



MIMER

Administration Guide for UNIX[®]

Version 7.3

Copyright © 1996 Sysdeco Mimer AB

MIMER version 7.3 Administration Guide for UNIX

December, 1996

Copyright © 1996 Sysdeco Mimer AB.

Published by Sysdeco Mimer AB,

P.O.Box 1713,

S-751 47 Uppsala, Sweden.

Tel +46(0)18-18 50 00.

Fax +46(0)18-18 51 00.

Internet: <http://www.mimer.se>

Produced by Sysdeco Mimer AB, Uppsala, Sweden.

All rights reserved under international copyright conventions.

The contents of this manual may be printed in limited quantities for use at a Mimer installation site. No parts of the manual may be reproduced for sale to a third party.

CONTENTS

1	INTRODUCTION	
1.1	Organization of the documentation	1-1
1.2	Conventions	1-2
1.3	Important terms	1-3
1.3.1	MIMER structure terms	1-3
1.3.2	MIMER administrative terms	1-4
1.4	Acronyms and trademarks	1-4
1.5	Organization of installation and maintenance work.....	1-4
1.6	The MIMER modules	1-6
2	THE MIMER SOFTWARE	
2.1	The installed product	2-1
2.1.1	Directory structure and files.....	2-1
2.1.2	The MIMER Release and Security system	2-2
2.2	MIMER static parameters	2-4
2.2.1	Updating static parameters	2-4
2.2.2	Update MIMER/PG buffer size	2-6
2.2.3	Program and library updates	2-6
3	INSTALLING A MULTI-USER DATABASE SYSTEM	
3.1	The installation procedure	3-1
3.1.1	Verifying the /etc/sqlhosts file.....	3-2
3.1.2	Running the multiinstall script	3-3
3.1.3	Alternative MRS key.....	3-5
3.1.4	Installation errors	3-5
3.2	Optimizing databank file storage.....	3-6
3.2.1	TRANSDB and LOGDB storage.....	3-6
3.2.2	Use of optional directories.....	3-6
3.2.3	Databank directory search method	3-7
3.3	Database lookup.....	3-7
3.4	MULTIADM directory structure	3-8
3.5	Installing several multi-user systems.....	3-10
4	CONTROLLING A MULTI-USER DATABASE SYSTEM	
4.1	The multi-user database system.....	4-1
4.1.1	Monitor server	4-2
4.1.2	Kernel server	4-3
4.1.3	Disk I/O server.....	4-3
4.1.4	Shadow server	4-3
4.1.5	Cleanup server	4-3
4.2	The MULTIADM menu interface	4-4
4.2.1	The main menu	4-4
4.2.2	Starting the multi-user system.....	4-5
4.2.3	Shutting down a multi-user system	4-6
4.2.4	Inquiring system status.....	4-7
4.2.5	Databank files	4-9

4.2.6	Changing the multi-user system access permissions	4-11
4.2.7	Changing system configuration	4-12
4.2.8	Listing the multi-user system log	4-14
4.3	Multi-user system control utilities.....	4-15
4.3.1	Start a multi-user database system.....	4-15
4.3.2	Stop a multi-user database system.....	4-15
4.3.3	Close a multi-user database system.....	4-16
4.3.4	Open a multi-user database system.....	4-16
4.3.5	Report multi-user database system state.....	4-16
4.4	The multi-user database system log file	4-17
4.5	Related environment variables.....	4-17
4.6	Multi-user database system parameters.....	4-19
4.6.1	Size of bufferpool.....	4-19
4.6.2	Cleanup buffer size.....	4-20
4.6.3	Number of mim_kernel servers	4-20
4.6.4	Number of shadow servers.....	4-20
4.6.5	Number of I/O servers.....	4-20
4.6.6	Maximum number of open tables.....	4-20
4.6.7	Maximum number of open databanks.....	4-21
4.6.8	Maximum number of logged on users	4-21
4.6.9	Time between cleanups	4-21
4.6.10	Databank check option	4-21
4.7	Automatic start-up at UNIX restart.....	4-21
4.8	If the multi-user system fails	4-22
4.8.1	Multi-user system corrupt.....	4-22
4.8.2	Using mimkill	4-23
4.8.3	Multi-user system dump directories	4-23
4.8.4	Using dumpall	4-23
5	ORGANIZING DATABANK FILES	
5.1	Background to the MIMER physical file structure	5-1
5.2	About the UNIX file system	5-2
5.3	MIMER databanks in the UNIX file system	5-2
A	UPGRADING FROM V5 TO V7	
B	UPGRADING FROM V7.x TO V7.3	
B.1	Running the upgrade program.....	B-1
B.2	Relinking applications	B-1

1 INTRODUCTION

This guide describes the system databank generation and the maintenance of the MIMER relational database system version 7.3 on a computer running the UNIX operating system. The guide is intended for a MIMER system administrator who is already familiar with MIMER. A good knowledge of UNIX is required, not only to understand this guide, but also because it is essential for being a good MIMER system administrator on a UNIX platform.

Before the actions described in this guide can take place, the UNIX system must be prepared and the distributed files installed. These activities must be performed by the UNIX system administrator and are described in the *MIMER Installation Guide for UNIX*.

1.1 Organization of the documentation

The documentation describing installation and maintenance of MIMER on a UNIX system is divided into two guides:

MIMER Installation Guide for UNIX:

This guide is intended to help the UNIX system administrator prepare the UNIX system and install MIMER.

MIMER Administration Guide for UNIX:

This guide supports the MIMER system administrator in configuring and maintaining the MIMER software and database systems.

The following document is a guide for users of the MIMER system:

MIMER User's Guide for UNIX:

This guide is a supplement to the general MIMER reference manuals, and contains UNIX specific information, such as file specifications. This guide is intended for users of a MIMER system under UNIX.

The following document contains machine dependent information:

MIMER Release Notes for UNIX:

This document includes comments on a machine specific implementation of MIMER, such as keypad layouts, and commands specific to that machine.

1.2 Conventions

The following terms and conventions are used in this guide:

~username	Following the normal csh (C shell command interpreter) syntax, the tilde (~) represents the name of the UNIX home directory for the given user name.
environment variable	A shell variable that can be assigned a string value. See your UNIX documentation for a more thorough discussion.
group id	The numerical identification codes used by UNIX to identify a group of users when checking access privileges to files or other UNIX resources.
user id (uid)	The numerical identification code used to identify a specific login name. The identification number is assigned when the account is created on the particular UNIX system and is then used when checking access privileges to files or other UNIX resources.
path name	A series of directory names separated by “/” characters and ending in a directory or file name.
absolute path name	A path name beginning with the “/” character, which means “from the root directory”, when used as the first character.
search path	A series of path names which are tried one after the other when looking for a specific file.
superuser	The user who has special privileges, such as no file or access restrictions. This user is normally responsible for performing UNIX administration tasks and has a login name called root.

Note! Long path names may, for aesthetic reasons, be split over more than one line in this document. This does not imply that the names may contain spaces or be split over two lines in UNIX.

1.3 Important terms

The following terms are important in the discussion presented in this document.

1.3.1 MIMER structure terms

database	A database is a collection of tables, databanks, idents, domains, etc. All of these objects are defined in the MIMER data dictionary, SYSDB. A UNIX system may contain several MIMER databases operating simultaneously, each with its own SYSDB, but no information may be shared between different MIMER databases. Each database has a dedicated UNIX user account for administration.
databank	A databank corresponds to one UNIX file or raw device. A databank may contain several tables.
table	Tables (or relations) hold all information in the relational database. A table may not be split over several databanks, but a databank can hold several tables.
shadow	A MIMER databank may have one or several shadows. A shadow is a copy of the original (master) databank and is continuously updated by MIMER/DB. If the master databank is lost, it is possible to continue operations from the shadow databank without stopping the multi-user system.

For other terms such as idents and domains, see the *MIMER System Management Handbook*.

1.3.2 MIMER administrative terms

- MIMERADM** **The MIMER software administrator.** This is a specific UNIX user, created to own and administer the libraries, executable files, data files, etc. distributed with MIMER. Only one user is to be created for this purpose, and, for example, could be given the UNIX user name **mimer7**, which will be used in examples shown in this document.
- ~MIMERADM The HOME directory of the MIMERADM user. The installation path **/opt/products/mimer7** will be used in examples shown in this document.
- MULTIADM** **The MIMER multi-user database administrator.** This is a specific UNIX user, created to own and administer a specific database. One user should be created for each MIMER multi-user system to be installed. Suitable UNIX user names are **multi1**, **multi2** etc. Note that the user name given the MULTIADM user will become the main database name, although aliases may be created for each database. Also note that this user must not be the MIMERADM user. The mentioned names will be used in examples shown in this document.
- ~MULTIADM The HOME directory of the MULTIADM user. The database paths **/d1/multi1** and **/d2/multi2** will be used in examples shown in this document.

1.4 Acronyms and trademarks

- MRS** MIMER Release and Security.
- IPC** Inter Process Communication.
- UNIX** UNIX is a trademark registered by X/Open Company Ltd.

(All other trademarks are the property of their respective holders.)

1.5 Organization of installation and maintenance work

Organizing the support documentation into two guides emphasises the division of work between the UNIX system administrator (superuser) and the MIMER system administrators (MIMERADM and MULTIADM(s), which are ordinary UNIX users). The policy is "the less work which must be performed as superuser the better".

The UNIX system administrator (superuser - root) checks the UNIX system resources, creates new users (among them the MIMER system administrator) and initiates the MIMER installation (see the *MIMER Installation Guide for UNIX* for details).

The MIMER system administrator (MIMERADM) owns and maintains the installed MIMER software, and should not have any other responsibilities or be used for any purpose.

The MIMER multi-user database administrator (MULTIADM) owns and controls a MIMER multi-user database system, and should not have any other responsibilities or be used for any purpose. Please observe that there may be several multi-user systems, each having a unique MULTIADM user name.

1.6 The MIMER modules

This version of the MIMER system consists of the following modules, with additional utility programs and/or libraries:

- MIMER/DB Database handler, which includes ODBC support. A multi-user relational database handler, used by all other MIMER modules for access to the database. Also available through subroutine calls from application programs.
- MIMER/ESQL Embedded SQL (Structured Query Language). Used for writing SQL queries embedded in the C language. MIMER/ESQL also includes a dynamic function, which handles the submission of SQL queries generated at execution time.
- MIMER/FMD Forms Manager Development package. Used for managing screen forms in both synchronous and asynchronous terminal environments.
- MIMER/FMR Forms Manager Runtime package.
- MIMER/ISQL Interactive Structured Query Language includes an interactive, editor based, function for creating and maintaining databases (ISQL) and also includes a scrolling version that may be used for execution in batch mode (BSQL).
- MIMER/PGD Program Generator Development package. A 4GL (Forth Generation Language) tool for the development and maintenance of MIMER-based application programs. PG-programs can be developed interactively and used to generate corresponding C code.
- MIMER/PGR Program Generator Runtime package.
- MIMER/PI Programming Interface. MIMER Version 4 compatible database programming interface. Should not be used in new applications.
- MIMER/QF Query by Forms. An end-user facility for simple retrieval and maintenance of data in the database.
- MIMER/QL Query Language. A powerful query language for interpretative access to the database. MIMER/QL procedures allow development of simple MIMER applications, and may also be called from application programs.
- MIMER/RG Report Generator, for producing formatted reports based on the contents of the MIMER database. Simple reports can be defined interactively with a screen formula. A powerful report specification language allows definition of complex reports.

- MIMER/SHD Screen Handler Development package. Used for managing screen forms specifically in asynchronous terminal environments.
- MIMER/SHR Screen Handler Runtime package.
- MIMER/UTIL MIMER Utilities for database management, such as backup/restore, export/import, SQL statistics, databank shadowing control and a readlog utility.

Environment variable

MIMER_HOME Points to the MIMER installation. Should be set by MIMER users. Used by MIMER to locate different installed files in run-time.

Directories

~MIMERADM Symbolic name of the home directory for the owner of the MIMER software installed.

.adm Directory for storage of all the scripts for system administration.

bin Directory for storage of all the executable programs and scripts, available to the MIMER users.

dat Directory for storage of all the data files used and referenced by the MIMER system. The on-line documentation is stored here in PDF-files (see the **readme_doc.txt** for details).

lib Directory for storage of all the libraries and parameter files.

modules Directory containing various MIMER modules.

Files

.dist_version File holding internally used information on version and creation date for the product.

.information File holding machine-specific information on tools and options used during development of the product.

.inst_* Subordinate scripts used by **.install**.

.install This is the script that boots the MIMER menu system and sets correct permissions and ownership for the files included in the MIMERADM environment.

.mimrc Subordinate script used by **.profile**, that sets up the environment and starts up the menu system (see the section discussing "tuning" below).

.odbcinstall Script that installs the runtime part of the Visigenic ODBC Driver Manager product.

2.1.2 The MIMER Release and Security system

When a MIMER installation is completed several programs are available. However, to run a program, you must install a special security key file. The key data is provided by the MIMER agent responsible for the distribution, and should be stored in the **/mimkey7** file.

The security key file contains the following (encrypted) information:

- A list of the modules (and their version numbers) that you are permitted to use.
- The last date for which use is permitted.
- The total number of MIMER database users permitted on the actual machine, system wide, (can be spread over several MIMER multi-user database systems).
- The specific computer node on which the software may be used.

Whenever a MIMER module is started, the security key file is checked to determine if the site is permitted to run that module. If the module has not been licensed, it will be aborted and an error message displayed.

See the *MIMER Installation Guide for UNIX* for details about the MRS key installation.

2.2 MIMER static parameters

From the `mimeradm` menu system it is possible to update the MIMER system parameters, change the size of the MIMER/PG module and recreate executable programs and libraries.

Many of the storage areas are dynamically allocated by MIMER according to the current requirements and available resources, others are declared as fixed arrays. If any of the fixed storage areas or buffer sizes are too small, an error message will be displayed.

All options described below are performed from the `mimeradm` menu system.

2.2.1 Updating static parameters

When you have entered the `mimeradm` menu system, i.e. logged on as the MIMERADM user, choose the option "(2) MIMER Internal Parameter Preparation" and perform the changes desired. When this update is performed the programs and libraries need to be updated.

Parameters that can be changed are listed in the tables below.

DB - Database Manager:

Variable name	Default value	Range	Description
DKSHR1S	600000	100000..1000000	Single-user system bufferpool
DKNDB	40	10..1000	Maximum number of databanks in a single-user system (MIMER/DB error - 16151)
DKNTB	400	100..50000	Maximum number of open tables in a single-user system (MIMER/DB error - 16154)

PI - Programming Interface (Version 4.2 compatible):

Variable name	Default value	Range	Description
STORE4	16384	2048..1000000	Dynamic storage area (MIMER/PI error -102)
STORE4[2]	4000	2000..10000	CL command string buffer
STORE4[3]	512	128..40960	CL maximum token length
STACK	50	10..10000	PI Stack depth

QL - Query Language:

Variable name	Default value	Range	Description
QCI	20000	5000..1000000	Query command interpreter dynamic storage area (MIMER/QL error 54)
QPI	20000	5000..1000000	Query procedure interpreter dynamic storage area (MIMER/QL error 187)
MAXPRO	6	1..10000	Maximum procedure level depth (MIMER/QL error 153)
MAXPTB	100	1..10000	Maximum number of items in a command (MIMER/QL error 8)
LSTR	400	1..10000	Command buffer size (MIMER/QL error 10)
MAXALI	60	1..10000	Maximum number of table references (MIMER/QL error 72)
LWDEF	80	1..10000	Default terminal line width
LCDEF	23	1..10000	Default terminal lines per screen
PWDEF	80	1..10000	Default printer line width
PLDEF	44	1..10000	Default printer lines per page
CCDEF	100	1..10000	Default cumulative message count
PLSDEF	1	1..10000	Number of line feeds per printed line
SHORT	0	0..1	Flag for short printing of help and error messages: 0 = off, 1 = on
VERIFY	1	0..1	Flag for verification on certain commands: 0 = off, 1 = on
PHDEF	1	0..1	Flag for generation of printer file header: 0 = off, 1 = on

FM - Forms Manager:

Variable name	Default value	Range	Description
FSVCOM	80000	4096..1000000	Dynamic storage area. The error message displayed when memory is exhausted is presented as a screen form indicating the number of used forms and fields, initial buffer size, used buffer size, and the attempted allocation size. The application is aborted.

SH - Screen Handler:

Variable name	Default value	Range	Description
POOSH2	8000	4096..1000000	Dynamic storage area. The error message displayed is "Internal space area exhausted".

RG - Report Generator:

Variable name	Default value	Range	Description
STOGR3	40000	4000..1000000	Source code buffer (MIMER/RG error -5027)
STORG2	20000	4000..1000000	Dynamic storage area (MIMER/RG error -5011)

2.2.2 Update MIMER/PG buffer size

When you have entered the **mimeradm** menu system, i.e. logged on as the MIMERADM user, choose the option "**(3) MIMER/PG Internal Buffer Size Preparation**". This option allows you to choose between four different sizes for the PG module's internal storage area, used for processing user PG programs, ("small" is the default size). Choose "large" if, for example, MIMER/SD is going to be used. When this update is performed, the PG related programs and libraries need to be updated - see below.

2.2.3 Program and library updates

When parameters are updated according to the sections above the MIMER libraries, programs and user applications need to be updated.

When you have entered the **mimeradm** menu system, i.e. logged on as the MIMERADM user, choose the option "**(1) MIMER Module Generation**".

In the module generation menu, choose option "**(1) update parameters**", together with the options for the modules which are to be rebuilt. E.g. choose "**1 2 5**" if new values are required for ISQL (and DB). This will replace the parameters in the MIMER libraries with the new values, update the DB shared libraries and relink the ISQL and BSQL programs.

Relink any applications which use the modified libraries.

Example session

The following session shows how MIMER programs and libraries are updated:

- Log on to the MIMERADM user account which will automatically start the **mimeradm** menu script. The following menu is displayed (Note that the symbol **?**: at the end of a line denotes options for help):

```

MMMMM      MMMMM  MMMMM  MMMMM      MMMMM  MMMMMMMMMMMM  MMMMMMMMM
MMMMMMM    MMMMMM  MMMMM  MMMMMM    MMMMMM  MMMMMMMMMMMM  MMMMMMMMMM
MMMMMMM  MMMMMM    MMM    MMMMMM  MMMMMM    MMM    MMM    MMM    MMM
MMMMMMMMMMMMMMMMM    MMM    MMMMMMMMMMMMMMMM    MMMMM    MMMMMMMM
MMM  MMMMM  MMM    MMM    MMM  MMMMM  MMM    MMM    MMM    MMM  MMM
MMMMM  MMM  MMMM  MMMMM  MMMM  MMM  MMMM  MMMMM  MMMMMMMMMMMM  MMMM  MMMM
MMMMM  M  MMMM  MMMMM  MMMMM  MMMM  M  MMMM  MMMMMMMMMMMM  MMMM  MMMM

```

```

V7.3.1 (1) 961109
MIMER System Administration

```

```

(1)  MIMER Module Generation           ?:(11)
(2)  MIMER Internal Parameter Preparation ?:(12)
(3)  MIMER/PG Internal Buffer Size Preparation ?:(13)
(4)  Show generated MIMER programs
(5)  Show module versions in installation
(6)  Display log file
(7)  Escape to shell

(0)  Exit

```

```
Option ( 0 - 7 or 11 - 13 ):1
```

- Select option **"(1) MIMER Module Generation"**.
- Read the introductory text and then choose between generating single- or multi-user MIMER executables.

```
MIMER system generation
```

```
The following menu will display all available modules.
```

```
The license (MRS-key) for this MIMER installation includes a
certain number of MIMER modules. All the modules, including
those not licensed, can be generated. However, only those
programs belonging to licensed modules may be started.
```

```
Select a module to be generated by specifying its
identification number, as shown in the following menu.
Enter the required module numbers on the input line,
separated by blanks, or specify the character "a" for
all modules (this will include modules not licensed).
```

```
Do you want a multi- or single-user environment? (m/s) [m] m
```

- You will now be asked whether the existing programs should be relinked. If you choose to relink, all multi-user systems using this installation should be stopped.):

```
The MIMER multi utilities will now be built:
```

```
Processing multi ... multi updated
```

```
Type <return> to continue... <return>
```

- Press **<return>** to display a menu showing all MIMER modules that can be generated.

```
MIMER system generation
```

```
(1)    update parameters
(2)    db
(3)    esql
(4)    fmd
(5)    isql
(6)    pgd
(7)    pi
(8)    qf
(9)    ql
(10)   rg
(11)   shd
(12)   shr
(13)   util

(0)    Exit
```

```
Select modules to build (0, 1-13, a for all):
> a
```

- Enter **a** at the input line to generate all modules. If only some of the modules are to be generated specify the desired module numbers separated by blanks. For the MIMERADM user called **mimer7**, all building activities are logged in **/tmp/mimer7.log**.
- If the **pgd** module is included in the selection made above, the following prompt will be displayed:

```
Do you want the compiling version of the PGD module? (y/n) [y] y
```

Answer **y** or just **<return>** at the prompt to build the compiled version of the **pgd** module. This version is faster but also larger than the interpretative version that will be generated if **n** is given.

3 INSTALLING A MULTI-USER DATABASE SYSTEM

Provided the MIMER product has been properly installed as described and supported by Sysdeco Mimer AB, any number of MIMER multi-user database systems may be installed, each having their own MULTIADM. However, the configuration of the UNIX kernel usually sets global limits on the number of MIMER multi-user systems which may actually be started. The kernel configuration sets, for example, limits on the number of MIMER multi-user applications which may run simultaneously and on the total size of all MIMER multi-user system bufferpools. The MRS-key multi-user license also sets a limit for the total number of users allowed over all the multi-user systems.

In addition, the amount of main memory and the speed and size of your disk swap space may place practical limits on the ability of your computer to provide acceptable performance for large numbers of MIMER users, whether in single-user or multi-user mode.

The UNIX system administrator (superuser - root) should prepare the UNIX system for the installation of MIMER and install the distributed software. This is described and discussed in the *MIMER Installation Guide for UNIX*. Machine specific notes may be found in the *MIMER Release Notes for UNIX*.

3.1 The installation procedure

The MIMER multi environment is installed by running a MIMER utility called **multiinstall**. This utility is located with the other MIMER executable files. It will install a new MIMER multi-user database system, i.e. selected system databanks and scripts customised with access paths for administrative tools.

When **multiinstall** is executed, you are assumed to be logged into the MULTIADM user account.

3.1.1 Verifying the `/etc/sqlhosts` file

When initially setting up a multi-user database system the `/etc/sqlhosts` setup must be verified. Check that the `/etc/sqlhosts` file includes a `LOCAL` entry with the same name as the `MULTIADM` user, here called **multi1**, and with the first directory path pointing to the `~MULTIADM/multi` sub directory (here `/d1/multi1/multi`). See the *MIMER Installation Guide for UNIX* for a description of the `sqlhosts` file.

Example, including an alias called `PRODUCTION` for the `multi1` database system:

```

.  

.  

LOCAL:  

    multi1      /d1/multi1/multi:/dbdisk1/multi1:/dbdisk2/multi1  

    PRODUCTION /d1/multi1/multi:/dbdisk1/multi1:/dbdisk2/multi1  

.  

.
```

`/dbdisk1/multi1` and `/dbdisk2/multi1` is optional databank directories, e.g. paths to directories on other disks, where selected databank files can be moved (suitably those containing `LOGDB` and `TRANSDB`, see below). These directories, if entered in `sqlhosts`, must exist when executing **multiinstall**.

3.1.2 Running the multiinstall script

The following **multiinstall** example session assumes that the MIMER software is installed in the directory **/opt/products/mimer7** (~MIMERADM) and that the MULTIADM user is **multi1**. As shown by the **ls** command in the example, the **multiinstall** script is not executed before for the **multi1** user since no files are previously installed:

```
# su - multi1
$ ls -a
.  ..
$ /opt/products/mimer7/bin/multiinstall
Terminal type: [vt100] <return>
Initiating the terminal ... ready

MIMER 7.3.1 MULTI-USER DATABASE SYSTEM INSTALLATION UTILITY

Creating bin subdirectory...
Creating multi subdirectory...

This utility will install a new MIMER multi-user database system
below the home directory of a user account created for the purpose
of maintaining that system.
The utility will modify the user environment by installing a MIMER
specific .profile login file, creating bin and multi sub directories,
and installing the system databanks.
Please make sure that the user account is set up to use the Bourne
shell command interpreter. Note that the utility can be used on a
previously installed system to, for example, generate additional
system databanks as it will not overwrite any existing parts of an
installation.

This utility will operate on the system identified in the environment
variable MIMER_DATABASE.

The system defined is: < multi1 >

The system database path is:
< /dl/multi1/multi:/dbdisk1/multi1:/dbdisk2/multi1 >

Do you want to continue with the installation? (y/n) [y] y

INSTALLING A MIMER MULTI-USER SYSTEM

Installing a .profile ...
Installing a ./bin/dbstartup ...
Installing a ./bin/dbshutdown ...
Creating a multi/.multidefs configuration file...

GENERATION OF MIMER VERSION 7.3.1 SYSTEM DATABANKS

Current MIMER installation is located in: /opt/products/mimer7

OK to continue? (y/n) [y] y

The MIMER system requires a system administrator password.
This user is identified by the username SYSADM (or sysadm)
and the password given below.

Type the SYSADM password: XXXXXX
Confirm the SYSADM password: XXXXXX

Databanks will be generated in the primary directory of the
search path defined in the /etc/sqlhosts file:
> /dl/multi1/multi
```

After they have been generated, databanks can be moved to other directories given in the /etc/sqlhosts search path. In particular, it is recommended that the TRANSDB and LOGDB databanks, be located on disks separate from each other and from the other system databank files.

Note that the multi-user system should be stopped when moving databank files to other locations.

The search path for multil is:

```
> /d1/multil/multi:/dbdisk1/multil:/dbdisk2/multil
```

Ok to continue? (y/n) [y] **y**

The following system databanks can be generated:

Option	Databank(s)
1	SYSDB/LOGDB/TRANSDB/SQLDB
2	SYSFM
3	SYSHELP
4	SYSMSG
5	SYSPG/WORKDB
6	SYSQF
7	SYSQL
8	SYSRG/SYSRGOUT
9	SYSSH/CMD
10	BOOKDB/HOTELDB/ROOMSDB
0	Exit

Select databanks to create ("a" for all):

```
> a
```

The operations are logged in /tmp/multil.mimgen

```
SYSDB/TRANSDB/LOGDB/SQLDB ... generated in /d1/multil/multi
```

Is it OK to start the multi-user system? (y/n) [y] **y**

```
Starting the multi-user system ... System started
```

```
SYSFM ... generated in /d1/multil/multi
SYSHELP ... generated in /d1/multil/multi
SYSMSG ... generated in /d1/multil/multi
SYSPG/WORKDB ... generated in /d1/multil/multi
SYSQF ... generated in /d1/multil/multi
SYSQL ... generated in /d1/multil/multi
SYSRG/SYSRGOUT ... generated in /d1/multil/multi
SYSSH/CMD ... generated in /d1/multil/multi
BOOKDB/HOTELDB/ROOMSDB ... generated in /d1/multil/multi
```

```
Exiting database generation!
```

```
Database generation completed!
```

```
Type <return> to continue... <return>
```

```
The installation is completed!
```

```
To employ the .profile and the ./multi/.multidefs files,
please log out and in again.
```

```
$
```

The next time you log in as the **multi** user, a menu will be displayed with the multi-user system control options. This menu is also accessible as a command named **multiadm**.

The following operations are performed by `multiinstall`:

- a user profile is installed, defining a path to the MIMER executable files and a default terminal type
- **bin** and **multi** sub directories are created
- a multi-user system configuration file (**.multidefs**) is created in the **./multi** sub-directory
- databanks are generated in the **./multi** sub-directory according to choices made in the dialogue
- scripts for automatic start and stop of the multi-user system are put under the **./bin** directory (**dbstartup** and **dbshutdown** - see the *MIMER Installation Guide for UNIX*)

The **multiinstall** script can be used on an already installed multi-user system without causing any harm to the existing files, e.g. when the system is to be completed by installing missing databanks.

3.1.3 Alternative MRS key

If this multi-user database system should use an alternative key file name, other than **/mimkey7**, using the `MIMER_KEYFILE` variable, remember to update the **.profile** and **bin/dbstartup** scripts of the `MULTIADM` user to set the environment variable correctly.

3.1.4 Installation errors

The `sdbgen` utility is used to create the system databanks (`SYSDDB`, `TRANSDB`, `LOGDB`, `SQLDB`). Under certain circumstances **sdbgen** may fail, giving the error message:

```
*** Failed to create databank. Error: -16149 ***
Do you want to try again(<No>) ? <return>
*** Incomplete System Databanks created ***
```

This error message originates from the databank I/O routines and is machine specific. In UNIX systems it can often be explained by the fact that there are no free file locks available. The number of file locks available in the system is configurable in the UNIX kernel (see your local UNIX system documentation).

When re-starting the installation procedure, remember to remove the old incomplete database for which the creation was aborted (see error message above).

Note. That similar error messages may occur in other situations for the same reason.

3.2 Optimizing databank file storage

When the multi-user database system has been installed with **multiinstall**, it is initially configured to store all databanks in `~MULTIADM/multi` directory. Unless the initial configuration is changed, this sub-directory will be the working directory of the multi-user system servers and also the default directory where the system servers search for databank files.

We recommend that the initial configuration be changed because storing all databank files in one single directory is dangerous and non-effective. Such a system is sensitive to running out of disk space, does not use multiple I/O channels, and may not survive a disk crash.

Using optional directories as describe below is a simple way of distributing databank files.

3.2.1 TRANSDB and LOGDB storage

Databanks may be lost by accidental removal of a file or by a failing disk. The recovery system built into MIMER will not be able to save a lost databank if both the original databank and the log and transaction databanks are lost at the same time, a situation quite possible if the databank files are stored in the same directory. The following guidelines can be given:

- LOGDB should always be kept on a disk separate from all other databanks in the database.
 - A) If you loose the rest of the system, i.e. the databank files other than the file containing LOGDB, those parts are restored from the latest backup and then updated to current state using the intact LOGDB.
 - B) If the LOGDB file is lost, a new backup is taken on the rest of the databanks and the LOGDB file is recreated.
- TRANSD should if possible be kept on a separate disk, not the same as LOGDB, for performance reasons. It is usually heavily accessed and having it on a separate disk will spread the I/O load efficiently.

A more detailed discussion on data security is found in the *MIMER System Management Handbook*.

3.2.2 Use of optional directories

As mentioned above in the section describing the installation procedure, it is possible to define optional directories in the `/etc/sqlhosts` file. All databank files, except the file for SYSDB, may be moved between the directories in the directory list defined.

The `/etc/sqlhosts` file is read by the multi-user system when the system is started, therefore the system must be restarted to employ a new definition. The first directory in the path is denoted as the "primary path" and is where the newly created databank files will appear. All directories for databank storage must be owned and accessible for read and write by the MULTIADM.

We continue with the example where TRANSDB and LOGDB are moved to optional directories. It is recommended that the LOGDB directory be added at the end of the path in `sqlhosts`, since all new databanks are created in the first usable directory along the path. If LOGDB is located in the primary directory, which is default, the separation between LOGDB and other databanks is lost.

When the additional directories for TRANSDB and LOGDB is created and defined in `/etc/sqlhosts`, the databanks can be moved there using the UNIX `cp` and `rm` commands in sequence (moving files between different UNIX file systems may not always be possible with the normal `mv` command).

Note! The multi-user system must be stopped before moving a databank from one directory to another!

It is possible to move databank files, except for the SYSDB file, in this manner provided their MIMER physical filename, stored in the data dictionary and entered at creation, does not include an absolute path. If an absolute path is used the databank file will be accessed exactly where specified, regardless of the default databank directory and any optional directories. If such databank file is to be moved it must be renamed in the data dictionary using a SQL 'ALTER DATABANK' command.

3.2.3 Databank directory search method

If a list of optional directories is defined in the `/etc/sqlhosts` file, and a relative filename is used, the system searches for the file in all the directories in the list. The search is performed by prefixing the filename passed to the system with the names of the directories one by one until the name of an existing file with appropriate access permissions is found.

A file which is requested by MIMER to be created should not exist anywhere in the search just described, as the creation operation would destroy the existing file. If a databank file to be created already exists in one of the databank directories, an error is raised. If no file is found, a second search will be performed prefixing the directory part of the passed filename (the directory part need not exist) with the names of the optional directories one by one until the name of a directory with appropriate access permissions is found. Finally, the pathname of the file to create is generated from the chosen directory's name and the passed filename.

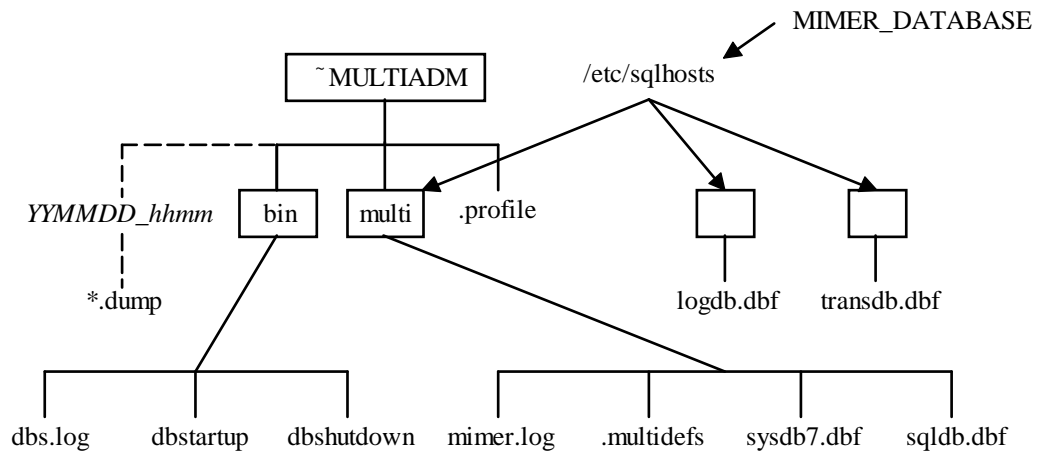
3.3 Database lookup

To enable a user access to a MIMER multi-user database, the user defines the environment variable `MIMER_DATABASE` to point to the database name, for example `multi1` or `PRODUCTION`, that appears in the `/etc/sqlhosts` file. The multi-user database system will then be accessed by all MIMER multi-user executables and application programs that points it out.

In the `/etc/sqlhosts` file one database will be set up to be the `DEFAULT`, which means that a user with no `MIMER_DATABASE` environment variable defined will connect to the `DEFAULT` database multi-user system.

3.4 MULTIADM directory structure

An example MULTIADM directory structure is illustrated below, where the LOGDB and TRANSDB files are indicated to be located on separate storage, looked up using the `/etc/sqlhosts` file and the UNIX environment variable `MIMER_DATABASE`. No user defined databanks are included in the picture:



Environment variable

MIMER_DATABASE Points to the database, defined in `/etc/sqlhosts`, to which a connection is made.

Directories

~MULTIADM	Symbolic name of the home directory for the owner of the multi-user system.
bin	Directory for storage of multi-user system specific tools.
multi	Working directory of the multi-user system servers. This is where the multi-user system log file, the configuration file and the SYSDB databank file resides (other databanks may also be found here if they are not moved to another directory).
YYMMDD_hhmm	A directory which is created when a fatal error is detected which causes the system to shutdown. The directory name reflects the time of the shutdown in a year-month-day-hour-minute format. In the directory a number of system dump files are collected.

Files

*.dump	Multi-user system dump files which can be useful for the MIMER product support department when investigating the error that caused the dump.
/etc/sqlhosts	Valid database names are found in the sqlhosts file where paths for each database are defined. A path can be a directory list or a remote network interface description. See the <i>MIMER Installation Guide for UNIX</i> for more information.
dbshutdown	Shutdown script, suitably used if the shutdown of this MIMER multi-user database system is connected to UNIX system shutdown.
dbstartup	Startup script, suitably used if the startup of this MIMER multi-user database system is connected to UNIX system startup.
dbfs.log	System shutdown and startup log, if using the dbstartup and dbshutdown scripts.
mimer.log	Log file for the multi-user system.
.multidefs	Configuration file, only visible to the ls -a command. Defines the configuration of the multi-user system. Executed by .profile at MULTIADM login.
.profile	Login shell script for the MULTIADM user, always executed at MULTIADM login. Contains the definitions of the environment variables necessary for the operation of the multi-user system and defines the terminal in use, as well as the command to start the multiadm menu program. Only visible to the ls -a command.
sysdb7.dbf	System databank, includes the entire data dictionary. All table definitions, views, users, indexes, etc. are recorded in this databank. The file names of the other databanks in the system are found in this databank. This file should <i>not</i> be moved from the multi sub directory.
logdb.dbf	System databank which should be kept on a disk separate from the other databanks, for data consistency reasons.
sqldb.dbf	System databank used for internal work.
transdb.dbf	System databank which should be kept on a disk separate from the other databank files, mainly for performance reasons. TRANSDB may be stored with the SYSDB file if a separate disk can not be used.

3.5 Installing several multi-user systems

It may be convenient to have several MIMER multi-user systems at the same time, for example if several projects are under development in parallel.

To build several MIMER multi-user systems, simply repeat the instructions above using a different MULTIADM account for each multi-user system. Each MIMER end-user then selects a specific multi-user system by setting the MIMER_DATABASE user environment variable.

See the *MIMER Installation Guide for UNIX* for details, such as IPC resources needed, when installing additional multi-user systems.

4 CONTROLLING A MULTI-USER DATABASE SYSTEM

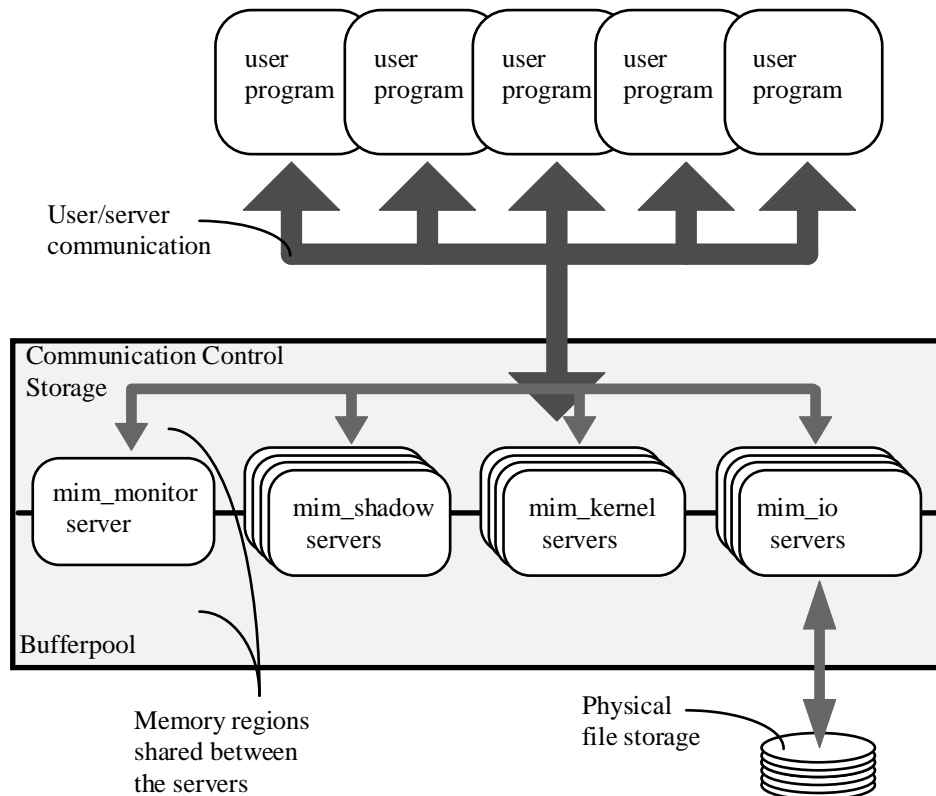
A MIMER multi-user database system may be accessed after it has been installed and started. This chapter describes how to start, stop and reconfigure a MIMER multi-user database system, which is performed by the MULTIADM user.

Starting a MIMER multi-user database system means starting a number of servers (background processes) whose function is to receive user database requests and perform the requested operations against the databanks belonging to the multi-user system. Databank files in a MIMER multi-user system are owned by the MIMER multi-user system administrator, and users have no direct access to the files.

4.1 The multi-user database system

In a MIMER multi-user database system, a number of users may run programs that send database requests simultaneously to a set of MIMER database server processes. The database servers perform the requested operations on the database and send replies back to the user programs. Users have no direct access to the database. The database and the database servers are owned by a separate user and are not accessible by other users, except by using the MIMER inter process communication protocol. A communication connection using this protocol is automatically established and controlled by routines in the MIMER kernel interface library, a library linked to all MIMER multi-user utility programs and user applications (see the *MIMER Installation Guide for UNIX* for detailed information).

The illustration below shows the interface between user programs and the MIMER multi-user database system (the **mim_cleanup** server, which is linked as a user program but included in the MIMER multi-user system, is omitted).



The communication interface between user programs and the MIMER servers uses UNIX IPC. This is described in the *MIMER Installation Guide for UNIX*.

4.1.1 Monitor server

UNIX process name: **mim_monitor**

The monitor server is the main process which creates the others. Once the multi-user system has been started, this server monitors the existence of the multi-user system processes. If a process is found to be missing, the system is brought down and the cause of error is examined.

As long as the **mim_monitor** server is active and the MULTIADM user can access the system, it is possible to control and receive error messages from the multi-user system. If the **mim_monitor** server is dead or hung, the multi-user system can not be controlled using the MULTIADM menu interface or the MIMER utility programs. The individual processes, which are part of the system, must be stopped using the **mimkill** command.

4.1.2 Kernel server

UNIX process name: **mim_kernel**

Database access requested by application processes is taken care of by the kernel servers. These servers retrieve and store data, and communicate with the rest of the processes in the multi-user system.

The number of kernel servers may range from 1 to 32. The actual number of servers is decided at system startup (**mimstart**).

4.1.3 Disk I/O server

UNIX process name: **mim_io**

The I/O processes are used for all database I/O in the multi-user system. By using multiple I/O servers, parallel I/O requests can be issued from several kernel and shadow servers at the same time. The requests are then performed as soon as possible on the databank files.

To improve performance, the UNIX system priority for the MIMER I/O processes can be increased by letting them run in superuser mode. See Chapter 4 of the *MIMER Installation Guide for UNIX* for the actions that must be performed to achieve this.

The number of **mim_io** servers may range from 1 to 64. A rule of thumb is that the number of disk I/O servers should be equal to the number of shadow and kernel servers added together. The actual number of servers is decided at system startup (**mimstart**).

4.1.4 Shadow server

UNIX process name: **mim_shadow**

Shadow servers perform database services such as transaction completion and databank shadowing. This is done in background to the application processes, which means that no application process has to wait for the completion of a shadow update.

The number of shadow servers may range from 2 to 32. The actual number is decided when the system is started (**mimstart**).

4.1.5 Cleanup server

UNIX process name: **mim_cleanup**

This server is used for multi-user system cleanup. The cleanup server releases allocated resources in the event of an abnormal termination of a user process. The cleanup process is activated repeatedly at a time interval decided at system startup (**mimstart**).

The cleanup server employs the user level interface, and is thus linked as an ordinary application program. However, it is controlled by the monitor server.

4.2 The MULTIADM menu interface

The **multiadm** menu system is started automatically whenever a user logs on to the MULTIADM account. It provides an easy-to-use interface for controlling the multi-user system. This menu is described in detail below. When you exit from the menu you will be logged off from the MULTIADM account.

The menu system consist of a main menu and a number of sub menus. In the following paragraphs each menu is displayed as it appears in a real operator session. When a question is raised by the system it will show the alternative answers in parenthesis, (), and the default answer (if any) in brackets, [].

4.2.1 The main menu

The initial menu prints status information of the multi-user system, then lists a number of operations that may be performed on the multi-user system, as shown below:

```

MMMMM      MMMMM  MMMMM  MMMMM      MMMMM  MMMMMMMMMMMM  MMMMMMMMM
MMMMMMM  MMMMMM  MMMMM  MMMMMM  MMMMMM  MMMMMMMMMMMM  MMMMMMMMMMM
MMMMMMM  MMMMMM  MMM      MMMMMM  MMMMMM  MMM      MMM  MMM  MMM
MMMMMMMMMMMMMMMM  MMM      MMMMMMMMMMMMMMMM  MMMMM      MMMMMMMM
MMM  MMMMM  MMM      MMM      MMM  MMMMM  MMM      MMM  MMM  MMM  MMM
MMMM  MMM  MMMM  MMMMM  MMMM  MMM  MMMM  MMMMMMMMMMMM  MMMM  MMMM
MMMM  M  MMMM  MMMMM  MMMM  M  MMMM  MMMMMMMMMMMM  MMMM  MMMM

```

```

MULTIADM for system multi1, using MIMER 7.3.1.
The system is stopped.
No System mail

```

```

(1) System startup           (5) System configuration
(2) System status          (6) System log
(3) Databank files         (7) System mail
(4) Access permissions     (8) Escape to shell

```

```
(0) Logout
```

```
Option ( 0 - 8 ):
```

Note that the lines under the MIMER banner in this menu provide status information (name and status of the system and status of system mail). The first option in the menu changes according to the system status (see examples below, where the MIMER banner is omitted for brevity).

4.2.2 Starting the multi-user system

The multi-user system is started by selecting option **1**, System startup:

```

MULTIADM for system multi1, using MIMER 7.3.1.
The system is stopped.
No System mail

(1)  System startup                (5)  System configuration
(2)  System status                 (6)  System log
(3)  Databank files                (7)  System mail
(4)  Access permissions            (8)  Escape to shell

      (0)  Logout

Option ( 0 - 8 ): 1

```

The screen is cleared and the startup menu displayed, asking the operator to confirm the startup:

```

S Y S T E M   S T A R T U P

System: multi1

The multi-user system will be started using the
following configuration:

Number of 2K buffer pool blocks      : 1024
Number of 16K buffer pool blocks     : 128
Number of 64K buffer pool blocks     : 32
Number of 2K cleanup buffer blocks   : 64
Number of kernel servers              : 4
Number of shadow servers              : 2
Number of I/O servers                 : 6
Maximum number of open tables         : 4000
Maximum number of open databanks     : 200
Maximum number of logged on users    : 10
Time in seconds between cleanups     : 30 seconds
Databank check option                 : complete

Ok to start? (y/n) [y] <return>

Starting the system...done

Type <return> to continue... <return>

```

The multi-user system is now started and the main menu displayed again. Note that the status line and option **1** have changed:

```

MULTIADM for system multi1, using MIMER 7.3.1.
The system is running.
No System mail

(1)  System shutdown                (5)  System configuration
(2)  System status                 (6)  System log
(3)  Databank files                (7)  System mail
(4)  Access permissions            (8)  Escape to shell

      (0)  Logout

Option ( 0 - 8 ):

```

4.2.3 Shutting down a multi-user system

To shut down a running multi-user system, choose option **1** from the main menu:

```

MULTIADM for system multi1, using MIMER 7.3.1.
The system is running.
No System mail

(1)  System shutdown                (5)  System configuration
(2)  System status                  (6)  System log
(3)  Databank files                 (7)  System mail
(4)  Access permissions            (8)  Escape to shell

      (0)  Logout

Option ( 0 - 8 ): 1

```

If there are any users logged on to the MIMER system, these will be listed. You may choose to record the user list in a file:

```

S Y S T E M   S H U T D O W N

The following users are attached to the multi system:

      olos (pid 14526)          linda (pid 14590)
      steve (pid 14544)
(EOF) Type <return> to continue (h for help) <return>

Do you want the list in a file (users.lst)? (y/n) [n] n

```

The shutdown menu is now displayed. Enter a delay time, i.e. number of minutes before shutdown. Type a zero delay time to shut down the system immediately:

```

S Y S T E M   S H U T D O W N

There are users logged on to the system!!

Further logons to the multi system will be prevented
immediately. The system will be brought down after a
specified time has passed. Users still attached at
that time are forced to a stop.

A scheduled shutdown may be cancelled (main menu
option 1).

Time before bringing down the system ( 0 - 60 minutes ) [1]: 0

```

If immediate shutdown is requested, you are asked to confirm the request. Any current user connections will be broken immediately:

```

I M M E D I A T E   S H U T D O W N

Ok to bring the multi system down immediately? (y/n) [y] y

Stopping the system...done

Type <return> to continue... <return>

```

The system is now stopped and the main menu is displayed.

If you enter a non-zero delay time, the shutdown is scheduled. A list of users still logged in to the MIMER system will be displayed 5 minutes before the scheduled shutdown. Option **1** in the main menu allows a scheduled shutdown to be cancelled.

4.2.4 Inquiring system status

The operational status of the multi-user system is displayed in the status line of the main menu. By selecting option **2** in the main menu, information about attached users, the multi-user system configuration and bufferpool parameters can be displayed.

```

MULTIADM for system multil, using MIMER 7.3.1.
The system is running.
No System mail

(1) System shutdown                (5) System configuration
(2) System status                  (6) System log
(3) Databank files                 (7) System mail
(4) Access permissions             (8) Escape to shell

(0) Logout

Option ( 0 - 8 ): 2

```

The terminal is cleared and the status menu displayed. The system status menu lists the following options:

```

M U L T I   S Y S T E M   S T A T U S

System: multil

(1) Show attached users
(2) Show system configuration snapshot
(3) Show bufferpool parameters

(0) Back to main menu

Option ( 0 - 3 ): 1

```

Selecting option **1** displays the attached users menu:

```

A T T A C H E D   U S E R S

The following users are attached to the system:

olos (pid 14084)          linda (pid 14590)
steve (pid 14544)
(EOF) Type <return> to continue (h for help) <return>

```

Type **<return>** to clear the screen and display the system status menu again.

Select **2** to show the system configuration:

M U L T I S Y S T E M S T A T U S

System: multil

- (1) Show attached users
- (2) Show system configuration snapshot
- (3) Show bufferpool parameters

- (0) Back to main menu

Option (0 - 3): **2**

The screen is cleared and the system configuration displayed:

S Y S T E M C O N F I G U R A T I O N S N A P S H O T

System: multil

No. of kernel servers : 4
No. of shadow servers : 2
No. of I/O servers : 6
Time between cleanups : 30 seconds

Type <return> to continue... **<return>**

The third option in the status menu shows the system bufferpool parameters and some system statistics, i.e. the **mimserv** utility will be invoked (see the *MIMER System Management Handbook* for details of **mimserv**).

Type **0** to return to the main menu.

4.2.5 Databank files

Option 3 in the main menu is used to get information about the databank files.

```

MULTIADM for system multil, using MIMER 7.3.1.
The system is running.
No System mail

(1) System shutdown           (5) System configuration
(2) System status            (6) System log
(3) Databank files           (7) System mail
(4) Access permissions       (8) Escape to shell

(0) Logout

Option ( 0 - 8 ) : 3

```

The terminal is cleared and the databank files menu displayed. The menu lists the following options:

```

D A T A B A N K   F I L E S

System: multil

(1) List databank directories
(2) List databank files
(3) Open all available databanks

(0) Back to main menu

Option ( 0 - 3 ) : 1

```

Selecting option **1** clears the system databank files menu and displays the directories available to this multi-user system :

```

D A T A B A N K   D I R E C T O R I E S

Current open databank directories for multil are:

/dl/multil/multi           /dbdisk1/multil
/dbdisk2/multil

Type <return> to continue... <return>

```

Type **<return>** to clear the screen and display the system databank files menu again. Type **2** to display databank file information:

```

D A T A B A N K   F I L E S

System: multil

(1) List databank directories
(2) List databank files
(3) Open all available databanks

(0) Back to main menu

Option ( 0 - 3 ) : 2

```

The screen is cleared and the databank file information is displayed. Note that the **customers.dbf** databank in this example is stored in a raw device file (see the *MIMER Installation Guide for UNIX* for information on creating and using raw device files):

D A T A B A N K F I L E S

The following is a list of the files in the databank file directories for **multil**:

Directory: /dbdisk1/multil

Filename	Type	Size in Mbytes
transdb.dbf	regular	0.154

Total amount used in /dbdisk1/multil: 0.154 Mbyte

Directory: /dbdisk2/multil

Filename	Type	Size in Mbytes
logdb.dbf	regular	0.252

Total amount used in /dbdisk2/multil: 0.252 Mbyte

Directory: /d1/multil/multi

Filename	Type	Size in Mbytes
sysdb7.dbf	regular	0.691
customers.dbf	raw disk	94.531
sqldb.dbf	regular	0.057
syshelp.dbf	regular	0.398
sysmsg.dbf	regular	0.057
BOOKDB	regular	0.117
ROOMSDB	regular	0.031
HOTELDB	regular	0.020

Total amount used in /d1/multil/multi: 96.172 Mbyte
(EOF) Type <return> to continue (h for help) <return>

Type <return> to display the databank files menu again. Type **3** to run the **dbopen** utility:

D A T A B A N K F I L E S

System: multil

- (1) List databank directories
- (2) List databank files
- (3) Open all available databanks
- (0) Back to main menu

Option (0 - 3): **3**

The **dbopen** utility prompts for a username and password. The utility immediately opens all available user defined databanks. If the system was not brought down correctly, it is possible that a dbcheck operation will be performed on a databank when it is first opened after the system has come up again. Use of dbopen may thus be a convenient way of getting all the potential dbcheck operations performed (see the *MIMER System Management Handbook* for details of **dbopen**).

Type **0** to return to the main menu.

4.2.6 Changing the multi-user system access permissions

Selecting option **4** in the main menu displays the access permissions menu.

The access permissions menu displays the status of users' access to the multi-user system and options for changing the access permissions.

```

A C C E S S   P E R M I S S I O N S

System: multil

Multi-user system logon access          : system open
Multi-user system file access (execute) : group (mimer)

(1) Close the system for logons
(2) Change multi-user system file access

(0) Return to main menu

Option ( 0 - 2 ): 1

```

Type **1** to disable new user logons. The menu changes to show multi-user system logon access "system closed" and to allow access to be reopened:

```

A C C E S S   P E R M I S S I O N S

System: multil

Multi-user system logon access          : system closed
Multi-user system file access (execute) : group (mimer)

(1) Open the system for logons
(2) Change multi-user system file access

(0) Return to main menu

Option ( 0 - 3 ): 2

```

The multi-user system file access permissions, for users in general (world) and users in the group that "owns" the multi-user system can be changed by selecting option **2** in the menu above.

Type **0** to return to the main menu.

4.2.7 Changing system configuration

The multi-user system configuration can be changed using the main menu option 5.

Note: A change in the configuration will not affect a running multi-user database system. The change takes effect the next time the multi-user system is started.

The configuration menu displays the configuration parameters which will be used the next time the multi-user system is started. Please note that these parameters are not necessarily the same as the parameters of the running system, which are displayed in the system status menu. A detailed description of the parameters can be found later in this chapter.

C H A N G E S Y S T E M C O N F I G U R A T I O N

The following parameters may be changed to configure the Mimer multi-user system. Please observe that a running system will not be affected by a change in the configuration until restarted.

```

(1) Number of 2K buffer pool blocks      : 1024
(2) Number of 16K buffer pool blocks     : 128
(3) Number of 64K buffer pool blocks     : 32
(4) Number of 2K cleanup buffer blocks   : 64
(5) Number of kernel servers             : 4
(6) Number of shadow servers             : 2
(7) Number of I/O servers                : 6
(8) Maximum number of open tables        : 4000
(9) Maximum number of open databanks    : 200
(10) Maximum number of logged on users   : 10
(11) Time in seconds between cleanups    : 30
(12) Databank check option               : complete

(0) Return to main menu

Option ( 0 - 12 ): 1

```

Selecting **1** displays the menu for changing the number of 2K bufferpool blocks, where a new value may be supplied.

A new value on the number of 2K bufferpool blocks is installed by simply typing the number of pages required:

B U F F E R P O O L 2 K B L O C K S

The current number of 2K blocks in the bufferpool is 1024. The minimum number of blocks is related to the defined number of kernel- and shadow servers for the system. Lowest value ever possible is 64.

```
New number of 2K blocks ( 64 - 100000 blocks ): 2000
```

Once the new value is typed the configuration menu is displayed again.

The rest of the configuration parameters may be changed the same way as described for the bufferpool size.

The configuration menu with updated "Number of 2K bufferpool blocks":

C H A N G E S Y S T E M C O N F I G U R A T I O N

The following parameters may be changed to configure the Mimer multi-user system. Please observe that a running system will not be affected by a change in the configuration until restarted.

(1)	Number of 2K buffer pool blocks	: 2000
(2)	Number of 16K buffer pool blocks	: 128
(3)	Number of 64K buffer pool blocks	: 32
(4)	Number of 2K cleanup buffer blocks	: 64
(5)	Number of kernel servers	: 4
(6)	Number of shadow servers	: 2
(7)	Number of disk i/o servers	: 6
(8)	Maximum number of open tables	: 4000
(9)	Maximum number of open databanks	: 200
(10)	Maximum number of logged on users	: 10
(11)	Time in seconds between cleanups	: 30
(12)	Databank check option	: complete

(0) Return to main menu

Option (0 - 12): 0

Type **0** to return to the main menu.

4.2.8 Listing the multi-user system log

The multi-user system log can be listed using the main menu option **6**.

The system log menu lists a number of options for limiting the output from the log and one option for listing the log. By default, all messages since the last system startup are displayed.

```

S Y S T E M   L O G

System: multi1

Displayed messages  : events warnings errors
Start of display   : last system startup

(1) Select start of display
(2) Select displayed message types
(3) Display selected log messages
(4) Dump selected log messages to a file
(5) Empty the log (only when the system is stopped)

(0) Return to main menu

Option ( 0 - 5 ): 3

```

Options **1** and **2** set selection criteria for the log display. Option **3** displays the log page by page on the screen:

```

L I S T I N G   L O G

-----
Message type:  Event      Date:          Tue Nov 13 08:18:46 1996
Origin:       mim        no:   0       Pid:          12172
Source file:  mimstart.c          Line number:  1281
-----

      *** SYSTEM STARTING ***
      System startup routine called
-----

Message type:  Event      Date:          Tue Nov 13 08:18:54 1996
Origin:       mon        no:   0       Pid:          12178
Source file:  mim_monit.c          Line number:  1832
-----

      *** SYSTEM STARTED ***
      3244 page bufferpool, 30 second cleanup interval
      4 kernel servers, 2 shadow servers, 6 io servers, 1 cleanup
server
      (EOF) Type <return> to continue (h for help) <return>

```

Type **<return>** to display the system log menu again.

Option **4** lists log messages to a file, and option **5** clears the log.

4.3 Multi-user system control utilities

The utilities described below, in general, perform the same functions as the corresponding **multiadm** menu options. You are encouraged to use the **multiadm** menu options whenever possible. However, the utilities can be used as commands in a regular shell to control the MIMER multi-user system.

4.3.1 Start a multi-user database system

The **mimstart** utility is used to start a MIMER multi-user database system:

Usage: `mimstart [-D]`

Argument:

-D Start a default configured system, i.e. not using the **.multidefs** parameter file.

By default, i.e. without any argument, **mimstart** will use the configuration parameters set in the configuration file, `~MULTIADM/multi/.multidefs`. This will employ the settings made at the latest system configuration.

The **mimstart** utility is the back end utility used to start the MIMER multi-user database system. **Mimstart** must be run by a **MULTIADM** user and starts up the corresponding multi-user database system.

4.3.2 Stop a multi-user database system

The **mimstop** utility is used to stop a running MIMER multi-user database system:

Usage: `mimstop [-n | time | -u]`

Arguments:

-n Shutdown the system immediately

time Shutdown the system after *time* minutes, where *time* is a numeric argument.

-u Cancel a previous "mimstop *time*" shutdown request

By default, i.e. without any argument, **mimstop** will schedule a 1 minute delayed shutdown.

The **mimstop** utility is the back end utility used to stop a MIMER multi-user database system. It stops the system that corresponds to the current user, which is presumed to be a **MULTIADM** user. When the system is shut down the servers are stopped immediately without checking for existing users. The multi-user database system will not let the pending database requests finish (the completion of these transactions will be taken care of at next startup).

Note! When stopping the multi-user system using the **multiadm** menu interface a check will be performed to see whether any users are using the system and ask for confirmation before stopping it.

4.3.3 Close a multi-user database system

The **mimclose** utility is used to close a running MIMER multi-user database system:

Usage: mimclose

The **mimclose** utility is the back end utility used to close a MIMER multi-user database system, i.e. the system is set so that no further MIMER logins are allowed. This can be useful when preparing a shutdown for maintenance. It closes the multi-user system that corresponds to the current user, which is presumed to be a MULTIADM user. The system will be opened again when the **mimopen** command is executed.

4.3.4 Open a multi-user database system

The **mimopen** utility is used to open a closed MIMER multi-user database system:

Usage: mimopen

The **mimopen** utility is the back end utility used to open a previously closed MIMER multi-user system. It opens the multi-user database system that corresponds to the current user, which is presumed to be a MULTIADM user.

4.3.5 Report multi-user database system state

The **mimstat** utility is used to look at the state of a running MIMER multi-user database system:

Usage: mimstat [-s | -a | -u | -c | -S | -A | -U | -C]

Arguments:

- s Show the state of the multi-user database system (default)
- a Show users access to the multi-user database system
- u List users attached to the multi-user database system
- c List the running multi-user database system configuration snapshot
- S Like "-s" (script formatted *)
- A Like "-a" (script formatted *)
- U Like "-u" (script formatted *)
- C Like "-c" (script formatted *)

(* The term "script formatted" means that just one word is reported, e.g. "running" or "stopped", so that it can be used from a command script.)

The **mimstat** utility is the back end utility used to check the state of a MIMER multi-user database system. It reports the state of the system that corresponds to the current user, which is presumed to be a MULTIADM user.

4.4 The multi-user database system log file

The MIMER multi-user servers perform logging of important events in a log file called **mimer.log** located in the `~MIMERADM/multi` directory.

The log file may be read directly by any suitable UNIX command, but the log listing option in the **multiadm** menu interface gives a formatted output and the opportunity to select different parts of the log file, etc.

4.5 Related environment variables

Programs started by a UNIX user have access to all the user's exported shell variables, also called environment variables. The MIMER multi-user system servers have access to the exported shell variables defined in the MULTIADM's login shell. Some of these variables, defined in the MULTIADM's **.profile** and the **multi/.multidefs** file, are used to modify the behaviour of the multi-user database system servers.

In addition to the variables discussed below an environment variable for locating **shared libraries** in runtime may be defined. The name of this variable is different between the various UNIX implementations (see the *MIMER User's Guide for UNIX* section on 'Shared library lookup'). For the MULTIADM it is defined automatically.

The following environment variables should be regarded:

MIMER_DATABASE

Used to select a directory path (for a local database) or a remote network database from the `/etc/sqlhosts` file. The given database information will be used to establish the database connection. For local databases a directory path will be looked up, where the first directory in the path is the location of the SYSDB file. Other databanks can be moved to directories included in the path given. This variable is automatically defined for the MULTIADM user.

MIMER_HOME

Used to point out the location of the MIMER software, i.e. it should hold the `~MIMERADM` directory path. This variable is automatically defined for the MULTIADM user.

MIMER_KEYFILE

`MIMER_KEYFILE` points to the MRS key file that contains the encrypted MIMER license. If the environment variable is undefined (or does not point to an existing file), the file `/mimkey7` will be used.

MIMER73_PATH

MIMER73_PATH is used as a search path for sequential files used by the MIMER system. If you have no MIMER73_PATH variable defined, the system will attempt to access files relative to the current working directory. Several directory path names can be set up separated by colons, in which case the system will search through the paths in the order listed:

```
"pathname1:pathname2"
```

When trying to open an existing file, each directory named in MIMER73_PATH will be checked, one after the other, in an attempt to locate the desired file.

When attempting to create a new file, files are created in the first writable directory found mentioned in MIMER73_PATH, but only after all of the succeeding directories have first been checked for the existence of a file with the same name.

This variable is automatically defined for the MULTIADM user.

MIMER73_EXTEND

If this variable is defined when the multi-user database system is started, the number of pages by which a MIMER databank file is extended will be changed to the specified number of 2K blocks (default 25 blocks).

This variable is accessed by the low level input/output routines during initialization at system startup. Thus, if the value needs to be changed, the multi-user system must be stopped and restarted after the value assigned to the environment variable has been changed.

Apart from these MIMER-specific environment variables, there are ordinary UNIX shell variables that are used to run MIMER:

PATH

PATH is not used in any special manner by MIMER. But, as usual, it must be defined so that MIMER user application programs and command scripts can be located by the UNIX command shell.

TERM

MIMER modules using terminal function keys (i.e. based upon MIMER/FM or MIMER/SH) will not work properly with the operating system supplied terminfo definitions. The software distribution provides MIMER specific terminfo definitions for supported terminal types. TERM should be set to one of the supported terminals for a proper behaviour. The supported terminals are listed in the *MIMER Release Notes for UNIX*.

TZ

The TZ (for Time Zone) environment variable is important for the correct display of time stamps in the logging function. TZ is often set and exported in the file **/etc/profile** for system wide use.

4.6 Multi-user database system parameters

Each MIMER multi-user system has a number of parameters that can be adjusted to improve the performance of the system. The values of these parameters are stored in the system's **.multidefs** file found in the `~MULTIADM/multi` directory. The parameters can be modified using the **multiadm** menu interface or by editing the **.multidefs** script. A running multi-user system will not employ new parameters until it has been stopped and started again.

4.6.1 Size of bufferpool

The bufferpool is divided into sections of different block sizes; a 2K block section, a 16K block section and a 64K block section. The number of blocks in the three sections may be modified independently of each other.

Larger bufferpools are required for systems with more servers and users. The size of the bufferpool limits the maximum number of shadow and kernel servers that can be started. The number of servers can be estimated by the following formula:

$$\text{Kernels} + \text{Shadows} \leq \text{MIN}(2\text{K-pages}, 16\text{K-pages}, 64\text{K-pages}) / 5$$

With 1024 2K-pages, 128 16K-pages and 32 64K-pages, we get:

$$\text{Kernels} + \text{Shadows} \leq 32/5$$

With 2 shadow processes (which is the minimum) we get:

$$\text{Kernels} + 2 \leq 6$$

A maximum of four kernel servers may be started.

Note: Generally a larger bufferpool gives a better performance. If the primary memory on the computer system is too small for the bufferpool, as well as the user and server process virtual memory working sets, virtual memory swapping may cause a decrease in performance.

Note also that the maximum amount of shared memory which may be allocated to a given process, or a located system wide, may be limited by the UNIX kernel parameters or by the UNIX implementation. If the kernel imposes limits on the MIMER configuration we recommend you to try to rebuild the kernel with larger amounts of the needed resources. Refer to the *MIMER Installation Guide for UNIX* for information on which kernel parameters may be modified to increase the necessary UNIX kernel limits.

4.6.2 Cleanup buffer size

This is used by the cleanup facility, freeing resources allocated to abnormally terminated user programs. The cleanup area needed is proportional to the number of users logged in to the multi-user system, and the number of tables each user has opened. The number of 2K pages to use for the cleanup area should be:

$$\text{Max_tables_per_user} * \text{Max_users} / 100$$

A cleanup area of 64 2K pages is the default.

4.6.3 Number of `mim_kernel` servers

The `mim_kernel` servers execute user's database requests. The number of kernel servers should reflect the maximum number of users that will be concurrently connected to the multi-user system. A user request typically employs a kernel server for fractions of a second, but a request may also span over several seconds or even minutes. The default value is 4 `mim_kernel` servers but as many as 32 may be started. Practical tests have shown that hundreds of users can be supported by 6 to 8 kernel servers.

4.6.4 Number of shadow servers

A multi-user system not using databank shadowing will run well with 2 shadow servers. If databank shadowing is used 3 or 4 servers will be appropriate (it is possible to start 32 `mim_shadow` servers). See the *MIMER Shadowing Reference Manual* for a more thorough discussion of the number of shadow servers to use.

4.6.5 Number of I/O servers

If maximum system throughput is essential the number of `mim_io` servers should exceed the total number of `mim_kernel` and `mim_shadow` servers. A maximum of 64 `mim_io` servers may be started.

The I/O servers can be made to run at a higher priority (giving faster I/O response times). This is further described in the *MIMER Installation Guide for UNIX*.

4.6.6 Maximum number of open tables

The maximum number of tables that may be opened concurrently in the multi-user system is limited by a tuneable parameter. The default limit is 4000. Values between 128 and 20000 are valid.

4.6.7 Maximum number of open databanks

The maximum number of databanks that the system can open is limited by a parameter. The value of the parameter may range from 100 to 2000. The default is 200.

Note: The UNIX kernel puts a limit on the number of files a process can open (and the total number of open files in the system). If you increase the maximum of databanks limit, you may have to re-configure the UNIX kernel.

4.6.8 Maximum number of logged on users

This parameter limits the number of users that may be logged in on the multi-user system at the same time. The sum of this parameter for all started multi-user systems can not exceed the number of users specified in the MRS key.

This parameter also controls the number of UNIX semaphores that are allocated, so specifying a larger value than what is actually needed may waste system resources.

4.6.9 Time between cleanups

The `mim_cleanup` server is designed to activate itself at regular time intervals and sleep in between. The sleep time is user selectable from 10 to 1800 seconds. Each cleanup performed will marginally increase the load on the system and momentarily lock certain MIMER resources, so care should be taken when using a short interval. A long interval between cleanups may keep users waiting trying to access a resource locked by a crashed user process. The default interval is 30 seconds.

4.6.10 Databank check option

The first time a databank is opened after the multi-user system has been started, the databank is checked to find out if it was properly closed when the system went down. If a databank was not properly closed, a check of the databank internals is performed. The check may be a complete check or a quick check. The complete check goes through both the index and the data pages of the databank. The quick check saves time by checking only the index pages of the databank, but corrupted data pages will not be detected.

4.7 Automatic start-up at UNIX restart

One or more MIMER multi-user systems can be automatically started when the UNIX multi-user system is rebooted after a shutdown, power failure or similar problem. This type of installation can only be performed by the UNIX system administrator having access to the UNIX system control files. The necessary steps are described in the *MIMER Installation Guide for UNIX*.

4.8 If the multi-user system fails

A running multi-user system which encounters a fatal problem will automatically write an error message to the system log file and shutdown. User applications connected to the system at the time of the shutdown will terminate. Normally the system can be restarted if the cause of the shutdown does not prevent a new start.

If it appears that a system failed because an IPC resource was exhausted, the UNIX system administrator should be informed, and it may be appropriate to rebuild the UNIX kernel with a larger supply of the failing resource. Check **mimer.log** for help in tracing the cause of a failure.

4.8.1 Multi-user system corrupt

If a system loses its **mim_monitor** server, the system will continue to run. In case of further trouble the system will not shutdown in the way described above, as the shutdown is a responsibility of the **mim_monitor** process. A system which has reached this state is corrupt and should be stopped and restarted. Unfortunately the **multiadm** menu or the **mimstop** utility cannot be used to bring the system down since these utilities depend on the **mim_monitor** process in order to function.

The following will happen if the MULTIADM user is logged on to a corrupt system:

```
Terminal type: [vt100] <return>
```

```
Initiating the terminal ... ready
```

```
System directory: /dl/multi1/multi
```

```
System message:  No mim_monitor server running - the system is corrupt
```

```
System directory: /dl/multi1/multi
```

```
System message:  Remainders of a corrupt Mimer system found
```

```

The system is corrupt or incorrectly installed, please check
the installation and consult the manual on how to kill and
clean up a corrupt system. Aborting.
```

```
Type <return> to continue... <ctrl-c>
```

Instead of pressing **<return>** "to continue", use **<ctrl-c>** (interrupt) to escape out to the shell, from where UNIX commands and the **mimkill** command can be run to clean up the situation (see example that follows).

If a system with no **mim_monitor** process is to be stopped, the server processes have to be killed and the IPC resources removed by manually issuing a number of UNIX commands. A script called **mimkill** does this for you, provided it is supplied an argument specifying the name of the MULTIADM who owns the system which is to be stopped.

4.8.2 Using mimkill

In the following example, the **multi1** MULTIADM checks the multi-user system processes. The system has no **mim_monitor** process so the MULTIADM shuts it down using **mimkill**.

```
multi1: ps -u multi1
  PID TTY          TIME COMMAND
 17019 ttys3      0:00 ps
 16999 ttys3      0:01 sh
 14797 ?           0:22 mim_kernel
 14788 ?           0:00 mim_io
 14789 ?           0:00 mim_io
 14802 ?           0:00 mim_shadow
 14790 ?           0:00 mim_io
 14791 ?           0:00 mim_io
 14794 ?           0:00 mim_io
 14796 ?           0:30 mim_kernel
 14793 ?           0:00 mim_io
 14798 ?           0:08 mim_kernel
 14795 ?           0:04 mim_kernel
 14805 ?           0:04 mim_cleanup
 14799 ?           0:00 mim_shadow
multi1: mimkill multi1
```

4.8.3 Multi-user system dump directories

If the multi-user system should crash, a directory will be created with the name **~MULTIADM/YYMMDD_HHMM** (YYMMDD being the current date, and HHMM is the time of the crash). This directory will contain a copy of the file **mimer.log**, and some dump files. These files are of interest to MIMER support personnel only (if the reason for the crash should be investigated).

If the incident is to be investigated by MIMER support, the whole directory should be included with the report sent to Sysdeco Mimer AB. To create a single compressed file of the directory the commands given in the following example can be used:

```
multi1: tar cvf dump.tar 950920_0850
multi1: compress dump.tar
```

When completed, the file **dump.tar.Z** will be available for delivery to the Sysdeco Mimer product support department.

4.8.4 Using dumpall

When the MIMER multi-user database system is in a state where it seems to hang or loop, the **dumpall** utility should be executed to do a complete manual dump of the system (if possible). When the **dumpall** session is completed a directory is created and named as shown in the following example:

```
multi1: dumpall
multi1: ls
960617_160551.dumpall      bin          multi
```

The directory **dumpall** directory created contains several dump files with diagnostic information that may be useful to Sysdeco Mimer product support department in determining the cause of the problem. A compressed summary file called **dumpall.tar.Z** is also created in the directory. This file should be sent to Sysdeco Mimer AB for analysis.

5 ORGANIZING DATABANK FILES

This chapter discusses MIMER databanks and how MIMER interacts with the UNIX file system. A few hints on databank integrity and system performance are also presented. For a complete discussion on protecting your data, see the *MIMER System Management Handbook*. Information given in this chapter should be treated as a set of guidelines.

All MIMER databanks appear as files in the UNIX file system, except those located on a raw device. These files may become large and/or numerous. It is the MULTIADM's responsibility to maintain an efficient file system structure. If this is neglected, system performance will decrease and the system may become difficult to maintain.

A part of the maintenance procedure involves UNIX system backup and restore, for which a strategy and structure must be set up to include the MIMER system databases. Any system without a complete and correct backup/restore procedure runs the risk of being lost. See the *MIMER System Management Handbook* for a detailed discussion on backup/restore.

5.1 Background to the MIMER physical file structure

"Databank" is the MIMER term for the physical file in which one or more MIMER tables are stored. While any given table must be fully contained within a single databank, a given databank can contain any number of tables.

A few databanks in each database, the MIMER system databanks, are fundamental to the functioning of the database. The most important of these is the main system databank, SYSDB, which contains certain tables, generally known as the data dictionary. Among other things, these tables record which users are known to the MIMER system, what type of access rights these users have, which databanks are currently connected to the MIMER system and the tables they contain.

The SQLDB, TRANSDB and LOGDB system databanks are also critical in a production system. These databanks do not contain tables in the normal sense, but are instead used to support MIMER transaction handling and recovery.

Except at the time when tables are defined, the existence of databanks is transparent to users and application programs. When tables are opened, the data dictionary is referenced by MIMER routines to locate the required table and, if permissible, to make it available. This allows considerable flexibility in the physical placement of data on the computer system.

5.2 About the UNIX file system

Files in UNIX are accessed through a hierarchical directory structure. A given file must reside completely within a single file system which is usually synonymous with a portion of a single physical disk unit (a UNIX disk partition).

As with other operating systems, UNIX provides the best performance if physical I/O is balanced as much as possible over all of the available physical disk devices. Because of the manner in which file space is allocated to files as they grow, UNIX is more sensitive to fragmentation of files than some other operating systems. (Fragmentation means that physical placement of the data for a file is not contiguous.) The fuller a UNIX file system becomes and the longer it has been in use, the more likely it is that the various physical blocks that make up a file are spread widely over the disk surface.

There is always a danger that a UNIX file system can become damaged due to the failure of a system program, a power failure, or even a defect in the actual disk. In such a case, there is no guarantee that the data can be recovered. This is why adequate backups are required at all times. See the *MIMER System Management Handbook* for more information.

5.3 MIMER databanks in the UNIX file system

There are a number of things which a MIMER system administrator can do to improve the performance and reliability of a MIMER database with respect to the allocation and use of the physical file space.

The issues discussed regarding tuning a UNIX system apply equally well when the system is used for running non-MIMER applications. Only a couple of special considerations are mentioned which apply specifically to MIMER.

1. Whenever possible, pre-allocate file space for databanks when a file system is "young" and relatively empty.

The MIMER databank creation routines allow the creation of a new databank of any specified size. A databank created with the size it will need in production, will generally be accessed more efficiently than one created small and allowed to grow in small increments. There is a special standalone MIMER utility designed to be used by the UNIX system administrator for creating a physical file to contain a new databank, or for extending an existing databank file in large increments. This utility, **dbextend**, is also able to create or extend files which are larger than the maximum user-file-size limit imposed by the UNIX kernel (**ulimit**). See the *MIMER Installation Guide for UNIX* for details on the dbextend tool.

A databank may also be extended using an ALTER DATABANK statement in one of the MIMER SQL utilities.

2. If more than one physical disk unit is available, the various databanks should be distributed across the available physical disk units in order to balance the I/O load on a system.

To maximize the distribution of I/O across disks, place databanks on physical disks in such a way that databanks which are accessed at the same time are on separate units. It is recommended to have each of the TRANSDB and LOGDB databanks on separate disks, separated from the rest of the databanks. (If this is not possible, group TRANSDB with SYSDB on one disk, and have LOGDB separated from the other databanks on another disk, for data security reasons in case of a disk failure).

Which databanks to store on which physical disk unit will depend on exactly how they are used in practice. External factors such as the speed of disks and the amount of virtual memory paging can often be of greater importance to overall performance than MIMER's database oriented I/O.

3. To enhance performance, TRANSDB may be placed on a RAM disk if sufficient memory is available. Note that if the RAM disk is not backed up with batteries, the contents will be lost if there is a power failure. This will almost certainly lead to an inconsistent database. Do not use a RAM disk if the survival of its contents cannot be guaranteed in the event of a power or system failure.

A more detailed discussion on data security is found in the *MIMER System Management Handbook*.

A UPGRADING FROM V5 TO V7

This appendix describes the actions to be taken to upgrade an existing MIMER version 5 (V5) installation to version 7 (V7).

Phase 1: Install a separate V7 installation

This installation will not disturb the V5 environment if you observe the rules given in this appendix carefully.

Define new MIMERADM and MULTIADM users under new groups. Make a complete installation of MIMER V7 as described in the *MIMER Installation Guide for UNIX* and in this guide.

When the multiinstall script is used to initiate a new multiadm user, a fresh databank environment will be created. This environment can be used to test the V7 installation before proceeding further.

Phase 2: Prepare for migration from V5 to V7

Relink all applications that should run with MIMER V7. Install the database description /etc/sqlhosts file.

Phase 3: Backup the V5 databanks

Stop the V5 multi-user database system to make sure that the V5 databanks are not in use.

Note: The conversion is irreversible, and that the databanks cannot be accessed by V5 again, once converted. If you want to keep the V5 system, make a backup copy of all databanks before proceeding with the conversion.

Phase 4: Convert the databanks

Convert the databanks by running the conv57 utility.

Start the V7 multi-user system, with the MIMER_DATABASE environment variable set to the database name that points to a directory path list in /etc/sqlhosts corresponding to the MIMER52_PATH of the V5 multi-user system.

When the V7 multi-user system has been started, the former V5 users can run the applications that have been relinked to V7.

Phase 5: Update tool-specific databanks (SYSxx)

Some MIMER modules use specific databanks. These are:

Module	Databank
FMD	SYSFM
ISQL	SYSHELP
PGR	SYSPG
QF	SYSQF
QL	SYSQL
RG	SYSRG
SHD	SYSSH and CMD
UTIL	SYSMSG

It is recommended that these databanks are updated with V7-specific data. (Since these databanks are converted from V5, they contain only V5-specific information.)

To reload the databanks, follow these steps:

- Make sure that `~MIMERADM/dat` is at the end of the directory list in the `MIMER73_PATH` environment variable.
- Run the UTIL program and choose the SYSxxGEN utility.
- For each of the databanks in the menu, select the "Generate Databanks" option.

General notes

- Please note that an application linked with MIMER/DB V5 cannot use a V7 multi-user system (and a V7 application can not use a V5 multi-user system).
- Please make sure that applications (or MIMER modules) linked to different versions of MIMER/DB are kept separate. In particular, the environment variable `PATH` should not refer to both V5 and V7 directories.
- For detailed information concerning upgrade from the previous release of MIMER see the *MIMER Release Notes for UNIX*.

B UPGRADING FROM V7.x TO V7.3

MIMER version 7.3 adds some new information in the system data dictionary. Before a version 7 database can be used in a version 7.3 environment, it must be upgraded. This is done by the **upgr73** program.

The V7.x to V7.3 upgrade is irreversible. It is strongly recommended that you take a backup copy of your database before starting the upgrade.

The upgrade process does not alter any user databanks. The time needed to do the upgrade is *not* proportional to the size of the database, as only system databanks are affected.

For detailed information concerning upgrade from the previous release of MIMER see the *MIMER Release Notes for UNIX*.

B.1 Running the upgrade program

The upgrade program is run in single-user mode. You must move to the directory where the **sysdb7.dbf** file is before running the program. Login as the MULTIADM user, stop the multi-user database system, escape out to the shell and perform the following:

```
multil: cd multi
multil: upgr73
This program upgrades the system databank from version 7.1
or version 7.2 format to version 7.3 format.
```

The system databank to be upgraded must be located in the current directory.

Before continuing, you are strongly advised to take a full backup of all your databanks.

```
Do you want to Continue or Quit (C/Q): c
```

```
The system databank is successfully upgraded to version 7.3
multil:
```

B.2 Relinking applications

Applications that connect to a local multi-user database system have to be relinked with the new version libraries. An older version 7 application will not be able to locate a local version 7.3 multi-user database system.

Older version 7 applications can connect to remote version 7.3 multi-user database systems without any modifications. However it is recommended that these applications are also relinked so that they can use the enhanced client/server protocol.