reasons of disk space and might be run less frequently if the deposit and ingest areas are mounted on a separate disk. . See “Admin Units Ingest and Deposit areas”.

Delete Unlinked Metadata Objects:

This job should be run once every 3-6 months. It is a cleanup of Metadata objects that are not pointed to by any Digital Entities. It is not a critical job.

Checksum:

This job should be run 1-2 times a month. It checks if the checksum stored in the DigiTool Metadata is identical to the actual streams checksum.

The results can indicate if any streams are manipulated by non-DigiTool authorities, and might indicate a possible security problem.

4.3.2 Report jobs

In the DigiTool management interface, choose the “Maintenance” tab, “Jobs” sub tab, Filter by “Report Jobs”.

To view the report created by the job you ran, move to “Reports” sub tab and choose the HTML or XML version of the appropriate report.

Count Repository Objects:

This report gives us an indication of the number of objects we have in the Admin units, number of metadata by types and unlinked metadata. It is recommended to run this report once a week for the first months of the repository life, and later on, once a month.

Count Repository Streams:

This report gives us an indication of the number of streams in the repository, by file extension, Admin Unit and location type (local or remote). It is recommended to run this report once a week for the first months of the repository life, and later on, once a month.

Digital Entities Viewing reports:

This is a group of reports checking the actual viewing of streams in the system. It is recommended to run these reports once a week for the first months of the repository life, and later on, once a month.

DIGITOOL VERSION 3.0

System Administrator's Guide - JBoss

Last Update: January 10, 2006
Document Version 1.0
Code: D-ver3-SAGJBoss-1.0

CONFIDENTIAL INFORMATION

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

This document is provided for limited and restricted purposes in accordance with a binding contract with Ex Libris. The information herein includes trade secrets and should all be considered 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, errors and omissions excepted.

Any references in this document to non-Ex Libris Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this Ex Libris product and use of those Web sites is at your own risk.

Documentation produced January 2006.

Document version 1.0

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

Table of Contents

1 General

The purpose of this document is to describe the J2EE-based implementation of DigiTool 3.0. This document is not a JBoss guide. For more information about JBoss, refer to: http://www.jboss.com/products/jbossas

The Java portion of DigiTool 3.0 is J2EE-ompliant software that uses JBoss 3.2 as an Application server. JBoss is a J2EE-ompliant application server certificated by Sun Microsystems.

Note
The examples in this document use the following:

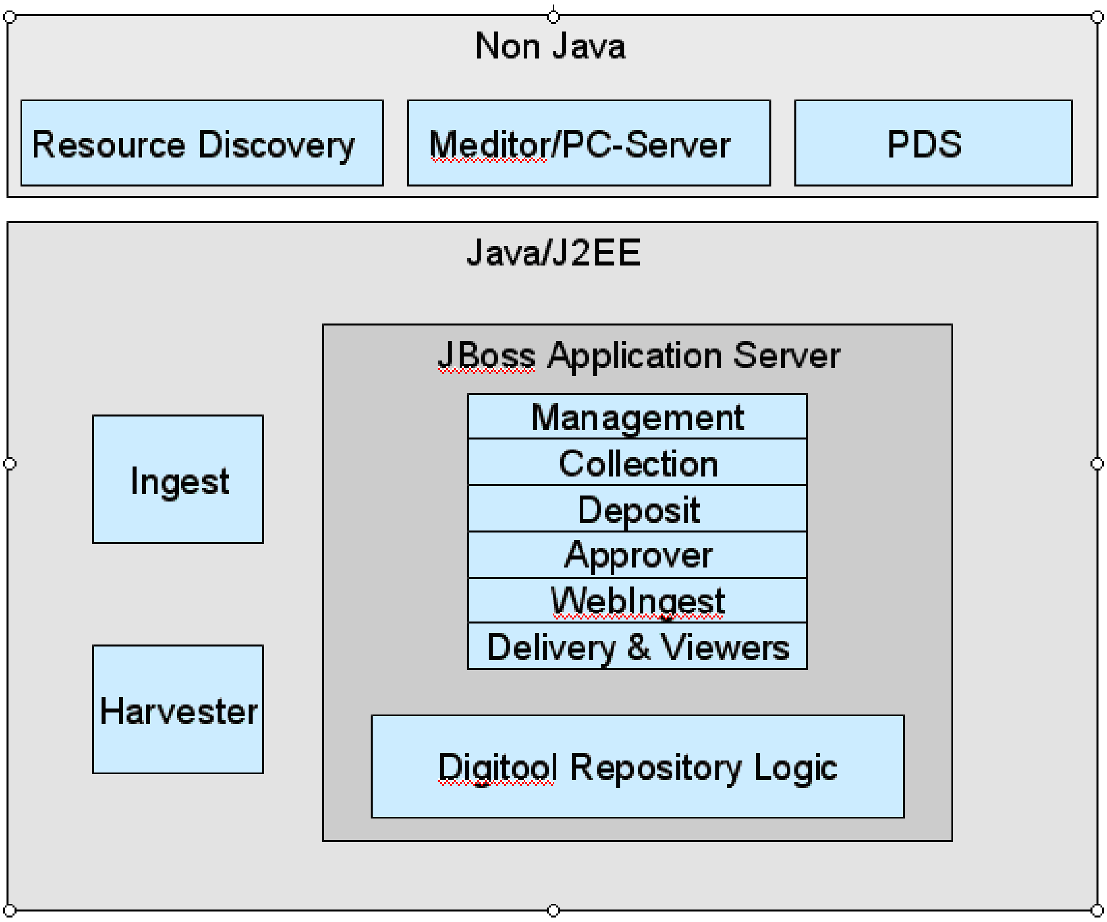
- DigiTool Server Name: dtl-server
- JBoss http port: 1801
- Base installation folders: /exlibris/dtl/d3_1 & /exlibris/dtl/j3_1

2 DigiTool J2EE modules served by JBoss

DigiTool product modules can be divided into two categories:

- Java based – located under /exlibris/dtl/j3_1/
- Non Java – located under /exlibris/dtl/d3_1/

The following diagram shows the main DigiTool modules and the category they belong to:



DigiTool Repository

The DigiTool repository is a back-office module that provides a set of RMI and SOAP services for object storing/maintenance.

The list of web services and their WSDL files can be found under

http://dtl-server:1801/de_repository_web/services

DigiTool Delivery System

Provides a single URL for object access:

http://dtl-server:1801/webclient/DeliveryManager?pid=1001

Web Interfaces

DigiTool Web interfaces are listed below; note that every module has its own UI / Logic / Data layers.

- Management (http://dtl-server:1801/mng)
- Collection (http://dtl-server:1801/collection)
- Webingest (http://dtl-server:1801/webingest)
- Deposit (http://dtl-server:1801/deposit)
- Approver (http://dtl-server:1801/approver)
- Webingest (http://dtl-server:1801/webingest)

- Viewers (http://dtl-server:1801/view)
- MetsViewer (http://dtl-server:1801/metsviewer)
- JPEG 2000 Viewer (http://dtl-server:1801/metsviewer)

3 DigiTool J2EE console module

In addition to the modules that run under JBoss, there are some stand-alone modules that run as console applications in a separate process. Usually, those console application communicate with JBoss using RMI.

The console applications have a related .sh script (in home/system/bin) that executes them.

Examples:

Ingest/Tasker – activated by ingest.sh/tasker.sh

Harvester module used by the silo harvesting module – harvester.sh

4 Directory structure

The Java-related components in DigiTool are located under the /exlibris/dtl/j3_1/digitool/home directory.

The /exlibris/dtl/j3_1/digitool/home.orig directory is an inactive directory that is used for the service pack application process. The home.orig holds a vanilla replica of the DigiTool software.

The table below describes the directory structure under the home directory.

Path (relative to home directory)	Description
profile	The profile directory contains the customer's local settings and data. This document does not cover the structure of this directory.
profile/overwrites	Contains templates of configuration files (will be detailed/discussed later on.)
system	The system directory is the active software directory, The customer should not change any file in this directory directly. All changes should be implemented from the overwrites directory and initiated through the "set_globals" mechanism (will be described later on in this document)
system/bin	Scripts directory, this directory holds the JBoss startup/shutdown scripts in addition to other console applications scripts

Path (relative to home directory)	Description
system/conf	The conf directory holds configuration files such as DigiTool_repository_configuration.xml, repository_indexing_schema.xml, mime.types and more
system/conf/i18n	Holds configuration files for i18n (internationalization) support in the Web interfaces.
system/xsd	DigiTool product defines some XML schemas such as digital_entity.xsd & exlibris_mets_profile.xsd
system/xsl	Holds a set of XSL files that DigiTool uses for XML transformations. For example, digital_entity/descriptive_dc.xsl is used for DublinCore metadata display in the Web interfaces
system/client	Holds a set of jars (can be considered like binary files). Those jar files are used by the console application (for example, ingest.sh)
system/thirdparty	This directory contains the third-party components that DigiTool uses, for example, jhove for technical metadata extraction and j2k for JPEG 2000 support. JBoss's third-party components are also located under this directory
system/thirdparty/openserver	The JBoss home directory
system/thirdparty/openserver/ server/default	JBoss can run using different configuration settings (like apache and httpd.conf). Each configuration is actually a directory that holds setting/ J2EE components/logs, and so on. The "default" directory is the configuration directory that DigiTool uses. Note that "secure" & "web" directories are also JBoss configuration directories but they are not in use.
system/thirdparty/openserver/ server/default/deploy	The deploy directory contains the DigiTool J2EE-compliant software. This directory is the interface between the application server and the J2EE compliant application – DigiTool in our case.
system/thirdparty/openserver/ server/default/log	Log directory of the JBoss-based modules (Repository, Webingest, and so on) Will be detailed later on in this document.
system/thirdparty/openserver/ server/default/deploy/	This directory is actually a package of all other modules (In J2EE words this is the Enterprise Archive – the .ear file)

Path (relative to home directory)	Description
DigiTool-mng.ear	Under this directory you will find a set of .war directories. Each of the .war directories is a stand-alone Web application. For example, the digitool-metsviewer.war is the metsviewer Web application. In other words the digitool-mng.ear directory is the interaction directory between JBoss and DigiTool software.

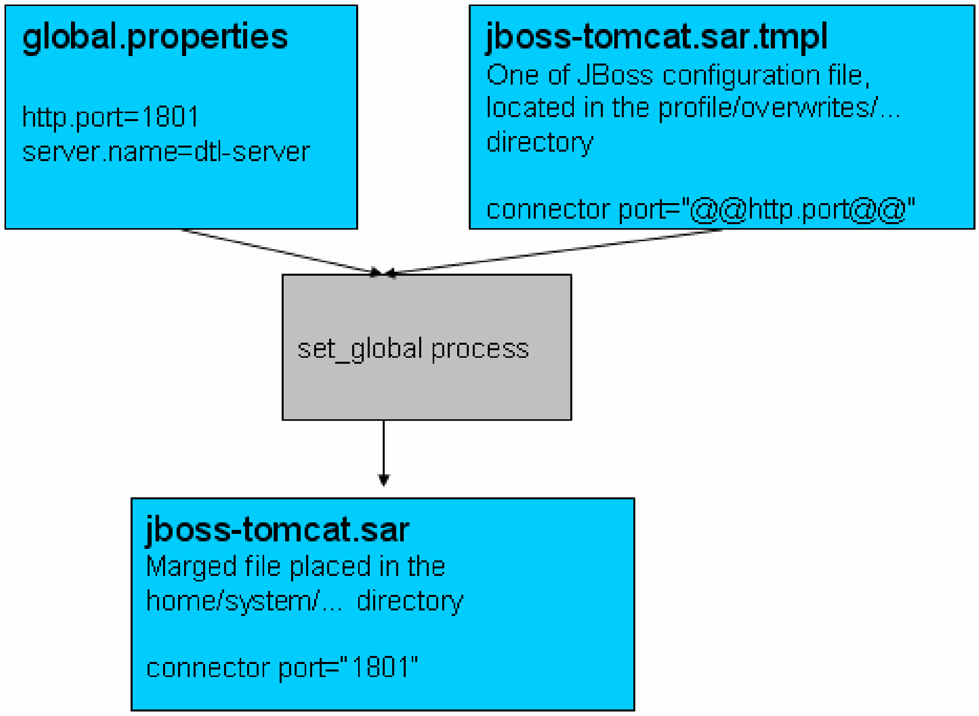
5 Configuration and Customization

As mentioned above, do not change files directly under the home/system directory. All changes should be done indirectly using the set_globals mechanism.

A manual setup such as the machine name/port setting can become difficult – it requires a very good understanding of both DigiTool and JBoss configuration files. The set_globals mechanism comes to simplify this process by combining all parameters to a single file – *home/profile/global.properties*.

The set_globals process is very simple - it merges the parameter values from the global.properties file with configuration file templates (.tmpl files) and copies them to the system directory.

The .tmpl files are located in the profile/overwrites directory.


global.properties
http.port=1801
server.name=dtl-server
jboss-tomcat.sar.tmpl
One of JBoss configuration file, located in the profile/overwrites/... directory
connector port="@@http.port@@"
set_global process
jboss-tomcat.sar
Marged file placed in the home/system/... directory
connector port="1801"

5.1 List of global.properties parameters

Parameter Name	Default Value	Description
dbconnection.url	jdbc:oracle:thin:@dtl-server:1521:dtl3	The jdbc connection string, used by the transaction manager
dbconnection.username	D31_rep01	Database user name
dbconnection.password	D31_rep01	Database user password
server.name	localhost	Server name should be the external server name for example, www.DigiTool.com
http.port	1801	Port used for all Web interfaces (except Resource Discovery)
http.connectiontimeout	20000	HTTP session timeout (in seconds), a login window will display if the user was inactive for more than http.connectiontimeout
jndi.port	2801	Naming server ports, EJB clients should use this port.
rmi.port	3801	JBoss Internal
rmi.object.port	3901	JBoss Internal
webservice.port	4801	JBoss Internal
debug.port	5001	Use for Java remote debug – for use only when the server starts in debug mode.
server.pool.port	6801	JBoss Internal
ajp.port	7801	This port is used when the Apache-Tomcat connector (mod_jk) was set.
pds.port	8881	The port of the PDS server.
pds.server	localhost	PDS server name

Parameter Name	Default Value	Description
index.directory	\\dtl-server\exlibris\dtl\j3_1\digitool\home\profile\work\indexes\d31_repXX	Relevant only when the server starts in development mode (running under Windows)
oracle.index.directory	/exlibris/dtl/j3_1/digitool/home/profile/work/indexes/d31_repXX	Defines where the repository index files should be placed
index.files.per.directory	1000	Define how many files should be placed in each index directory (to avoid huge directories
text.viewer.directory	/exlibris/dtl/d3_1/apache_media	Used by the TextViewerPreProcessor.,
Temporary text files are placed in this Web-enabled directory		
image.magick.lib	/exlibris/dtl/d3_1/product/ImageMagick-6.1.9/lib	Path to image magik lib directory, used by the Ingest module (for thumbnail creation) and by the image viewer (zoom-in/zoom-out)
jpeg2000.lib	/exlibris/dtl/d3_1/product/jpeg2000.sdk/sdk/lib	Path to jpeg2000 lib directory, used by the Ingest module (for thumbnail creation) and by the image viewer (zoom-in/zoom-out)
java.home	/exlibris/dtl/d3_1/product/j2sdk1.4.2_04	Path to the JDK, as you can see that the JBoss is using the JDK from the product tree.
activex_upload_lic	-	License key for the file uploading Active-X component (used in the Webingest interface)

In addition, the set_globals process also copies “standard” (non .tmpl files) files from the profile/overwrites folder to the relevant active system folder.

For example, the `profile`*`/overwrites/conf/mime.types`* will be copied to `system/conf/mime.types` during `set_globals`.

5.1.1 Basic customization

Basic customization involves modifying the globals.properties parameters.

Steps

1. Edit the global.properties file.
2. Type j_bin (this alias will move you to the home/system/bin directory. stop JBoss - jboss_shutdown.sh
3. run set_globals.sh
4. start JBoss - jboss_startup.sh

5.1.2 Advanced customization

Advanced customization involves customizing the configuration files in the system/conf directory.

Steps

1. Edit one of the configuration files listed in the table below.
2. Restart JBoss - jboss_shutdown.sh followed by jboss_startup.sh Note that changes related to the repository_indexing_schema.xml and repository_jobs_configuration.xml do not require restarting; you can reload the new configuration using the “Reload repository configuration” jobs in the Management interface.

The table below lists the configuration files and a short description of each. It is not within the scope of this document to describe all the configuration options in detail. Please refer to related documentation.

Configuration File	Description
repository_indexing_schema.xml	This file controls the internal indexes of the repository. Two main indexes can be customized using this file – 1. The digital entity index 2. The metadata index. When searching the repository using the management interface, those indexes are used.
mimetypes_configuration.xml	Defines the icons that will be used for each MIME type and entity type.
repository_jobs_configuration.xml	Defines the jobs and the jobs forms, that can be executed using the Management interface

Configuration File	Description
	(Management tab)
mime.types	Map between MIME types and file extensions. The repository uses this file to resolve the file extension for given MIME types and vice versa
mets_configuration.xml	Defines the mapping between METS metadata types and the digital entity metadata types stored in DigiTool. In addition, contains some definitions for the behavior of the METS viewer
repository_stream_handler.xml	Defines the task and task chains used by the Webingest module
ExLibMessageFile.properties	Defines all error/warning/information codes and text that are used by DigiTool modules. For example: *35005=Delivery rule cannot be empty.*
i18n/locale_codes	Maps locale code from 2-digit to 3-digit and vice versa. For example, *eng=en*
i18n/messages.properties	Defines the Web interface messages. For example: *c.login.username=User Name*
i18n/dynamic_messages.properties	Defines the Web interface dynamic messages. For example: *i.sys.chain.1=Empty Chain*

5.1.3 Customization of the repository_configuration.xml (.tmpl)

The repository_configuration.xml file is a special file that has a template in the overwrites folder. This template should be updated directly and set_globals should be used to initiate the changes into the active repository_configuration.xml file in system/conf.

Steps

1. Edit the home/profile/overwrites/conf/repository_configuration.xml.tmpl file.

2. Type j_bin (this alias will move you to the home/system/bin directory. stop jboss - jboss_shutdown.sh
3. run set_globals.sh
4. start jboss - jboss_startup.sh

6 System log

By default, all JBoss-related logs are located under the following directory: `system/thirdparty/openserver/server/default/log`.

Use the “jb_log” alias if you want to change the current directory to the log directory.

DigiTool uses the log4j framework. More information about this framework can be found at http://logging.apache.org/log4j/docs/.

There are three types of log files:

- boot.log
- localhost_access_log
- server.log

6.1 boot.log

This is the JBoss microkernel startup log; it contains information about microkernel startup, class path, environment parameters.

6.2 localhost_access_log

localhost_access_log.<date>.log – http access log, every http request – GET/POST is written to this log file, this log file is very similar to the Apache HTTPD server access log.

JBoss uses Tomcat as a Web container implementation. This log is actually a Tomcat log. For example:

Client Ip, Date & Time, HTTP Method, URL, Http Version Http return code, msg length

```
10.1.235.71 - - [01/Jan/2006:12:08:11 +0200] "GET /mng/css/loginStyles.css HTTP/1.1" 200 6350
10.1.235.71 - - [01/Jan/2006:12:08:11 +0200] "GET /mng/css/generalStyles.css HTTP/1.1" 200 43888
```

The HTTP access log is rolled over daily; the date is part of the log file name. For example:

localhost_access_log.2005-12-31.log

localhost_access_log.2006-01-01.log

This log file can be customized/tuned by editing the following file:

`home/profile/overwrites/thirdparty/openserver/server/default/deploy/jbossweb-tomcat.sar/server.xml.tmpl`

6.3 server.log

This is the main log file for JBoss activity. DigiTool modules that run under JBoss write to this log file.

A new log file is created – rolled over - each and every new day. When the server is restarted, a timestamp is added. For example:

```
-rw-rw-r--    1 dtl       exlibris   174985 Dec 29 23:57 server.log.2005-12-
29

-rw-rw-r--    1 dtl       exlibris 17546674 Dec 30 23:57 server.log.2005-12-
30

-rw-rw-r--    1 dtl       exlibris   281074 Dec 31 23:57 server.log.2005-12-
31

-rw-rw-r--    1 dtl       exlibris  5698992 Jan  1 11:09 server.log.0101.1050

-rw-rw-r--    1 dtl       exlibris    36822 Jan  1 10:51 server.log.0101.1100

-rw-rw-r--    1 dtl       exlibris    36822 Jan  1 11:01 server.log.0101.1108

-rw-rw-r--    1 dtl       exlibris   142407 Jan  1 14:25 server.log
```

This log shows that the server was up from 29/12 – 31/12 (rolled over). On 01/01 the server was restarted at 10:50, 11:00, 11:08.

The server.log log is the current active log file. Using "tail –f server.log" can be useful when debugging the server. For example:

```
Date, Time, Thread, Log Level, class name, Message

2005-11-06                       19:10:17,988                       INFO
[com.exlibris.DigiTool.delivery.DeliverySessionsMng]   Cleaning   old
delivery session - 1131296885128.

2005-11-06                       19:15:32,941                       INFO
[com.exlibris.DigiTool.repository.jobs.IndexFileSynchronizerJob]
found 8 items to index

2005-11-06                       19:15:33,425                       INFO
[com.exlibris.DigiTool.repository.jobs.IndexFileSynchronizerJob]
Indexing DIGITAL ENTITY:4530 at 484ms

2005-11-06                      19:08:49,128                       ERROR
[com.exlibris.DigiTool.delivery.DefaultAccessRightsChecker]  Fail  to
parse  access  right  metadata.  Exception:  error:  The  element  type
"xb:access_right_md"  must  be  terminated  by  the  matching  end-tag
"</xb:access_right_md>". Cause:

org.xml.sax.SAXParseException: The element type "xb:access_right_md"
must be terminated by the matching end-tag "</xb:access_right_md>".
```

Each message in the log file has a set of fixed header information, and a dynamic message.

Column Name	Description
Date	The date of the message in YYYY-MM-DD format.
Time	The time of the message in HH:MM:SS format
Thread	This information is very important when debugging because all JBoss threads write to this log. Filtering the log using a thread ID shows us the event sequence

	of a single thread.
Message Type	INFO,WARN & ERROR
Class	The Java class that outputs this messages
Free Message	A free text that describes the message.

The home/system/conf/ExLibMessageFile.properties file defines the log messages.

```
50005=Unable to create Ingest jobs log, ingest id = {0}, exception =
{1} 50006=Removed ingest activity {0} from the database.

50007=Removed ingest {0} from the disk.
```

The “{X}” indicates a placeholder for a dynamic parameter.

The server.log can be configured by editing the log4j configuration file. home/profile/overwrites/thirdparty/openserver/server/default/conf/log4j.xml

Use this file to control the log entry format pattern, disable enable log message, and define the log level (INFO/EARN/ERROR) for each Java package.

7 Database connectivity

The connection between DigiTool applications and the Oracle database is done in a J2EE manner. A data source is deployed. This example is a DigiTool data source defined in the home/system/thirdparty/openserver/server/default/deploy/digitool-ds.xml file:

```
<?xml version="1.0" encoding="UTF-8"?>


<datasources>

  <local-tx-datasource>

    <jndi-name>jdbc/DigiToolDS</jndi-name>

    <connection-url>jdbc:oracle:thin:@dtl-
server:1521:dtl3</connection-url>

    <driver-class>oracle.jdbc.driver.OracleDriver</driver-class>

    <user-name>d31_rep01</user-name>

    <password>d31_rep01</password>

  </local-tx-datasource>

</datasources>
```

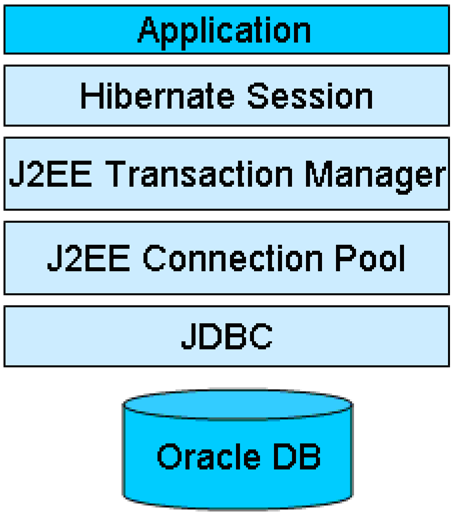
DigiTool does not uses the J2EE Entity Bean for persistence, but the Hibernate framework is used instead. Refer to http://www.hibernate.org for more information.

The Hibernate configuration file is located in home/system/conf/hibernate.cfg.xml. This file holds the Hibernate configuration parameters such as:

- What is the JNDI name of the data source?
- What is the name of the transaction manager?

- Whether to write the SQL of each transaction to the log or not?

The diagram below shows the layers between the application layer and the database.



8 JBoss administration interfaces.

JBoss has two Web-enabled administration interfaces – a Web console and a JMX console.

The Web console is a user-friendly subset of the JMX console. For more information about these consoles refer to JBoss documentation: http://www.jboss.com/docs/index

Web Console URL

http://dtl-server:1801/web-console/index.html

JMX Console URL

http://dtl-server:1801/jmx-console

Both interfaces are password-protected.

The default user and password are user=*digi* and password=*tool*.

The default user and password can be changed by editing the following file:

home/profile/overwrites/thirdparty/openserver/server/default/deploy/jmx-console.war/WEB-INF/classes/jmx-console-users.properties.tmpl

In addition, for security reasons, one can disable those interfaces entirely by deleting the following directories:

home/system/thirdparty/openserver/server/default/deploy/management

home/system/thirdparty/openserver/server/default/deploy/jmx-console.war

Note
If for security reasons the customer deleted these directories, this will need to be performed after each Service Pack application which will automatically reinstall the directories.

The management and jmx-console.war directories can also be restored manually from the home.orig directory.

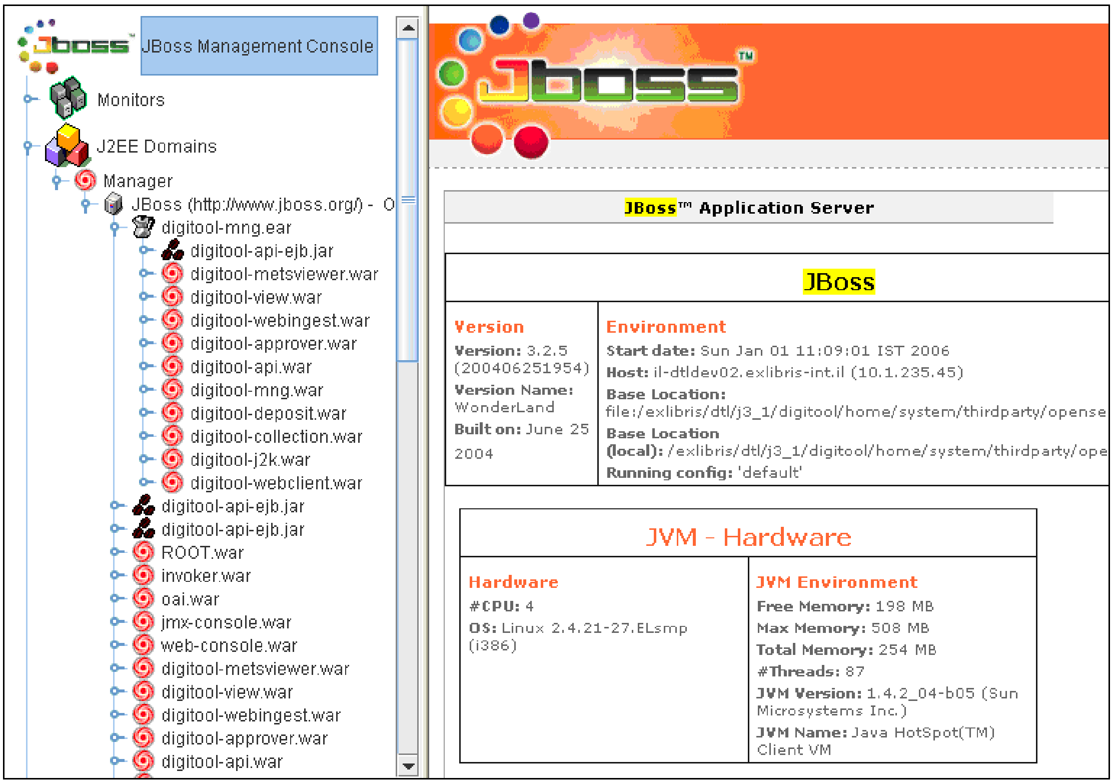
8.1 JBoss Web console

The Web Console interface, is an applet-based interface. In order to view the interface a JRE component should be installed on the client machine. (Can be downloaded for free from the Sun Microsystems Web site - http://java.sun.com/j2se/1.5.0/download.jsp)

Just to give you an impression – here are some screenshots.

Opening page-information about the deployed module can be viewed in the left pane. In the screenshot you can see the digitool-mng.ear that contains all Web interfaces.

System information about the status of the server can be viewed in the right pane:



JBoss™ Application Server

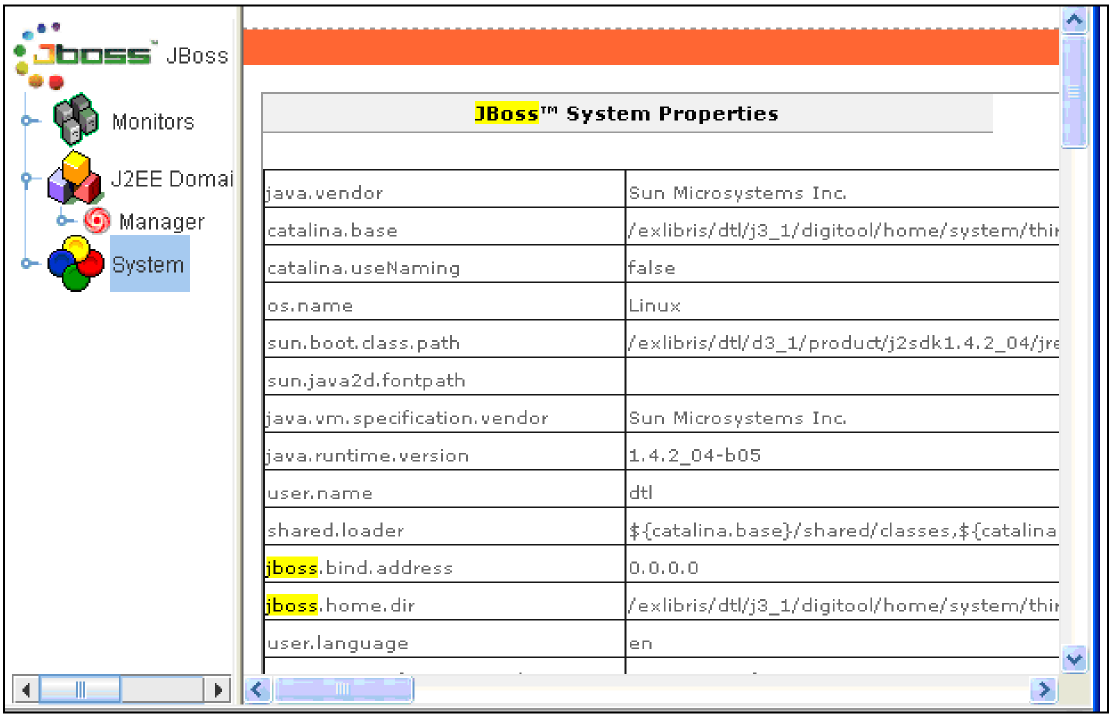
JBoss

Version	Environment
Version: 3.2.5 (200406251954) **Version Name:** WonderLand **Built on:** June 25 2004	**Start date:** Sun Jan 01 11:09:01 IST 2006 **Host:** il-dtldev02.exlibris-int.il (10.1.235.45) **Base Location:** file:/exlibris/dtl/j3_1/digitool/home/system/thirdparty/opense **Base Location (local):** /exlibris/dtl/j3_1/digitool/home/system/thirdparty/ope **Running config:** 'default'

JVM - Hardware

Hardware	JVM Environment
#CPU: 4 **OS:** Linux 2.4.21-27.ELsmp (i386)	**Free Memory:** 198 MB **Max Memory:** 508 MB **Total Memory:** 254 MB **#Threads:** 87 **JVM Version:** 1.4.2_04-b05 (Sun Microsystems Inc.) **JVM Name:** Java HotSpot(TM) Client VM

Selecting “System” from the left pane will display a list of JBoss system parameters.



JBoss™ System Properties

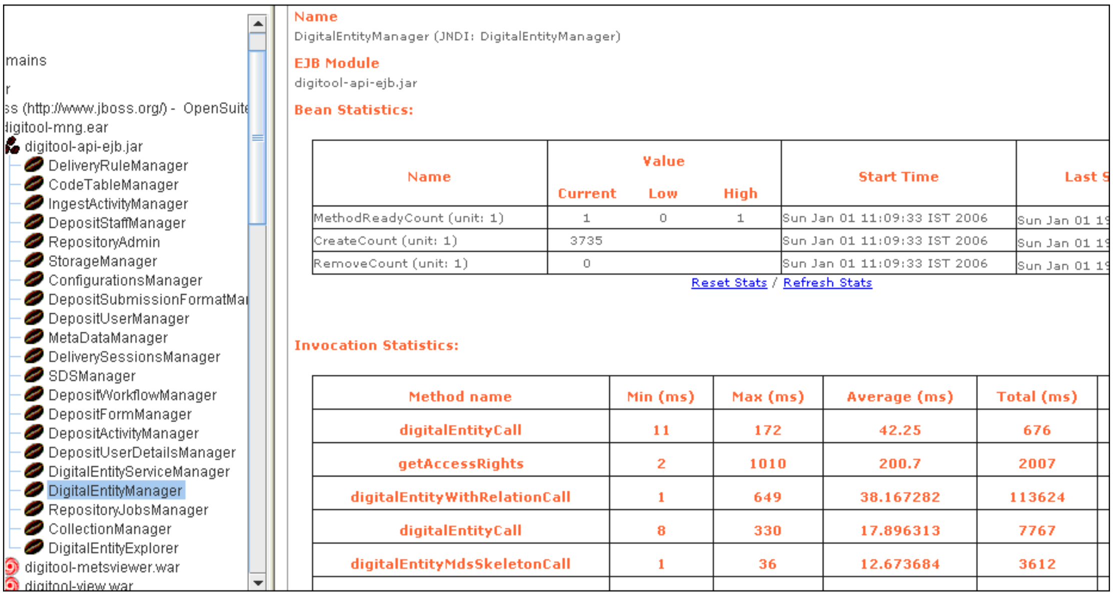
java.vendor	Sun Microsystems Inc.
catalina.base	/exlibris/dtl/j3_1/digitool/home/system/thir
catalina.useNaming	false
os.name	Linux
sun.boot.class.path	/exlibris/dtl/d3_1/product/j2sdk1.4.2_04/jre
sun.java2d.fontpath	
java.vm.specification.vendor	Sun Microsystems Inc.
java.runtime.version	1.4.2_04-b05
user.name	dtl
shared.loader	${catalina.base}/shared/classes,${catalina
jboss.bind.address	0.0.0.0
jboss.home.dir	/exlibris/dtl/j3_1/digitool/home/system/thir
user.language	en

By selecting the digitool-api-EJB.jar module in the left pane, you can view all RMI/EJB objects.

By selecting one of the objects, related information is displayed in the right pane.

For each object method, the following performance counters are displayed:

- Min (ms): minimum execution time of the method
- Max (ms) maximum execution time of the method
- Average (ms) average execution time of the method
- Total (ms) total execution time of all calls together
- #Invocations total number of invocation since sever start



Name

DigitalEntityManager (JNDI: DigitalEntityManager)

EJB Module

digitool-api-ejb.jar

Bean Statistics:

Name	Value			Start Time	Last S
	Current	Low	High		
MethodReadyCount (unit: 1)	1	0	1	Sun Jan 01 11:09:33 IST 2006	Sun Jan 01 19
CreateCount (unit: 1)	3735			Sun Jan 01 11:09:33 IST 2006	Sun Jan 01 19
RemoveCount (unit: 1)	0			Sun Jan 01 11:09:33 IST 2006	Sun Jan 01 19

Reset Stats / Refresh Stats

Invocation Statistics:

Method name	Min (ms)	Max (ms)	Average (ms)	Total (ms)
digitalEntityCall	11	172	42.25	676
getAccessRights	2	1010	200.7	2007
digitalEntityWithRelationCall	1	649	38.167282	113624
digitalEntityCall	8	330	17.896313	7767
digitalEntityMdsSkeletonCall	1	36	12.673684	3612

9 Troubleshooting

Here are some common problems and suggested solutions.

1.Q. JBoss is up and running but when trying to log into one of the Web interfaces with a correct user and password, a *User doesn't exist* message appears.

1.A. JBoss uses the PDS and the dtl server for patron and staff authentication. Make sure that both servers are up and running.

2.Q. JBoss does not start correctly; error messages dealing with connection to Hypersonic database appear in the server.log.

2.A. JBoss has some cache folders that in a rare situation can be corrected; this can be solved by simply clearing the cache directories (they will be re-created automatically.)

1. Run jboss_shutdown.
2. Delete the following directories:

   home/system/thirdparty/openserver/server/default/data

   home/system/thirdparty/openserver/server/default/work

   home/system/thirdparty/openserver/server/default/tmp

3.Q JBoss does not start correctly; error messages dealing with serialized sessions appear in the server.log

3.A see 2.A

4.Q JBoss fails to stop/start; server.log has errors messages dealing with bind address/addresses already in use.

4.A In rare situations a JBoss process can be stacked in a “zombie” state. In this case it should be “killed“ manually

Run “ps -ef | grep /dtl/j3_1” and kill all listed processes.

5.Q JBoss looks like it is running, but nothing really works with database-related errors appearing in the Web interfaces.

5.A There is a good chance that JBoss failed to connect to the database.

- Browse the server.log if you see Oracle database-related error messages.
- Check that Oracle is up and running. Type s+ rep00
- Check that the jdbc listener is up and running.
- Check the connection information in the home/system/thirdparty/openserver/server/default/deploy/digtool-ds.xml file
- Check that there is enough table space in the database. Use UTIL O.

DigiTool

DIGITAL ASSET MANAGEMENT

DIGITOOL VERSION 3.0

System Administrator’s Guide – Resource Discovery and Metadata Editing Overview

Ex Libris

Last Update: April 27, 2006
Document Version 1.0
Code: D-ver3-SAGRDMEOver-1.0

CONFIDENTIAL INFORMATION

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

This document is provided for limited and restricted purposes in accordance with a binding contract with Ex Libris. The information herein includes trade secrets and should all be considered 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, errors and omissions excepted.

Any references in this document to non-Ex Libris Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this Ex Libris product and use of those Web sites is at your own risk.

Documentation produced April 2006

Document version 1.0

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

Table of Contents

1 Resource Discovery and Metadata editing System Architecture Overview

The Resource Discovery application enables users to search for, discover and organize for their own use, objects that are part of a repository. The interface includes means of search, object organization in collections and private folders.

The metadata editor (Meditor) facilitates inserting and modifying object metadata relations and stream ref.

DigiTool's resource discovery and metadata editing architecture is based on a multitier, client/server model. Client/Server communication is based on a stateless (self-contained) transaction model - nonetheless, DigiTool Application Servers are continuously connected (with timeout) to the database, to ensure high performance.

DigiTool's resource discovery and metadata editing architecture is based on a scaleable, distributed logic model and relies on an object-oriented design. Two key features of DigiTool's architecture are:

> **Multitier, Client/Server model** - DigiTool resource discovery and metadata editing is split into logical segments with a clearly defined interface based on message passing. See *Figure 1* below.
>
> **Modularity** - The key notion of the distributed logic design that underlies DigiTool resource discovery and metadata editing is modularity - both vertical (between tiers) and horizontal (within the tiers). This ensures that the system is easy to maintain and extend, and that new technologies and concepts can be quickly integrated.

DigiTool can be installed on the following platforms:

- SUN SOLARIS
- LINUX REDHAT

1.1 DigiTool Server Architecture

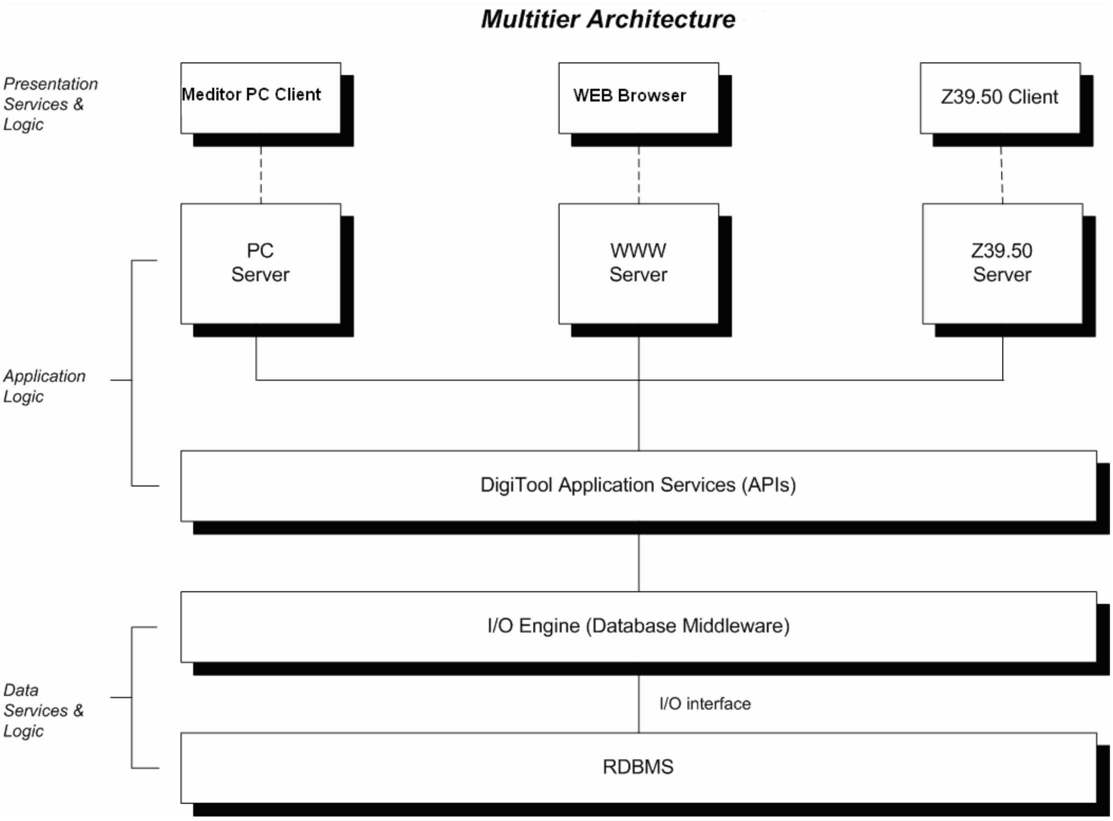


Figure 1

The DigiTool server is composed of the following tiers:

1.1.1 Presentation Services & Logic

Provides the interface with which the user interacts.

1.1.2 Application Logic

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

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

1.1.3 Data Services & Logic

DigiTool Database Middleware (or I/O Engine) - This is a high-level database management layer. A logical server provides data services to the application services objects. It contains a group of objects that intermediates between the application and the database. The I/O engine translates an application request to a sequence of database commands.

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

DigiTool Database - The DigiTool database runs under Oracle RDBMS.

2 DigiTool Unix Logins

The DigiTool server requires the following Unix users:

Login	**Home Directory**	**Description**
`dtl`	/exlibris/dtl	DigiTool administrator
`oracle`	`/exlibris/app/oracle/product/920`	Oracle administrator (DBA)

The person who is in charge of all the repositories, the System Administrator, can log in as `dtl`, and thus will be able to modify the parameters and data of all repositories.

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

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

3 DigiTool Directory Structure

3.1 The dtl Root Directory

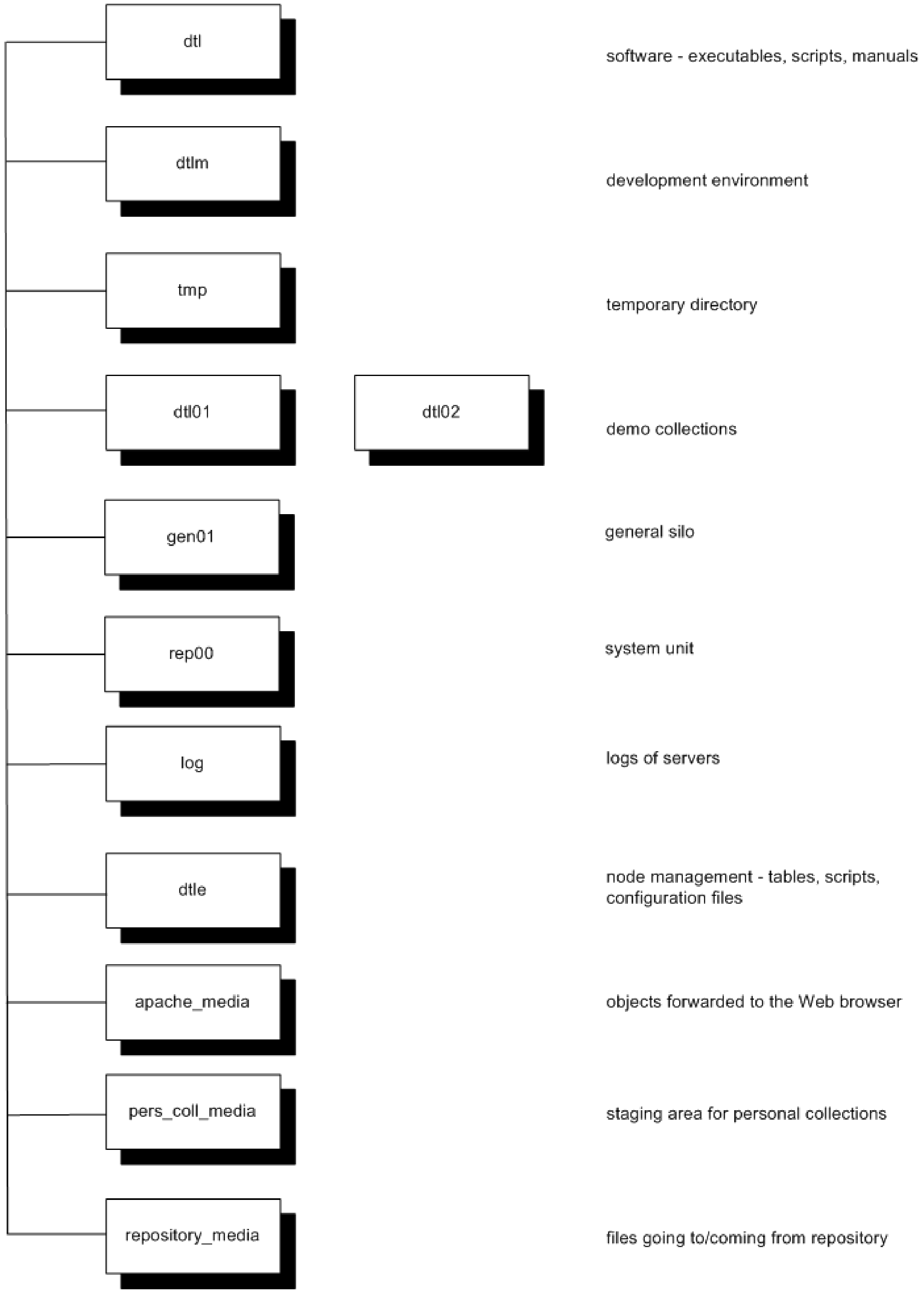

dtl
software - executables, scripts, manuals
dtlm
development environment
tmp
temporary directory
dtl01
dtl02
demo collections
gen01
general silo
rep00
system unit
log
logs of servers
dtle
node management - tables, scripts, configuration files
apache_media
objects forwarded to the Web browser
pers_coll_media
staging area for personal collections
repository_media
files going to/coming from repository

Figure 2

A digitool root is composed of three types of directories:

DigiTool software dtl, dtlm, pds, tmp, log

Staging areas apache_media,repository_media,pers_coll_media

Demo Admin Units/ Silos.

dtl01 – Admin Unit 1

dtl02 – Admin Unit 2

gen01 – General Silo.

3.1.1 Node management

As part of the installation phase, the dtle directory (node management) is relocated to a different directory (for example d3_1). This directory will normally contain site-specific Admin Units/Silos as well. A sample scheme is given in Figure 3.

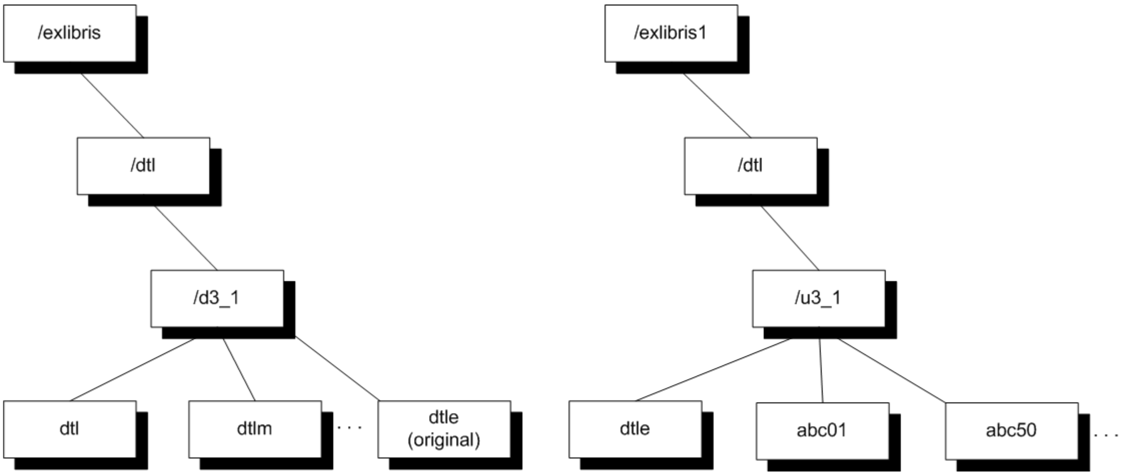


Figure 3

This structure supports the ability to upgrade the DigiTool software (under d3_1) without interfering with local customization (under u3_1).

In addition to the core software directories dtlm and pds, the system has directories that include management tables and parameter files pertaining to the specific installation (dtle), and to each of the Admin Units and Silos.

3.2 The dtle Directory

The primary directories in the `dtle` tree are:

`tab`	node management tables
`www_r_<lng>`	HTML files for DigiTool Resource Discovery.
`scratch`	intermediate files and logs
`apache`	`conf/htdocs/logs/bin/icons`
`error_<lng>`	messages that are displayed to the end user
`gate`	Z39.50 setup
`unicode`	tables which translate from and to Unicode
`coll_mng`	Collection management files.

3.2.1 The Main Configuration File - dtl_start

`dtl_start` is DigiTool's main configuration file. It contains definitions of DigiTool Admin Units and Silos (`abc01`, `abc02`, and so on) and environment variables. It also contains logical assignments (the only place with physical references to DigiTool directories). `dtl_start` is in the `dtle` directory (`cd $dtle_root`).

In order for changes in `dtl_start` to take effect, you must exit DigiTool, re-login, and then restart the daemons and servers. Running `dtl_shutdown` stops all daemons and servers. Running `dtl_startup` restarts daemons and servers depending on definitions in `dtl_start` and `dtl_start.private`.

Following are primary portions of `dtl_start`:

Version information

```
setenv    DTL_VERSION       3
setenv    DTL_COPY          1
setenv    DTL_REVISION      00
setenv    DTL_APP_VERSION   d${DTL_VERSION}_${DTL_COPY}
setenv    ORA_USER_PREFIX   D${DTL_VERSION}${DTL_COPY}_
setenv    DTL_SUB_PORT      1
setenv    DTL_MOUNT          /exlibris/dtl
setenv    USER_MOUNT         /exlibris/dtl
```

Host definitions

```
setenv    ORA_HOST          hostname
setenv    IO_HOST           hostname
setenv    Z39_HOST          hostname
setenv    WWW_HOST          hostname
setenv    PDS_HOST          hostname
setenv    DELIV_SYS_HOST    hostname
setenv    JBOSS_HOST        hostname
```

Port definitions

```
setenv    WWW_SERVER_PORT    488${ALEPH_SUB_PORT}
setenv    PC_SERVER_PORT     688${ALEPH_SUB_PORT}
setenv    Z39_GATE_PORT      788${ALEPH_SUB_PORT}
setenv    Z39_SERVER_PORT    988${ALEPH_SUB_PORT}
setenv    HTTPD_PORT         8881
setenv    PDS_PORT           HTTPD_PORT
setenv    DELIV_SYS_PORT     1801
setenv    JBOSS_PORT         1801
setenv    IO_SERVER_PORT     566${DTL_SUB_PORT}
```

Repository definitions

```
setenv    ext01_dev          ${DTL_MOUNT}/d${DTL_VERSION}_$DTL_COPY
setenv    dat01_dev          ${DTL_MOUNT}/d${DTL_VERSION}_$DTL_COPY
setenv    rep00_dev          ${DTL_MOUNT}/d${DTL_VERSION}_$DTL_COPY
setenv    gen01_dev          ${DTL_MOUNT}/d${DTL_VERSION}_$DTL_COPY
setenv    vir01_dev          ${DTL_MOUNT}/d${DTL_VERSION}_$DTL_COPY
setenv    dtl01_dev          ${DTL_MOUNT}/d${DTL_VERSION}_$DTL_COPY
setenv    dtl02_dev          ${DTL_MOUNT}/d${DTL_VERSION}_$DTL_COPY
```

Objects transitions definitions

```
setenv    REPOSITORY_MEDIA
${DTL_MOUNT}/d${DTL_VERSION}_$DTL_COPY/repository_media
setenv    APACHE_MEDIA
${DTL_MOUNT}/d${DTL_VERSION}_$DTL_COPY/apache_media
setenv    PERS_COLL_MEDIA
${DTL_MOUNT}/d${DTL_VERSION}_$DTL_COPY/pers_coll_media
```

Queues startup definition

```
setenv    QUE_STARTUP_LIBS      "dtl01 dtl02 gen01"
```

Daemons startup definitions

```
setenv    WORD_STARTUP_LIBS     "gen01"
setenv    REQUEST_STARTUP_LIBS  "dtl01 dtl02"
setenv    ACC_AUT_STARTUP_LIBS  ""
setenv    MESSAGE_STARTUP_LIBS  ""
```

Server's startup definitions

```
setenv    Z39_GATE_STARTUP      Y
setenv    Z39_SERVER_STARTUP    N
```

Oracle definitions

```
setenv    TNS_PORT           1521
setenv    ORACLE_SID         dtl39

setenv    ORACLE_OWNER       oracle9
setenv    ORACLE_VERSION     92
setenv    ORACLE_CONF        ${DTL_MOUNT}/ora_dtl

setenv    NLS_LANG           American_America.UTF8
setenv    NLS_SORT           Binary
setenv    NLS_COMP           Binary


setenv    ORACLE_BASE        /exlibris/app/oracle
setenv    ORACLE_HOME        ${ORACLE_BASE}/product/920
setenv    ORACLE_ALERT_LOG
         $ORACLE_BASE/admin/$ORACLE_SID/bdump/alert_${ORACLE_SID}.log
setenv    DEFAULT_TS         TS0
setenv    TEMPORARY_TS       TEMP

setenv    ora_connect_mode   NO_LISTENER

setenv    dtl_db           ${ORA_HOST}.${ORACLE_SID}
```

DigiTool environment

```
setenv  dtl_dev     ${DTL_MOUNT}/d${DTL_VERSION}_$DTL_COPY
setenv  jdtl_dev    ${DTL_MOUNT}/j${DTL_VERSION}_$DTL_COPY
setenv  user_dev    ${USER_MOUNT}/u${DTL_VERSION}_$DTL_COPY
setenv  dtlm_dev    ${DTL_MOUNT}/d${DTL_VERSION}_$DTL_COPY
setenv  dtle_dev    ${USER_MOUNT}/u${DTL_VERSION}_$DTL_COPY
setenv  dtle_synch_dev
                    ${DTL_MOUNT}/d${DTL_VERSION}_$DTL_COPY
setenv  dtl_utf          /tmp
setenv  dtl_product
                 ${DTL_MOUNT}/d${DTL_VERSION}_$DTL_COPY/product
setenv  TMPDIR
                 ${DTL_MOUNT}/d${DTL_VERSION}_$DTL_COPY/tmp
setenv  LOGDIR
                 ${DTL_MOUNT}/d${DTL_VERSION}_$DTL_COPY/log
setenv  PROCESS_NUMBER     $$
setenv  FILE_TMPDIR        $TMPDIR
```

Apache definitions

```
setenv    apache_dir         ${aleph_product}/local/apache
setenv    httpd_bin          ${aleph_product}/local/apache/bin
setenv    httpd_root         $dtle_dev/dtle/apache
```

Additional customization

```
#  Use dtl_start.private to define customer Admin Units and Silos.

        if (-f  $dtle_dev/dtle/dtl_start.private) then
                source $dtle_dev/dtle/dtl_start.private
        endif
```

Note
This template is already set up so that the Table of Contents will start on an odd page. Only lines that are above the following line can be configured:

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

Everything under this line is hard-coded.

3.3 DigiTool Admin Units/Silos Structure

Under `digitool`, you can find **Admin Units** as well as **Silos**. Each has its own directory under the dtl tree as well as a separate Oracle User with tables of data.

Under these directories you can find:

Primary files :

`file_list`	Configuration file for Oracle tables, indexes and objects
`prof_library`	Admin Unit/Silo logical assignments

The primary directories are:

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

On top of these directories, for all **Admin Units,** you will find the following directories :

`pers_coll_forms/`	Default location for personal collection forms.
md/	Meta Data information

And for Silos, you will find the following directories :

`object_thumbnail/`	Default location for storing a Silo’s thumbnail images.
`object_index/`	Default location for storing a Silo’s full-text index-related data.
`object_ctx/`	Default location for storing all Silos index related data.

4 Utilities

Each Admin unit/Silo’s database is implemented within Oracle as a separate Oracle user. Each Oracle user owns a set of tables that contains the Admin unit/Silo’s data.

Each Admin Unit/Silo has access to the DigiTool utilities by activating the `util` command from the prompt. The **UTIL** main screen will appear:

```
            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. GUI MEDITOR 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
    S. Statistics
    W. Server management (Monitor, Stop, Start, Log files)
    X. Clean up
    Y. Node management
Please select [exit]:
```

4.1 UTIL C - Monitor batch jobs

```
Monitor batch jobs
       ------------------
        0. Exit procedure
        1. Check unit running process
        2. Start unit batch queue
        3. Stop unit batch queue
        4. Check unit lock status
        5. Lock unit
        6. Unlock unit
        7. Show queue of jobs waiting execution
        8. Delete entry from queue of jobs waiting execution
        9. Alter run-time of job waiting execution
       10. Display logfile of a batch job
       11. Display log list of batch jobs

Please select [exit]:
```

4.2 UTIL X - Clean Up

```

X. Clean Up
       ---------------------
0.  Exit Procedure
        1.  Clean dtle_scratch
        2.  Clean data_scratch
        3.  Clean data_files
        4.  Clean TMPDIR
        5.  Clean Server Log Files
        6.  Clean apache_media
        7.  Clean utf_files Directory
        8.  Clean Virtual Admin Unit (VIR01)
        9.  Clean data_print
        10. Clean repository_media
        11. Clean pers_coll_media
        12. Clean jboss log files

Please select [exit]:
```

Figure 4

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

Note
Before running the utilities, you might need to close DigiTool servers, batch queues and daemons.

5 Servers, Daemons, Batch Jobs and Problem Diagnosis

There are various DigiTool servers and daemons:

- www_server
- PC server
- ue_daemons
- Batch jobs

The conventional servers' port numbers are:

WWW server (Web)	488x
PC server (GUI)	688x
Z39.50 gate	788x
Z39.50 server	988x
Apache	888x
PDS	888x
DELIV_SYS	180x
JBOSS	180x
IO_SERVER_PORT	566x

where *x* denotes the SUB_PORT

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

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

```
   W. Server Management (WWW,PC,Z39)
       --------------------------------
        0. Exit procedure
        1. Monitor servers
        2. Stop servers
        3. Start servers
        4. View log file
Please select [exit]:
```

5.2 Server Logs

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

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

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

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

```
      Oct 20 12:10 www_server_4881.log.2010.1210
```

Apache server logs are written to `./dtle/apache/logs`.

5.3 Starting Your Own Server for Testing

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

For example:

Resource Discovery

The syntax for starting your own Web (Resource Discovery) server is:

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

For example:

```
      www_server 4881 80 1 stdout
```

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

To use your own Web server, enter the URL:

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

In our example, to use the DigiTool Resource Discovery enter:

```
      http://<URL>:4881
```

GUI

The syntax for starting your own GUI server is:

```
      pc_server <port number> stdout
```

For example:

```
      pc_server 6881 stdout
```

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

Specify **`6881`** as the address in your PC's `.\common\Tab\library.ini` file.

5.4 PC Client / PC Server

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

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

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

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

Add the following to the catalog.ini file to allow viewing objects from the GUI , the apache related information (address + port number) should be written

```
[DTL]
HostName=ram11:8888
```

In the following section, we will see how the PC and the PC server interact.

5.5 Connecting

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

Assuming that the password is verified, you then connect to an Admin Unit. Select **File / Connect** from the main menu. The system displays the Admin units listed in the module's `per_lib.ini` file, for example, `.\Catalog\Tab\per_lib.ini`.

When you click on a particular Admin Unit, the client software goes to the `.\common\Tab\library.ini` file and tries to connect to the IP address specified for this Admin Unit.

Service Requests

> **Note**
> See the *DigiTool 3.0 User Guide*, available from the Ex Libris Documentation Center.

The functions you perform on the PC generate service requests for the PC server. The `pc_com/pc_server` program processes all incoming requests. It checks the license date, user limit, and so on, and passes control to the program specified in the service request. For example, if the service request is **c0202** and the module is **CAT**, it passes control to the pc_cat/pc_cat_c0202 program. Just before doing this it writes an entry to the pc_server log:

```
SERVICE    : C0202
MODULE     : Catalog Services
DESCRIPTION: Get doc
ACTION     : GET
PROGRAM    : pc_cat_c0202
```

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

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

For example, if there was an error 0011 in pc_cat_c0202, the service_error_message would read the

./dtle/error_<lng>/pc_cat_c0202 file and find this entry:

0011 0000 L Unable to read key translation record

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

5.6 Analyzing PC Server Problems

There is normally just a single PC server (6881). Multiple logs for this server (for example, pc_ser_6881) in a time period when the server was not stopped intentionally (for example, for a backup) indicate a server failure. To get more information than what you see in the pc_ser_*nnnn* file, type:

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

For example

```
>>pc_server view 6881 25
```

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

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

For example

```
      pc_server check 6881 55459
```

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

```
      pc_server checkx <port number> [<line number>]
```

For example

```
      >>pc_server checkx 6881 55459
```

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

5.7 PC Server Configuration (pc_server_defaults)

5.7.1 Max Response Time

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

The only downside is that problematic transactions (transactions which are looping, and so on) may run even longer. So before changing the `PC_SERVER_MAX_RESPONSE_TIME` value, make certain that the problematic transactions are not due to a lack of the appropriate Oracle indexes: compare your `xxxnn` Admin Unit/Silo's `file_list` to the `dtlnn`'s `file_list`.

5.8 ue_01

This utility is initiated to execute as a daemon, constantly checking to see if indexing needs to be carried out.

5.9 Batch jobs

Batch jobs are executed either periodically (daily, weekly, and so on) or on request. They may be initiated via the **Services** menu in the GUI or the utilities **UTIL E/15 Managing Job Daemon** and **UTIL E/16 Update Daemon Job List**.

5.10 Services

Each GUI module has a **Services** menu. This option lets you submit batch jobs from a menu, choosing the values you want for each parameter. The submission screens which you see are in the `../dtle/pc_b_<lng>` directory (unless directed elsewhere in `path_convert`) and can be modified.

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

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

The job will appear in **UTIL C/1 Check Library Running Process** with its parameters. In this example the batch service is **Harvest Repository into Silo (p_harvest_01):**

```
start GEN01,A
procedure=p_harvest_01
Fixed param: GEN01,A,20050414,20050414,0800,1000,
setenv p_active_library "GEN01"
setenv p_flag "A"
setenv p_start_date_x "20050414"
setenv p_end_date_x "20050414"
setenv p_start_time_x "0800"
setenv p_end_time_x "1000"
```

The **`p_harvest_01`** component is the procedure that is being executed. The procedures are in the `dtl/proc` directory. `./dtl/proc/`p_harvest_01 executes a few actions, some directly to database and some by running programs.

Print templates are stored in `dtl01/form_<lng>`. All formats are taken from XSL files.

Any printed output produced by the batch job will be in the Admin Unit/Silo print directory (for example, `./dtl01/print`). The name of the output file is controlled by the **Output File** value on the submission screen. The log of the job will appear in the `./dtle/scratch` directory under the name of the process (in our example, `gen01_p_harvest_01.00523`).

5.11 Job List

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

5.12 cron Jobs

"cron jobs" are similar in principle to the DigiTool `job_list` described in the preceding section, except that their control and maintenance are outside of DigiTool, directly by the Unix Operating system.

5.13 www_server (Public)

The Web server for DigiTool Resource discovery is accessed via `http://<URL>/R`. For example: `http://ram11:8881/R`.

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

DIGITOOL VERSION 3.0

System Administrator's Guide - Oracle

Last Update: November 30, 2005
Document Version 1.0
Code: D-ver3-SAGOracle-1.0

Table of Contents

1. DigiTool / Oracle

1.1. Introduction

DigiTool is based on **Oracle 9*i* RDBMS**.

A typical DigiTool installation includes a number of DigiTool repositories, where each repository has its own set of data definitions (configuration tables) and data tables (Oracle tables). The different repositories represent either different types of information (metadata, administrative), or separate digital collections in the real world.

Typically, one real site will have at least four DigiTool repositories: one administrative repository and at least three metadata repositories.

1.2. Structure

Each DigiTool repository is implemented by:

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

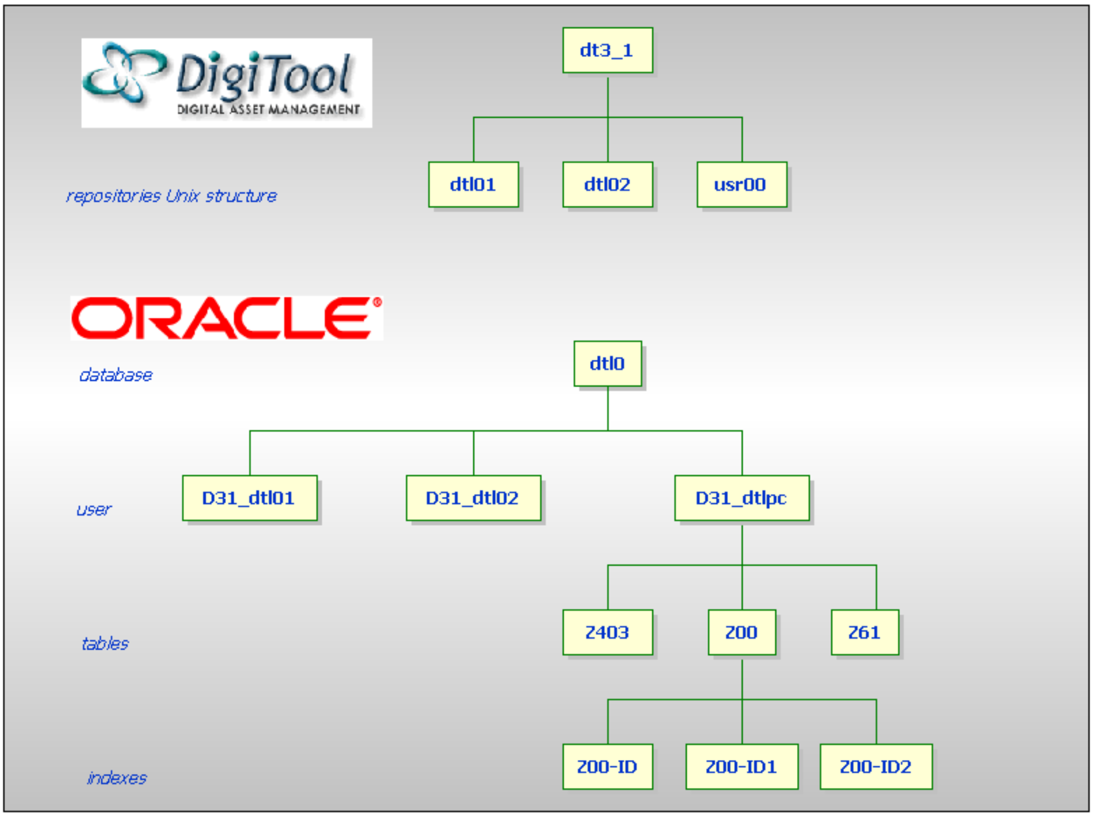


Figure 1

There are two types of repositories in DigiTool. Each repository is identified by a code that is made up of three characters followed by two digits. The digits identify the repository type (following the Ex Libris scheme of digits is a naming convention and not a system requirement).

- **Administrative repository** (DAT01) which contains information about users, profiles and permissions.
- **Metadata repository** where all the information regarding the physical collection is stored. This type of repository is used when searching for all information regarding the digital objects and their related metadata. Although only one metadata repository database is used, it can be constructed from several logical bases.

A new installation includes the following repositories:

- **DAT01 -** Administrative repository
- **DTL01** - Metadata repository for collections that were cataloged using MARC21
- **DTL02** - Metadata repository for collections that were cataloged using Dublin Core

1.2.1. SQL Access to the Oracle Tables

You can use SQL *Plus to access DigiTool's Oracle tables.

```
wallach.exlibris-int.il-d3(1) DTL31-01 >>s+ dtl31
```

```
      SQL*Plus: Release 9.2.0.6.0 - Production on Tue Mar 29 14:27:14
      2005

      Copyright (c) 1982, 2002, Oracle Corporation.  All rights
      reserved.

      Connected to:
      Oracle9i Enterprise Edition Release 9.2.0.6.0 - Production
      With the Partitioning option
      JServer Release 9.2.0.6.0 - Production

      SQL> desc z00
       Name                                      Null?    Type
       ----------------------------------------- -------- -----------
      -----------------
       Z00_DOC_NUMBER                            NOT NULL CHAR(9)
       Z00_NO_LINES                                       NUMBER(4)
       Z00_DATA_LEN                                       NUMBER(6)
       Z00_DATA                                           LONG

SQL>
```

1.2.2. Oracle Users in DigiTool

Each DigiTool collection is implemented as an Oracle user. The name of the Oracle user is composed of a prefix and the name of the collection. The prefix must be unique on the server, and is defined in the `$dtle_root/dtl_start` file during installation. The prefix is usually in the format `Dnm_` where `n` is the DigiTool version and `m` is the revision number. In addition, there are several Oracle users used by the DigiTool application, which are not related to a specific collection.

- **DTL**

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

- **DTL_ADMIN**

  This is an administrative user. This is a more privileged user, who, in addition to the privileges of the DTL user, can create, drop and alter Oracle tables, indexes, users, triggers, etc. The DTL_ADMIN Oracle user is used for these purposes in all DigiTool procedures.

- **DTL_DBA**

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

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

1.2.3. Passwords

DigiTool contains an encrypted file with the passwords of the Oracle users used by DigiTool (for example DTL, DTL_ADMIN, DTL_DBA and DTL31). 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 DigiTool password file.

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

```
Enter User Name:DTL
Enter New Password:<new_password>
Do you want to update this password ([n]/y)? y
If you want to update this password in Oracle
Enter DTL_DBA user/passwd,
or press [Enter] to exit:DTL_DBA/<DTL_DBA password>
Change passwd in Oracle

SQL*Plus: Release 9.2.0.6.0 - Production on Tue Mar 29 14:27:14 2005

Copyright (c) 1982, 2002, Oracle Corporation.  All rights reserved.

SQL> Connected.
SQL> SQL>
User altered.

SQL> Disconnected from Oracle9i Enterprise Edition Release 9.2.0.6.0
- Production
With the Partitioning option
JServer Release 9.2.0.6.0 - Production
```

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	Usage	Physical File Name
System	Oracle system tables	../exlibris/oradata/dtl3/system01.dbf
Temp	Temporary space (for sorting index creation and so on)	../exlibris2/oradata/dtl3/temp01.dbf
ts0	DigiTool tables	../exlibris2/oradata/dtl3/ts0_0.dbf
ts1	DigiTool indexes	../exlibris/oradata/dtl3/ts1_0.dbf

Each Oracle table and index has to be mapped to a tablespace. In DigiTool, this mapping is done via a configuration file named file_list. The file_list file is located under the root of each database.

1.3.2. Users

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

- Default tablespace - which specifies where Oracle 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 page 7.

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 page 7.

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 (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

Tablespaces in Oracle can have one of two types of extent management:

- Dictionary-managed tablespaces
- Locally-managed tablespaces

Locally-managed tablespaces

All tablespaces in DigiTool are created as locally-managed tablespaces . 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
```

```
----------- --------- -------- --------- --------- ---- ----
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
```

There are two types of extent allocation when using Locally-managed Tablespaces. In the **ALLOC_TYP** column, you may see these values:

SYSTEM = auto allocate

UNIFORM = uniform

- **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, one extent or any other combination.

- **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, five extents will be used.

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 repositories use Auto Allocate and customer repositories use Uniform.

The **file_list** is a table located under each repository root directory. It consists of parameters used to manage all Oracle objects (**table**, **index**, **synonym**, and so on) of the particular repository.

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 repository name (to link to)
- **NA**table name (not applicable in current repository)
- **SEQ** sequence name
- **RS** table name repository name alias 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 49 for more details.

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

```
TAB z52                 10K             10K             ts0
IND z52_id              10K             10K             ts1
TAB z401                50M             50M             tslob
IND z401_id             2M              2M              ts1
```

In DigiTool, you can 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         2M          1M    ts0         <other host>.dtl3
IND   z02_id      1M          1M    ts1         <other host>.dtl3
IND   z02_id1     1M          1M    ts1         <other host>.dtl3
RS    z52         dtl31       <other host>.dtl3
```

Notes

If the BUDGET-PER-ORD-UNIT variable is set to Y and the Search Modes are Group or Wildcard, when one budget in the group is denied, all budgets included in the group are not displayed.

In the Z02 table, the table and indexes are physically located on a different node (`<other host>.dtl3`), 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 (`<other host>.dtl3`), 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 `<other host>.dtl3`) must be defined in the file:

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

For example:

```
<other host>.dtl3=(description=
(address=
(protocol=tcp)
(host=<other host>)
(port=1521))
(connect_data=(sid=dtl3)))
```

For more information see also *Working With Remote Oracle Tables* on page 49 and *UTIL O - Oracle Management* on page 20

2. Utilities

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

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

```
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. GUI MEDITOR 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
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 DTL31
----------------------------------------------------
     0. Exit Procedure
     8. List Analyzed Tables / Indexes
     9. Delete Statistics for Analyzed Tables / Indexes
    10. Drop Result Set Tables (z05,z110)
    11. Drop Session Tables (z62/z64/z65)
    12. Drop Lock Tables (z60/z50)
    13. Drop Statistics Table (z34)
    14. Drop Update Doc Table (z07)
    15. Drop Web Basket Table (z109)
    16. Drop Event Table (z35)
    17. Manage Oracle Tables
    19. Export Repository
    20. Import Repository
    21. List Oracle Objects
```

UTIL A/8 List Analyzed Tables / Indexes

UTIL A/9 Delete Statistics for Analyzed Tables / Indexes

DigiTool was written and is tuned to work with the rule-based optimizer. The queries issued on DigiTool tables are very simple. It is usually a select by index key prefix. There is no benefit in having cost-based-optimization of DigiTool queries because the rule-based query plan is always optimal. The Cost Based Optimizer can only slow them down.

Performing analyze or dbms_stat on DigiTool tables causes Oracle to choose the Cost Based Optimizer and can cause a malfunctioning of DigiTool functions.

UTIL A/8 and UTIL A/9 were created in order to prevent the problem with analyzed tables/indexes.

UTIL A/8 checks in the database if any tables or indexes have been analyzed and prepares a list of analyzed tables/indexes. If the list created is not empty, activate UTIL A/9 to delete these statistics.

UTIL A/10 Drop Result Set Tables (Z05, Z110)

The Z05 and Z110 tables contain the search result sets of FIND requests. These tables should be cleaned out periodically (for example weekly) with the Drop Result Set Tables (Z05, Z110) utility.

When you select this utility, the system will notify you that you will be deleting tables Z05 and Z110 in repository VIR01. For example:

```
delete Z05,Z110 in VIR01
```

Notes

This utility may be run from any repository, even though Z05 and Z110 are always defined in the VIR01 repository.

Instead of running UTIL A/10 you may want to routinely execute the clear_vir01 procedure which will clear out all of the tables in VIR01:

```
digitool-d3(1)> dlib vir01
digitool-d3(1) VIR01-DTL> $DTL_proc/clear_vir01 vir01
```

UTIL A/11 Drop Session Tables (Z62/Z64/Z65)

Each time a user enters DigiTool via the Web OPAC a session is started. The session definition is written in tables Z62, Z64 and Z65. These tables should be cleaned out periodically with the Drop Session Tables (Z62/Z64/Z65) utility. The utility drops the tables and then creates them empty.

When you select this utility, the system notifies you that you will be deleting the Z62, Z64 and Z65 tables in the DAT01 repository. For example:

```
delete Z62/Z64/Z65 in DAT01
```

Note

This utility may be run from **any** repository, even though Z62, Z64 and Z65 are only defined in the DAT01 repository.

UTIL A/12 Drop Lock Tables (Z60,Z50)

When a record is locked, a line is written in table Z60. This table should be cleaned out periodically with the **Drop Lock Table (Z60,Z50)** utility. The utility drops the tables and then creates them empty.
When you select this utility, the system will notify you that you will be deleting tables Z60 from VIR01 and Z50 from the defined repository. For example:

```
delete Z60/Z50 in VIR01/DTL31
```

Note

This utility may be run from any repository, even though Z60 will be dropped from the VIR01 repository.

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 `$dtle_root/www_server.conf` file and the `$dtle_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 DAT01
```

Note

This utility may be run from **any** repository, even though Z34 is only defined in **one** of the repositories through the environment variable **pw_library** in the `$dtle_root/dtl_start` file.

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

When a cataloging (BIB) 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 repository. For example:

```
delete Z07 in DTL31
```

UTIL A/15 Drop Web "Basket" Table (Z109)

Each time a user puts records in a Web "basket", information is stored in the Z109 table. This table can be cleaned out periodically with the **Drop Web "Basket" Table (Z109)** utility.

When you select this utility, the system will notify you that you will be deleting table Z109 in the VIR01 repository. For example:

```
delete Z109 in VIR01
```

Notes

This utility may be run from **any** repository, even though Z109 is only defined in the repository VIR01.
Instead of running **UTIL A/15 Drop Web "Basket Table (Z109),** you may want to execute the **clear_vir01** procedure from time-to-time in order to clear out all of the tables in VIR01. See *UTIL A/10 Drop Result Set Tables (Z05, Z110)* on page 12 for commands.

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 repository. For example:

```
delete z35 in DTL31
```

UTIL A/17 Manage Oracle Tables

```
A.17. Manage Oracle Tables of DTL31
       0.  Exit Procedure
       1.  Drop & Create Table and Index
       2.  Create Index
       3.  Rebuild Index
       4.  Drop Index
       5.  Synonyms
       6.  Triggers
       9.  Sequences
      10.  Edit file_list of DTL31 Tables
      11.  Space Utilization
      14.  List Existing Indexes for a Table
      15.  Analyze Table/Index
      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 repository. The following prompts appear:.

- **all** - Enter `all` to create/recreate all of the tables in the repository 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 repository. 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 repository. 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 (for example, 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 DTL31

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

UTIL A/17/5/1 List Repository Synonyms

Lists the synonyms defined in the current repository.

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

Drops all existing synonyms in the repository and then creates all the repository’s synonyms as defined in the repository’s **file_list**.

UTIL A/17/6 Triggers

```
     A.17.6 Manage Triggers in DTL31

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

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

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

UTIL A/17/6/2 Enable Trigger

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

UTIL A/17/6/3 Disable Trigger

Disables a repository 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 repository, the repository name, the table which the trigger is defined on, and the current status of the trigger (enabled or disabled).

UTIL A/17/9 Sequences

```
     A.17.9 Manage Sequences in DTL31

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

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

Creates or recreates repository sequences. You will be prompted.

- **all** - Enter `all` to Drop and create all the repository'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 DTL31 Tables

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

UTIL A/17/11 Space Utilization

```
     A.17.11 Space Utilization in DTL31

       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 DTL31 files *****

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

     ***** D31_DTL31 Tables statistics *****

                 SEGMENT     TABLESPACE
SEGMENT NAME     TYPE        NAME        BYTES (K)     BLOCKS  EXTENTS
---------------  ----------  -------  ----------  ----------  -------

SYS_IL000007543   LOBINDEX    TSLOB        8192        1024     1
SYS_LOB00000754   LOBSEGMENT  TSLOB        8192        1024     1
Z00               TABLE       TS0          2048         256     2
Z00H              TABLE       TS0           128          16     2
Z00H_ID           INDEX       TS1           128          16     2
Z00_ID            INDEX       TS1           128          16     2
Z07               TABLE       TS0           128          16     2
Z07_ID            INDEX       TS1           128          16     2
Z07_ID1           INDEX       TS1           128          16     2
Z101              TABLE       TS0           448          56     7
Z101_ID           INDEX       TS1           128          16     2
Z104              TABLE       TS0           128          16     2
Z104_ID           INDEX       TS1           128          16     2
```

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

```
Enter Table/Index Name: z01
TABLE_NAME = Z01
check = DTL31
active_library = DTL31

BYTES/1024      BLOCKS     EXTENTS  INITIAL_EXTENT  NEXT_EXTENT
----------  ----------  ----------  --------------  -----------
      2048         256           2         2097152
Elapsed: 00:00:00.30

  COUNT(*)
----------
       696
Elapsed: 00:00:00.01
```

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
```

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

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

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 DTL31 yes

Index analyzed.


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

1 row selected.


 Enter to continue
```

UTIL A/17/17 Manage Content Indexes

For future use.

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.

2.2. UTIL O - Oracle Management

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

General:

DigiTool is based on **Oracle9*i* RDBMS**.

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

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

Note

DigiTool enables you to place the Oracle database on a separate server from the DigiTool application, and even to distribute Oracle tables across two or more databases on different servers. See *Working With Remote Oracle Tables* on page 49.

Placing the Oracle database on a separate server is done by setting the `TWO_TASK` environment variable (in the `dtl_start` file in the "DigiTool Server") to the alias pointing to the "Oracle server", as defined in the `tnsname.ora` file in the "DigiTool Server". See *Working in a TWO_TASK Environment* on page 53.

2.2.1. About the Oracle Listener

When DigiTool is installed on the same server as the database, DigiTool 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 DigiTool 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 DigiTool to work properly. See more about this option in *Working With Remote Oracle Tables* on page 49.

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
```

UTIL O/1/1 Activate Oracle Server

Note

Requires DigiTool DBA username and password

In order for DigiTool to interact with Oracle, the Oracle server must be running. The Oracle Listener must be running in certain cases, as explained in *About the Oracle Listener* on page 21. They may be started automatically at boot time (this is determined during installation) and also controlled by the DigiTool Oracle Management utilities under **UTIL O**.

The user **DTL_DBA** is 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 DTL DBA username/password.
Username/password: DTL_DBA/<DTL_DBA password>
```

Enter the DTL_DBA username and password.

UTIL O/1/2 Close Oracle server

Note

Requires DigiTool DBA username and password

The user **DTL_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 DTL DBA username/password.
      username/password:DTL_DBA/<DTL_DBA password>
```

Enter the DigiTool 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:

```
  817 ?        S      0:07 ora_pmon_dtl3
  819 ?        S      0:03 ora_dbw0_dtl3
  821 ?        S      0:17 ora_lgwr_dtl3
  823 ?        S      0:06 ora_ckpt_dtl3
  825 ?        S      0:02 ora_smon_dtl3
  827 ?        S      0:00 ora_reco_dtl3
  829 ?        S      0:00 ora_cjq0_dtl3
  831 ?        S      8:51 ora_s000_dtl3
  833 ?        S      0:46 ora_s001_dtl3
  835 ?        S      0:02 ora_s002_dtl3
  837 ?        S      0:00 ora_s003_dtl3
  839 ?        S      1:10 ora_d000_dtl3
  841 ?        S      0:46 ora_d001_dtl3
  843 ?        S      1:15 ora_d002_dtl3
  845 ?        S      1:10 ora_d003_dtl3
  847 ?        S      0:00 ora_d004_dtl3
  849 ?        S      0:00 ora_d005_dtl3
  851 ?        S      0:00 ora_d006_dtl3
  853 ?        S      0:00 ora_d007_dtl3
  855 ?        S      0:00 ora_arc0_dtl3
  857 ?        S      0:00 ora_arc1_dtl3
```

Note

This utility is only relevant if you are running the Oracle server on the same node as the DigiTool 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

```
Connected.

INSTANCE_N HOST_NAME       VERSION      STARTUP_TI STATUS     LOGINS
---------- --------------- ------------ ---------- ---------- -------
---
dtl3      wallach.exlibri 9.2.0.6.0    28-MAR-05  OPEN       ALLOWED
           s-int.il


BANNER
----------------------------------------------------------------
Oracle9i Enterprise Edition Release 9.2.0.6.0 - Production
PL/SQL Release 9.2.0.6.0 - Production
CORE    9.2.0.6.0       Production
TNS for Linux: Version 9.2.0.6.0 - Production
NLSRTL Version 9.2.0.6.0 - 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 DigiTool 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, DigiTool 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 DigiTool Oracle Management utilities.

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:

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

To close Oracle Listener 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:

```
  889    ?                               S                        0:00
/exlibris/app/oracle/product/920/bin/tnslsnr LISTENER
```

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 9.2.0.6.0 - Production on 29-MAR-2005
16:05:46

Copyright (c) 1991, 2002, Oracle Corporation.  All rights reserved.

Connecting to (DESCRIPTION=(address=(protocol=ipc)(key=dtl3)))
```

```
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 9.2.0.6.0 -
Production
Start Date                28-MAR-2005 08:51:53
Uptime                    1 days 7 hr. 13 min. 53 sec
Trace Level               off
Security                  OFF
SNMP                      OFF
Listener Parameter File
/exlibris/app/oracle/product/920/network/admin/listene
r.ora
Listener Log File
/exlibris/app/oracle/product/920/network/log/listener.
log
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=dtl3)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=dtl31)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=ram7)(PORT=1521)))
Services Summary...
Service "dtl3" has 1 instance(s).
  Instance "dtl3", status UNKNOWN, has 1 handler(s) for this
service...
Service "dtl3.exlibris-int.il" has 1 instance(s).
  Instance "dtl3", status READY, has 9 handler(s) for this service...
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 9.2.0.6.0 - Production on 29-MAR-2005
16:08:22

Copyright (c) 1991, 2002, Oracle Corporation.  All rights reserved.

Connecting to (DESCRIPTION=(address=(protocol=ipc)(key=dtl3)))
Services Summary...
Service "dtl3" has 1 instance(s).
  Instance "dtl3", status UNKNOWN, has 1 handler(s) for this
service...
    Handler(s):
      "DEDICATED" established:6 refused:0
         LOCAL SERVER
Service "dtl3.exlibris-int.il" has 1 instance(s).
  Instance "dtl3", status READY, has 9 handler(s) for this service...
    Handler(s):
      "D007" established:0 refused:0 current:0 max:1002 state:ready
         DISPATCHER <machine: wallach.exlibris-int.il, pid: 853>
         (ADDRESS=(PROTOCOL=ipc)(KEY=#853.2))
      "D006" established:0 refused:0 current:0 max:1002 state:ready
         DISPATCHER <machine: wallach.exlibris-int.il, pid: 851>
         (ADDRESS=(PROTOCOL=ipc)(KEY=#851.1))
      "D005" established:0 refused:0 current:0 max:1002 state:ready
         DISPATCHER <machine: wallach.exlibris-int.il, pid: 849>
         (ADDRESS=(PROTOCOL=ipc)(KEY=#849.1))
      "D004" established:0 refused:0 current:0 max:1002 state:ready
         DISPATCHER <machine: wallach.exlibris-int.il, pid: 847>
         (ADDRESS=(PROTOCOL=ipc)(KEY=#847.1))
      "D003" established:138 refused:0 current:9 max:1002 state:ready
         DISPATCHER <machine: wallach.exlibris-int.il, pid: 845>
         (ADDRESS=(PROTOCOL=tcp)(HOST=ram7)(PORT=32776))
      "D002" established:149 refused:0 current:7 max:1002 state:ready
         DISPATCHER <machine: wallach.exlibris-int.il, pid: 843>
         (ADDRESS=(PROTOCOL=tcp)(HOST=ram7)(PORT=32775))
      "D001" established:128 refused:0 current:9 max:1002 state:ready
         DISPATCHER <machine: wallach.exlibris-int.il, pid: 841>
         (ADDRESS=(PROTOCOL=tcp)(HOST=ram7)(PORT=32774))
      "D000" established:197 refused:0 current:7 max:1002 state:ready
         DISPATCHER <machine: wallach.exlibris-int.il, pid: 839>
         (ADDRESS=(PROTOCOL=tcp)(HOST=ram7)(PORT=32769))
      "DEDICATED" established:0 refused:0 state:ready
         LOCAL SERVER
The command completed successfully
```

UTIL O/3 Oracle Logs

```
     O.3 Oracle Logs
        ---------------
         0.  Exit Procedure
         1.  View Oracle ALERT LOG
         2.
```

UTIL O/3/1 View Oracle ALERT LOG

You will be prompted for the number of lines to display from the Oracle ALERT LOG. The displayed lines are the most recent.

UTIL O/6 NLS

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

UTIL O/6/1 Show NLS Parameters

DigiTool uses the UTF8 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

DigiTool 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. DigiTool processes (servers and batch procedures) are stopped (using the **dtl_shutdown** script in `$dtle_root`).
2. Oracle database is shut down.
3. Oracle database is started up.
4. DigiTool is restarted (using the **dtl_startup** script in `$dtle_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 recovery.

UTIL O/7/1 Turning Archiving On

Note

Requires DigiTool 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. DigiTool processes (servers and batch procedures) are stopped (using the **DTL_shutdown** script in `$dtle_root`).
2. Oracle database is shut down.
3. Oracle database is started up.
4. DigiTool is restarted (using the **dtl_startup** script in `$dtle_root`).

UTIL O/7/2 Turning Archiving Off

Note

Requires DigiTool 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. DigiTool processes (servers and batch procedures) are stopped (using the **DTL_shutdown** script in `$dtle_root`).
2. Oracle database is shut down.
3. Oracle database is started up.
4. DigiTool is restarted (using the **dtl_startup** script in `$dtle_root`).

UTIL O/7/3 Show Archiving Status

Note

Requires DigiTool 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 9.2.0.6.0 - Production on Tue Mar 29 16:25:54 2005

Copyright (c) 1982, 2002, Oracle Corporation.  All rights reserved.

SQL> Connected.
SQL> SQL> Database log mode            No Archive Mode
Automatic archival             Enabled
Archive destination            /exlibris/oradata/dtl3/arch/
Oldest online log sequence     608
Current log sequence           612
SQL> Disconnected from Oracle9i Enterprise Edition Release 9.2.0.6.0
- Production
With the Partitioning option
JServer Release 9.2.0.6.0 - Production
```

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
```

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 DigiTool collection users and others are administrative users.

```
The Database exlibris Contains the Following Users:
======================================================
SYS
SYSTEM
OUTLN
DBSNMP
TRACESVR
DTL_ADMIN
DTL
DTL_BACKUP
DTL_DBA
MDSYS
CTXSYS
PERFSTAT
D31_DTL32
D31_DAT01
D31_DTL31
```

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, you can use **UTIL Y/8 Update Oracle Passwords of DigiTool Users (ora_passwd)** to change the user's password.

```
Enter User Name to Create New User: <new user name>
enter yes to create oracle user dtl99 <yes/no>
default password is D31_dtl99
if user dtl99 exists all data will be erased!!!
enter no to reconfirm <yes/no>
source create_ora_user_b D31_dtl99
create_ora_user_b D31_dtl99

SQL*Plus: Release 9.2.0.6.0 - Production on Tue Mar 29 16:28:35 2005

Copyright (c) 1982, 2002, Oracle Corporation.  All rights reserved.
```

```
Enter user-name:
Connected to:
Oracle9i Enterprise Edition Release 9.2.0.6.0 - Production
With the Partitioning option
JServer Release 9.2.0.6.0 - Production

SQL> EXIT
Disconnected from Oracle9i Enterprise Edition Release 9.2.0.6.0 -
Production
With the Partitioning option
JServer Release 9.2.0.6.0 - Production

SQL*Plus: Release 9.2.0.6.0 - Production on Tue Mar 29 16:28:36 2005

Copyright (c) 1982, 2002, Oracle Corporation.  All rights reserved.

Enter user-name:
Connected to:
Oracle9i Enterprise Edition Release 9.2.0.6.0 - Production
With the Partitioning option
JServer Release 9.2.0.6.0 - Production

SQL> old   1: DROP USER &&1 CASCADE
new   1: DROP USER D31_dtl99 CASCADE
DROP USER D31_dtl99 CASCADE
          *
ERROR at line 1:
ORA-01918: user 'D31_DTL99' does not exist


old   1: CREATE USER &&1 IDENTIFIED BY &&1
new   1: CREATE USER D31_dtl99 IDENTIFIED BY D31_dtl99

User created.

old   1: GRANT DTL_LIBRARY TO &&1
new   1: GRANT DTL_LIBRARY TO D31_dtl99

Grant succeeded.

Disconnected from Oracle9i Enterprise Edition Release 9.2.0.6.0 -
Production
With the Partitioning option
JServer Release 9.2.0.6.0 - Production
```

UTIL O/10 SQL*Plus Session

In **Oracle 9*i***, **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 **DTL_DBA**.

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

```
***** DO: connect dtl_dba/dtl_dba_passwd as sysdba

SQL*Plus: Release 9.2.0.6.0 - Production on Tue Mar 29 16:30:38 2005
```

```
Copyright (c) 1982, 2002, Oracle Corporation.  All rights reserved.

SQL>
SQL> connect DTL_DBA as sysdba
Enter password: <DTL_DBA password>
Connected.
SQL>
```

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/dtl39/dtl39_ts1_01.dbf
Enter database block size [8192]:

To continue you will need to enter oracle9's password.
Password:
Execute oracle cshrc

DBVERIFY: Release 9.2.0.6.0 - Production on Thu Mar 31 16:14:49 2005

Copyright (c) 1982, 2002, Oracle Corporation.  All rights reserved.

DBVERIFY - Verification starting : FILE =
/exlibris/oradata/dtl3/dtl3_ts1_01.dbf
.....................................................................
...........
...................................................

DBVERIFY - Verification complete

Total Pages Examined         : 131072
Total Pages Processed (Data) : 0
Total Pages Failing   (Data) : 0
Total Pages Processed (Index): 32756
Total Pages Failing   (Index): 0
Total Pages Processed (Other): 8383
Total Pages Processed (Seg)  : 0
Total Pages Failing   (Seg)  : 0
Total Pages Empty            : 89933
Total Pages Marked Corrupt   : 0
Total Pages Influx           : 0
Highest block SCN            : 48473210 (0.48473210)
```

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
```

UTIL O/13/1 List of Database Files

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

```
The Database dtl3 contains the following files:
======================================================

T        NAME                                         SIZE K  F
-------  -------------------------------------------  -------  ---
DRSYS    /exlibris3/oradata/DTL5/DTL5_drsys_01.dbf    86016  7
DRSYS    /exlibris/oradata/dtl3/dtl3_drsys_01.dbf        86016  7
RBS      /exlibris/oradata/dtl3/dtl3_rbs_01.dbf         245760  3
SYSTEM   /exlibris/oradata/dtl3/dtl3_system_01.dbf      266240  1
TOOLS    /exlibris/oradata/dtl3/dtl3_tools_01.dbf       524288  2
TS0      /exlibris/oradata/dtl3/dtl3_ts0_01.dbf        2048000  4
TS1      /exlibris/oradata/dtl3/dtl3_ts1_01.dbf        2048000  5
TSLOB    /exlibris/oradata/dtl3/dtl3_tslob_01.dbf      2048000  6

T        NAME                                         SIZE K  F
-------  -------------------------------------------  -------  ---
TEMP     /exlibris/oradata/dtl3/dtl3_temp_01.dbf       1024000  1
```

UTIL O/13/2 Resize Oracle Datafile

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

UTIL O/13/3 Add File to Tablespace

Note

Requires DigiTool DBA username and password

Tablespaces are composed of one or more data files. When a tablespace does not have enough free space it needs to be enlarged. This may be done by 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 DigiTool 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 existent files will appear. For example:

Tablespace TS1 consists of the following files:

```
      /exlibris1/oradata/dtl3/dtl3_ts1_0.dbf
      /exlibris1/oradata/dtl3/dtl3_ts1_1.dbf
      /exlibris/oradata/dtl3/dtl3_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 DTL_DBA username/password.
username/password:DTL_DBA/<DTL_DBA password>

Enter Tablespace name: ts0

Tablespace TS0 consist of the following files:

/exlibris3/oradata/dtl3/dtl3_ts0_01.dbf
/exlibris3/oradata/dtl3/dtl3_ts0_02.dbf

Enter new file name: /exlibris3/oradata/dtl3/dtl3_ts0_03.dbf
Enter file size (MB): 2000

Tablespace:  TS0
New file:    /exlibris3/oradata/dtl3/dtl3_ts0_03.dbf
Size:        2000MB

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

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

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/DTL5/DTL5_ts
TS0     4    54665     1610688  /exlibris/oradata/dtl3/dtl3_ts0_01.
                                dbf
TS1     5    61449     1556416  /exlibris/oradata/dtl3/dtl3_ts1_01.
                                dbf
TSLOB   6    63753     1537984  /exlibris/oradata/dtl3/dtl3_tslob_0
                                1.dbf
TOOLS   2    5945       476736  /exlibris/oradata/dtl3/dtl3_tools_0
                                1.dbf

TABLES  F    BLOCK_ID  KBYTES   NAME
------  ---  --------  ------   -----------------------------------

RBS     3    5202       204152  /exlibris/oradata/dtl3/dtl3_rbs_01.
                                dbf
SYSTEM  1    9291       191920  /exlibris/oradata/dtl3/dtl3_system_
                                01.dbf
DRSYS   7    1929        70592  /exlibris/oradata/dtl3/dtl3_drsys_0
                                1.dbf
TS1     5    57737       28672  /exlibris/oradata/dtl3/dtl3_ts1_01.
                                dbf

TABLES  F    BLOCK_ID  KBYTES   NAME
------  ---  --------  ------   -----------------------------------
TS0     4    51465       24576  /exlibris/oradata/dtl3/dtl3_ts0_01.
                                dbf
TS0     4    49033       18432  /exlibris/oradata/dtl3/dtl3_ts0_01.
                                dbf
TS0     4    35593       17408  /exlibris/oradata/dtl3/dtl3_ts0_01.
                                dbf
```

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 Repository/Repositories in Each Tablespace
     5. Space Used by a Group of Repositories in Each Tablespace
     6. Coalesce Contiguous Free Extents
     8. DTL 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
        50                           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 Repository/Repositories in Each Tablespace

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

```
Enter Repository name (full or truncated, for example, dtl): dtl

OWNER                           TABLESPACE_NAME       SIZE_MB
---------------------------     ------------------    ----------
D31_DTL31                       TS0                   17.7597557
D31_DTL31                       TS1                   32.5178626
D31_DTL31                       TSLOB                 54.4674198
D31_DTL32                       TS0                   17.5721527
D31_DTL32                       TS1                   32.3302595
D31_DTL32                       TSLOB                 54.2798168
D31_DTL33                       TS0                   18.0098931
D31_DTL33                       TS1                   32.4553282
D31_DTL33                       TSLOB                 54.2798168
D31_DTL34                       TS0                   17.5096183
D31_DTL34                       TS1                    30.266626

OWNER                           TABLESPACE_NAME       SIZE_MB
---------------------------     ------------------    ----------
D31_DTL34                       TSLOB                 4.12726718
```

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

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

```
Enter first 3 characters of Repository code (for example, dtl): dtl

TABLESPACE_NAME                       SIZE_MB
------------------------------  ----------
TS0                             88.3610382
TS1                             159.837802
TSLOB                           221.434137
```

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 DigiTool 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 DigiTool DBA username and password.

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

Note

The procedure only coalesces extents for the TS0 and TS1 tablespaces.

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

```
NAME                            TOTAL SIZE M
------------------------------  ------------
TS0                                    15000
TS1                                    10000
TSLOB                                   2000

NAME                            TOTAL FREE M
------------------------------  ------------
TS0                                     1744
TS1                                     1745
TSLOB                                     20
```

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 DigiTool application's server.

For additional information about managing database links, see *Working With Remote Oracle Tables* on page 49.

Note

The use of database links is different from the use of TWO_TASK, where the entire 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 existent 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 [DTL]
Enter password to remote system [DTL passwd]
```

The database link will be created and the following messages will appear (in our example the TNS service name is `ram40.dtl3` and the username is `DTL`):

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

drop database link ram40.dtl3
                   *
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 DigiTool, database links are used when working with tables on a remote database. See *Working With Remote Oracle Tables* on page 49.

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.  Database Tablespaces
       ---------------------------
       0. Exit procedure
       1. Create a Tablespace
       2. List Tablespace Files
       4. Show Tablespaces Definition
       5. Show Tablespace Allocated/Free/Used Space
```

UTIL O/17/1 Create a Tablespace

One rarely needs to create a tablespace, since all needed tablespaces should have been created during system installation. This utility is used if there is a need for an additional tablespace. Please read about tablespaces and their types in section *1.4.1 Introduction to locally-managed tablespaces*.

You will be prompted:

```
To Create a new Tablespace, Enter DTL_DBA username/password.
username/password:DTL_DBA/<DTL_DBA password>
Enter Tablespace name: test
Enter new file name (full path) : /exlibris/oradata/dtl3/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/dtl3/test_01.dbf
File size:    1000MB
Allocation : UNIFORM SIZE 4m
confirm (y/[n]):
```

UTIL O/17/2 List Tablespace Files

```
Enter Tablespace name: ts1

Tablespace TS1 consist of the following files:

NAME                                      SIZE K   F
--------------------------------------    -------  ---
/exlibris/oradata/dtl3/dtl3_ts1_01.dbf    2048000  6
/exlibris/oradata/dtl3/dtl3_ts1_02.dbf    2048000  11
/exlibris/oradata/dtl3/dtl3_ts1_03.dbf    2048000  12
/exlibris/oradata/dtl3/dtl3_ts1_04.dbf    2048000  15
/exlibris/oradata/dtl3/dtl3_ts1_05.dbf    2048000  17
```

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 undoing 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

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/dtl3/dtl3_system_01.dbf
R01              Priv  RBS
/exlibris/oradata/dtl3/dtl3_rbs_01.dbf
R02              Priv  RBS
Segm Name        Ownr  In TabSpace
File containing header of rbs
/exlibris/oradata/dtl3/dtl3_rbs_01.dbf
R03              Priv  RBS
/exlibris/oradata/dtl3/dtl3_rbs_01.dbf
R04              Priv  RBS
/exlibris/oradata/dtl3/dtl3_rbs_01.dbf

Segm Name        Ownr  In TabSpace
File containing header of rbs
R05              Priv  RBS
/exlibris/oradata/dtl3/dtl3_rbs_01.dbf
R06              Priv  RBS
/exlibris/oradata/dtl3/dtl3_rbs_01.dbf
Segm Name        Ownr  In TabSpace
File containing header of rbs
R07              Priv  RBS
/exlibris/oradata/dtl3/dtl3_rbs_01.dbf
R08              Priv  RBS
/exlibris/oradata/dtl3/dtl3_rbs_01.dbf
Segm Name        Ownr  In TabSpace
File containing header of rbs
R09              Priv  RBS
/exlibris/oradata/dtl3/dtl3_rbs_01.dbf
R10              Priv  RBS
/exlibris/oradata/dtl3/dtl3_rbs_01.dbf
Online Rollback Segments:

Name NrEx  Size      Init      Next PctI  MinE  MaxE  Opt size  Stat

SYSTEM  7  552K    57,344    57,344    0     2   505            OnL
R01     8  4152K  524,288   524,288    0     8  #### 4,194,304  OnL
R02     8  4152K  524,288   524,288    0     8  #### 4,194,304  OnL
R03     8  4152K  524,288   524,288    0     8  #### 4,194,304  OnL
R04     8  4152K  524,288   524,288    0     8  #### 4,194,304  OnL
R05     8  4152K  524,288   524,288    0     8  #### 4,194,304  OnL
R06     8  4152K  524,288   524,288    0     8  #### 4,194,304  OnL
R07     8  4152K  524,288   524,288    0     8  #### 4,194,304  OnL
R08     8  4152K  524,288   524,288    0     8  #### 4,194,304  OnL
R09     8  4152K  524,288   524,288    0     8  #### 4,194,304  OnL
R10     8  4152K  524,288   524,288    0     8  #### 4,194,304  OnL
```

UTIL O/18/3 Rollback Segments Dynamic Allocation

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

```
NAME   EXT      RSSIZE      WRITES  SHRN   AVGSHR  WRAPS CUREXT WAITS
R01      8     4251648   263624626     0        0    585      2     1
R02      8     4251648   258801510     0        0    577      7     1
R03      8     4251648   252686664     0        0    564      3     1
R04      8     4251648   256508870     0        0    571      7     0
R05      8     4251648   263364698     0        0    587      2     2
R06      8     4251648   259428394     0        0    577      1     1
R07      8     4251648   270411704     0        0    598      1     0
R08      8     4251648   256912874     0        0    571      2     0
R09      8     4251648   264234626     0        0    586      3     2
R10      8     4251648   259544752     0        0    577      6     1
SYST     7      565248        6540     0        0      1      6     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                         73888
db_block_buffers              163840000
log_buffer                      2621440
free memory                   116266332
miscellaneous                    728736
db_block_hash_buckets            451144
ktlbk state objects              105716
KGFF heap                          7760
PL/SQL MPCODE                   1048432
PLS non-lib hp                     2096
table definiti                     4320

NAME                              BYTES
--------------------------  ----------
type object de                    55232
VIRTUAL CIRCUITS                 367212
trigger inform                      760
sessions                         485100
pl/sql source                      5292
State objects                    244144
trigger defini                      196
branches                          59520
log_buffer                        81920
long op statistics array          99000
trigger source                      320
```

```
NAME                            BYTES
-------------------------- ----------
fixed allocation callback         640
KQLS heap                     1406704
PL/SQL DIANA                   755476
dictionary cache               874128
transactions                   220324
character set object            53808
table columns                   28232
message pool freequeue         124552
library cache                 3794472
db_block_buffers              2720000
sql area                      5321128
```

UTIL O/19/2 Flush Shared Pool

You will be prompted for the DTL_DBA user/password.

UTIL O/20 Multi Threaded Server

```
O.20. Multi Threaded Server
      ----------------------
       0.  Exit Procedure
       1.  Show MTS Parameters
       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 DigiTool, The MTS infrastructure exists in the database but is only implemented in special cases, website in coordination with Ex Libris.

UTIL O/20/1 Show MTS Parameters

```
DTL_DBA/DTL_DBA

SQL*Plus: Release 9.2.0.6.0 - Production on Thu Mar 31 16:26:05 2005

Copyright (c) 1982, 2002, Oracle Corporation.  All rights reserved.

SQL> Connected.
SQL> SQL>
```

```
NAME                                 TYPE        VALUE
------------------------------------ ----------- --------------------
----------
mts_circuits                         integer     225
mts_dispatchers                                                string
(PROTOCOL=TCP)(DISPATCHERS=4),

(PROTOCOL=IPC)(DISPATCHERS=4)
mts_listener_address                 string
mts_max_dispatchers                  integer     30
mts_max_servers                      integer     4
mts_multiple_listeners               boolean     FALSE
mts_servers                          integer     4
mts_service                          string      dtl39
mts_sessions                         integer     220
SQL> Disconnected from Oracle9i Enterprise Edition Release 9.2.0.6.0
- Productio
n
With the Partitioning option
JServer Release 9.2.0.6.0 – Production
```

UTIL O/20/2 Show Listener Services

```
Service "dtl3" has 1 instance(s).
  Instance "dtl3", status UNKNOWN, has 1 handler(s) for this
service...
    Handler(s):
Service "dtl3.exlibris-int.il" has 1 instance(s).
  Instance "dtl3", status READY, has 9 handler(s) for this service...
    Handler(s):
      "D007" established:0 refused:0 current:0 max:1002 state:ready
         DISPATCHER <machine: wallach.exlibris-int.il, pid: 853>
         (ADDRESS=(PROTOCOL=ipc)(KEY=#853.2))
      "D006" established:0 refused:0 current:0 max:1002 state:ready
         DISPATCHER <machine: wallach.exlibris-int.il, pid: 851>
         (ADDRESS=(PROTOCOL=ipc)(KEY=#851.1))
      "D005" established:0 refused:0 current:0 max:1002 state:ready
         DISPATCHER <machine: wallach.exlibris-int.il, pid: 849>
         (ADDRESS=(PROTOCOL=ipc)(KEY=#849.1))
      "D004" established:0 refused:0 current:0 max:1002 state:ready
         DISPATCHER <machine: wallach.exlibris-int.il, pid: 847>
         (ADDRESS=(PROTOCOL=ipc)(KEY=#847.1))
      "D003" established:288 refused:0 current:8 max:1002 state:ready
         DISPATCHER <machine: wallach.exlibris-int.il, pid: 845>
         (ADDRESS=(PROTOCOL=tcp)(HOST=ram7)(PORT=32776))
      "D002" established:308 refused:0 current:8 max:1002 state:ready
         DISPATCHER <machine: wallach.exlibris-int.il, pid: 843>
         (ADDRESS=(PROTOCOL=tcp)(HOST=ram7)(PORT=32775))
      "D001" established:295 refused:0 current:9 max:1002 state:ready
         DISPATCHER <machine: wallach.exlibris-int.il, pid: 841>
         (ADDRESS=(PROTOCOL=tcp)(HOST=ram7)(PORT=32774))
      "D000" established:387 refused:0 current:9 max:1002 state:ready
         DISPATCHER <machine: wallach.exlibris-int.il, pid: 839>
         (ADDRESS=(PROTOCOL=tcp)(HOST=ram7)(PORT=32769))
```

2.3. Working With Remote Oracle Tables

The most common network configuration is to have one server where both the DigiTool application and the Oracle tables are located. However, DigiTool also enables you to place Oracle tables on a separate server from the DigiTool 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 DigiTool 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 DigiTool system has many collections, one group of collections may be located on one server and another group of collections 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:

- *Separate Servers for DigiTool Application and Oracle Tables* on page 49
- *Accessing a Remote Table (Without Managing It)* on page 50
- *Stopping Access to a Remote Table* on page 51
- *Managing a Remote Table* on page 51
- *Changing a Table from Remote to Local* on page 53

2.3.1. Separate Servers for DigiTool Application and Oracle Tables

You can set up your system from the very beginning with the DigiTool application on one server and the Oracle tables on another server. To support this configuration, go to the `$dtle_root` directory and open the `dtl_start` file. Make sure that the following variables appear in that file:

```
setenv    DTL_db        <hostname>.<SID>
setenv    TWO_TASK        <hostname>.<SID>
```

The host name 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 loading, 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 [DTL] <*remote user name*>
   Enter password to remote system [DTL passwd] <*remote user password*>

   Notes

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

   2. If the remote DTL 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.dtl3
   ```

4. Use **UTIL A/17/5/2 Create/Recreate All Repository Synonyms** to recreate all repository synonyms.

5. Check in your local host using **sqlplus DTL/DTL** 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 Repository Synonyms** to recreate all repository 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 [DTL]: *<remote user name>*
Enter password to remote system [DTL passwd]: *<remote user password>*

Notes

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

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

4.

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.dtl3
```

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 DTL/DTL
desc <lib>.<table>
```

   in your local host.

8. Use **UTIL A/17/5/2 Create/Recreate All Repository Synonyms** to recreate all repository synonyms.

9. Check in your local host using **sqlplus DTL/DTL** 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 Repository Synonyms** to recreate all repository 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 DTL/DTL** 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 DigiTool 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 DigiTool 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 DigiTool server.

DigiTool software has to be installed only on the DigiTool 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 DigiTool batch jobs on the database server should be taken into consideration. If this option is chosen, DigiTool software must also be installed on the database server.

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

```
      setenv    dtl_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 DigiTool, reboot the server and test. During the test you should ensure that the only DigiTool activity on the local server will be the Web server activation and use, and no other DigiTool 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**
`dtl2`	**DB on remote host**

1. On `ram02` - install oracle and create DB `dtl2`.

2. On `ram01` - install oracle and DigiTool and perform the following:

2.1. Login to DigiTool.
In `$dtle_root/dtl_start` perform the following modifications:

```
#         setenv   dtl_db          ${ORA_HOST}.${ORACLE_SID}
          setenv   dtl_db          ram02.dtl2
          setenv   TWO_TASK          ram02.dtl2
```

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

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

```
ram02.dtl2=(description=
                (address=
                        (protocol=ipc)
                        (key=dtl2))
                (address=
                        (protocol=tcp)
                        (host=ram02.exlibris.co.il)
```

```
                         (port=8003))
                 (connect_data=(service_name=dtl2)))
```

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

DIGITOOL VERSION 3.0

System Administrator's Guide - Preventative Maintenance, Backup and Recovery Policy

Last Update: November 16, 2006
Document Version 1.1
Code: D-ver3-SAGPrevMain-1.1

Table of Contents

1 Introduction

The purpose of this document is to define the maintenance activities that are necessary for the smooth running of the DigiTool application. It is recommended that the system administrator and/or the DBA on site read this document thoroughly and carry out the tasks described within.

A summary of the various maintenance activities can be found in Appendix B. Here is a list of the activities that are described in detail below:

- Ensure sufficient database space.
- Be in archive mode and ensure sufficient disk space for the archiving.
- Back up all system/application components.
- Perform disk cleanup.
- Review Oracle alert log.

☞Note: Whenever an online utility is mentioned in this document, please refer to the relevant documentation (the *System Administrator's Guide – Oracle* for UTILs A and O, or the *System Administrator's Guide – DigiTool 3.0 System Overview* for all other UTILs) for complete details.

2 Sufficient Database Space

2.1 Tablespace Free Space

In order for the Oracle database to function properly, there must be free space in the various tablespaces. The Oracle tablespaces are logical storage units made up of physical datafiles. Use UTIL O to see free and utilized space. Make sure you have at least 10% free in each tablespace or a minimum of 2 GB – the larger of the two. Use UTIL O in case you need to add datafiles to a tablespace.

☞See the chapter on UTIL O in the *System Administrator's Guide – Oracle* for complete details.

2.2 Database Temporary Tables

The DigiTool application creates and uses Oracle tables and indexes for temporary data in VIR01 library. Those library needs to be cleaned periodically by using the special cleanup utility $dtl_proc/clear_vir01. The utility could be activated using Job Manager or UTIL X

☞See the chapter on UTIL C in the *System Administrator's Guide – DigiTool System Overview* for complete details.

3 Archive Log Mode

In Oracle, all transaction made to the database are saved in special files called redo logs. The redo logs function in a cyclic manner. When all redo logs are full, the first one will be reused and its original contents will be overwritten. Archive log mode is a mechanism designed to preserve all the redo logs contents. When in archive log mode, all redo log files are saved to a designated directory. The purpose of saving these files is so that they can be applied to the database in case of recovery scenarios.

In order to ensure the smooth operation of the system, enough disk space must be available at all times for the archived redo logs. Make sure that the archive directory is on a disk with enough space for several days of work. The archived redo logs can be deleted only after they are backed up. By deleting the backed up archive files, free space is made for the new ones being created.

It is crucial to activate the archive mechanism prior to switching to production. If archive log mode was deactivated for some reason, when archive log mode is reactivated, a full database backup must be taken immediately.

4 Backup

Sufficient backup of the application components is crucial for recovery scenarios. Customers are recommended to use the Ex Libris Backup Package to handle their backups. If you choose not to use the Ex Libris Backup Package, contact your local Ex Libris office or distributor. It is important to understand the components of the application before dealing with backup methodology.

5 Components to Back Up

5.1 Database

Backing up the database datafiles is also known as physical backup. There are two types of database backup: cold and hot.

5.1.1 Cold Backup

All the database files are backed up to tape/disk while the database is down. The list of database files to back up is taken from the database data dictionary before shutting down the database. The database (and DigiTool) is down while the backup is being performed, thus no library activities can be held during this time.

Complete recovery of the database can always be done to the time of the backup. In order to bring the database to the time prior to the failure, all archive files which were generated from the time of the backup until the time of the failure must be available.

5.1.2 Hot Backup

All database files (except redo log files) are backed up to tape/disk while database continues to run. Hot backups can be done only if the database is in archive log mode. The list of database files to back up is taken from the database data dictionary.

The database (and DigiTool) continues with normal operation while the backup is being performed, thus normal library activity can be held during this time. Large batch jobs should not be running during the course of the hot backup.

Recovery from Hot backups can only be done if archive files exist. Assuming all archive files are available and in sequence, the recovery will be till the time prior to the failure.

5.2 Archive Files

Archived redo log files are backed up to tape/disk.

When recovering from Hot backups, archive files must be used in order to enable the recovery. When recovering from Cold backups, archive files may be used to minimize data loss and recover till the time prior to the failure.

5.3 Data

5.3.1 Oracle data extract

Backing up the data of an Oracle database is also known as logical backup.

Oracle tables contents are extracted to disk and are backed up to tape/disk. This can be done using Oracle export utility or via special export utilities provided in DigiTool version.

5.3.2 Digital media

The digital media files should be backed up periodically. The placement of media data storage is defined by customer in Storage Rules definitions of Management module (WEB interface).

The list of filesystem root storage points can be viewed using the Management interface, under section Repository/Storage.

In this you'll see the list of storage groups and storages. The storage root entry is the filesystem directory that needs to be backed up.

In the default installation you'll see something similar to this:

```
/exlibris/dtl/j3_1/digitool/home/system/bin/../../profile/storage/
```

5.4 Software

5.4.1 DigiTool Software

Back up the DigiTool application software to tape/disk.

5.4.2 Oracle Software

Back up the Oracle application software to tape/disk.

5.5 Site Configuration

- Back up the file structure of the libraries including exported data.
- Back up the /exlibris/dtl/u3_1 and /exlibris/dtl/j3_1 directories that contain global configuration for all site specific libraries.
- DigiTool backup can be done with/without export (see below).

6 Backup Strategy

Once we understand the components, how they are modified and how often, we can set up a backup plan. With the exception of the Oracle database, the other components are basically directories and files. The more frequently they are backed up, the more up-to-date any data recovered will be in the event of a crash. This will reduce the chance of data loss to a minimum.

As mentioned above, there are two types of backup - physical and logical. Physical backup means backing up the database files. Logical backup means backing up the data extracted from the database tables. Physical backup can be done in one of two methods – cold and hot. Cold backup is done while the database is closed. Hot backup is done while the database is open. A hot backup can run only when the database is running in archive log mode.

Cold backup has an advantage over the hot backup in the sense that a database can be recovered from a cold backup 'as it was at the time of the backup' with no need of additional files. If there are archived redo logs after the time the cold backup was taken, they can be applied. By applying those archived redo logs, the database can be brought up to date with minimum data loss if at all. The hot backup must be restored together with the archived redo logs in order to synchronize the database. Recovery from a hot backup itself without archived redo logs is not possible.

We recommend the following backup policy:

- cold – unless downtime is a major issue, this can run daily
- hot – any day that cold is not run
- archive – run daily
- application configuration – run daily
- export – as frequently as possible
- DIGITOOL application – once every two months and after each upgrade or patch
- Oracle application – once every two months and after each upgrade or patch

See Appendix A: Backup Strategy Guidelines *on page* 11 for strategy guidelines.

7 Disk Cleanup

File systems tend to fill up with temporary files, logs and various other material that can be deleted periodically. The system administrator should do anything and everything possible to avoid the file system(s) reaching full or very high capacity.

Cleanup should be done after backup and not vice versa.

☞ See the chapter on UTIL X in the *System Administrator's Guide – DigiTool System Overview* for complete details.

7.1 Application

The following directories should be cleaned up periodically:

- $TMPDIR (usually /exlibris/dtl/d3_1/tmp)
- $LOGDIR (usually /exlibris/dtl/d3_1/log
- $dtle_scratch (usually /exlibris/dtl/u3_1/dtle/scratch)
- $REPOSITORY_MEDIA (usually /exlibris/dtl/d3_1/repository_media)
- $PERS_COLL_MEDIA (usually /exlibris/dtl/d3_1/pers_coll_media)
- $APACHE_MEDIA (usually /exlibris/dtl/d3_1/apache_media)
- $JB_LOG
  /exlibris/dtl/j3_1/digitool/home/system/thirdparty/openserver/server/default/log
- $J_HOME/profile/work/mets_viewer_tmp
  /exlibris/dtl/j3_1/digitool/home/profile/work/mets_viewer_tmp

All those directories could be cleaned up using UTIL X (mets_viewer_tmp should be clearned up from the maintenance menu of the web-based management module).

☞ See the chapter on UTIL X in the *System Administrator's Guide – DigiTool System Overview* for complete details.

7.2 Oracle Logs

Trace files and the alert log are generated by Oracle under the directory $ORACLE_BASE/admin/$ORACLE_SID. Under this directory, the following sub-directories can be found and may be cleaned from time to time:

- bdump - background processes traces and the alert log
- cdump - core dumps
- udump - user traces

8 Review Oracle alert log

8.1 Oracle Trace Files and the Alert Log

When one of the server / background Oracle processes detects an error, it dumps information about the error to a trace file.

Each database also has an alert_<sid>.log file. The alert file of a database is a chronological log of messages and errors. Messages include information about administrative operations done on the database / tablespaces / rollback segments and errors such as lack of database space and more.

The traces mentioned above and the alert log are placed under the directory $ORACLE_BASE/admin/$ORACLE_SID. In addition UTIL O can be used to review the alert log.

☞ See the chapter on UTIL O in the *System Administrator's Guide – Oracle* for complete details.

9 Recovery Policy

Our recommendation is that the recovery from crash scenarios will be done by a professional DBA.

10 Starting/Stopping DigiTool Application Processes

DigiTool uses various application components when the system is up and running. This includes

- Apache
- JBOSS Application Server
- Background deamons
- WEB and PC Servers

10.1 Apache

Apache can be started and stopped using the apachectl script located in $dtle_root/apache/bin

To stop apache do the following

```
$dtle_root/apache/bin/apachectl stop
```

To start apache do the following

```
$dtle_root/apache/bin/apachectl start
```

10.2 JBOSS Application Server

JBOSS can be started and stopped using supplied scripts located in $jdtlh_bin (e.g. /exlibris/dtl/j3_1/digitool/home/system/bin)

To stop JBOSS do the following

```
cd $ jdtlh_bin
./ jboss_shutdown.sh
```

To start JBOSS do the following

```
cd $ jdtlh_bin
./ jboss_startup.sh
```

10.3 Background deamons, WEB and PC Servers

All other DigiTool related processes can be started and stopped using supplied scripts located in $$dtle_root

To stop DigiTool processes do the following

```
cd $dtle_root
./dtl_shutdown
```

To start DigiTool processes do the following

```
cd $dtle_root
./dtl_startup
```

Appendix A: Backup Strategy Guidelines

As a rule, the more frequent backups are made, the less the likelihood of data loss. As described in this document, we differ between backing up the database as files (a.k.a. physical backup) or as extracted data from the database tables (a.k.a. logical backup). In addition, there are directories/files that are not related to the database that require backup as well (for example, the library structures).

Basically, the ultimate backup strategy would be to run a cold backup of the database daily, including the archived redo logs, and a backup of the site configuration (alephe and libraries with their exported data) on a daily bases. This would mean that in case of need for recovery, recovery can be done from the previous night's backup.

For sites that cannot afford to run cold backup each night, for downtime reasons, hot backup should run each night that cold backup cannot be ran. This, as well, will enable recovery from the previous night's backup.

Sites that cannot run full backup each night (cold nor hot) should do their utmost to set the time intervals between full backups to a minimum. For these sites, the role of the archived redo logs is critical for restoring a full backup done some nights before and reapplying transactions to bring the database up to date. It is important to stress that to perform hot backups you MUST have archived redo logs regardless of the frequency that the hot backup is run.

As for the site configuration, the ability to restore an up to date file depends on the frequency the backup is taken. The library 'tab' directory is probably the directory with the most modifications. Since the 'tab' directory does not take up much disk space, a specific backup of this directory can be taken more often than others.

In addition to performing backup, the backup tapes must be read to check their validity. Run a listing of a full backup tape at least once a week. Besides verifying that the tape is okay as far as the media is concerned, check the listing and make sure all expected directories/files were backed up. Do not take any backup mechanism for granted.

Backup strategy examples:

Here is a chart with examples of backup strategies and their abbreviations. It is important to make sure you are familiar with all the components and that you have a comprehensive backup methodology.

C.A.S.E. – Cold + Archived redo logs + Site configuration + Export
H.A.S.E. - Hot + Archived redo logs + Site configuration + Export
A.S.E. - Archived redo logs + Site configuration + Export
T.V. – Tape Validity check
B.I. – Backup Integrity check

Monday	**Tuesday**	**Wednesday**	**Thursday**	**Friday**	**Saturday**	**Sunday**	**Weekly**
C.A.S.E.	C.A.S.E.	C.A.S.E.	C.A.S.E.	C.A.S.E.			T.V. + B.I.
C.A.S.E.	H.A.S.E.	C.A.S.E.	H.A.S.E.	C.A.S.E.			T.V. + B.I.
A.S.E.	H.A.S.E.	A.S.E.	A.S.E.	C.A.S.E.			T.V. + B.I.

Appendix B: Summary of Periodic Maintenance Activities

Activity	Recommended Time Interval	Method
Clean file system space	Weekly or more frequently (as needed).	Online using UTIL C or by script after backup.
Ensure free database space	Weekly.	UTIL O/14
Delete temporary database tables	Weekly.	UTIL C/3
Backup database	Varies per site. See Appendix A for recommendations.	Ex Libris Backup Package
Backup site configuration	Varies per site. See Appendix A for recommendations.	Ex Libris Backup Package
Backup software	Once monthly or after each upgrade.	Ex Libris Backup Package
Review backup media	Weekly.	Ex Libris Backup Package