

MIM System Administration Guide



**12301 Research Blvd.
Building IV, Suite 410
Austin, TX 78759**

U.S. Help Desk Phone: +1-800-546-9646 (or direct +1-512-697-3000), select ext. 3400

U.K. Help Desk Free Phone: 0800 032 6063

Europe Help Desk Phone: +44 20 7190 2947

Help Desk Email: support@lim.com

+1-512-697-3001 (Fax)

Build Number: 4.6.29

Part Number: 083_91

Date: March 6, 2008

Copyright © 2001-2008 by Logical Information Machines, Inc.

Patented May, 1995 U.S. Patent No. 08/392, 612

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of Logical Information Machines, Inc.

Restricted Rights Legend

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subdivision (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at 252.227-7013.

Logical Information Machines, Inc.
120 North LaSalle Street
Suite 2150
Chicago, IL 60602

(312) 456-3000

Product names mentioned herein are for identification purposes only and may be trademarks and/or registered trademarks of their respective companies.

While every precaution has been taken in the preparation of this manual, we assume no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of the information contained herein. Logical Information Machines, Inc. may revise this publication from time to time without notice.

Table of Contents

Chapter 1: MIM System Administration (Solaris)	1
MIM Server Basics	1
Architecture	1
Tree Structure	2
Installation	4
MIM Server Permissions	10
Free Pass	10
xmimsvr.acl File	10
Examples	11
MIM Database Write Permissions	12
Permission Algorithms	12
Troubleshooting	13
MIM Server Administration	14
Updating the Database	14
Starting/Stopping the MIM Server	15
System Monitoring	16
Printing	17
MIM Server Backups	18
What to Backup	18
Scheduling	18
Restoring	18
Installing Test Servers	19
Setting Up a Test Server for Data Loading	19
Create a New Database for Customer Data Only	19
Testing the New Database	21
Modify MIM System for Multiple Databases	22
Create Scripts to Load Data into Customer Database	22
Client Applications	24
Chapter 2: MIM System Administration (Red Hat Enterprise Linux RHEL 4)	25
MIM Server Basics	25
Architecture	25
Tree Structure	26
Installation	28
MIM Server Permissions	33
Free Pass	33
xmimsvr.acl File	33
Examples	34
MIM Database Write Permissions	35
Permission Algorithms	35

Troubleshooting	36
MIM Server Administration	37
Updating the Database	37
Starting/Stopping the MIM Server	38
System Monitoring	39
MIM Server Backups	41
What to Backup	41
Scheduling	41
Restoring	41
Installing Test Servers	42
Setting Up a Test Server for Data Loading	42
Create a New Database for Customer Data Only	42
Testing the New Database	44
Modify MIM System for Multiple Databases	45
Create Scripts to Load Data into Customer Database	45
Client Applications	47
Chapter 3: Entitlements	49
Introduction	49
Concepts	50
Free Pass	51
Enabling/Disabling Entitlements	52
Usage Guidelines	53
Batch Processing Mode	54
Creating Entitlements Text File	54
Example Entitlements Batch File (entitlements1.txt)	56
Sample BMIM Script to Run (ent1.bmim)	56
Results of Running Sample BMIM Script	57
Sample Entitlements Batch Output File (entitlements1_out.txt)	57
Processor Tag	57
Index	59

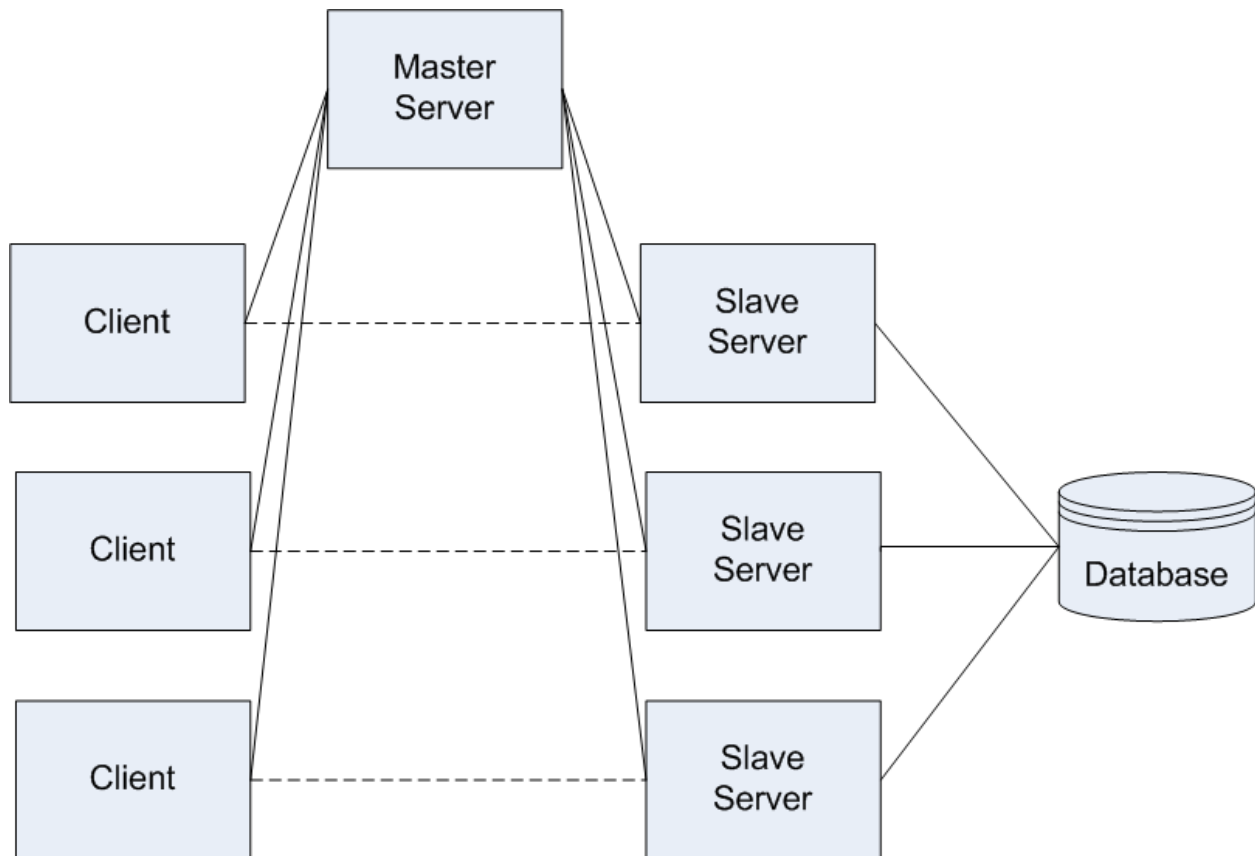
CHAPTER 1

MIM System Administration (Solaris)

MIM Server Basics

Architecture

The following shows the architecture of the MIM server.



The MIM client/server process functions as follows:

- The process is made up of four components – (1) Master Server (2) Slave Servers (3) Clients (4) Database
- The master server is the controlling process and acts like a traffic cop. Requests are made from the clients and the master server divides the workload among the slave servers.
- The slave servers access the database performing the bulk of the work.
- Once the client process has been assigned a slave server, communications is between the client and slave server until the request is complete.
- Clients can be UNIX or PC clients. All clients use the rpc protocol to communicate with the server over the network or localhost. Port numbers are automatically assigned by the port mapper and are above 1024.
- The master server handles all schema requests. If a request is made for data, the Master Server will return the simple hostname and slave server port number to the client. The client will then establish communications with the slave server
- It is very important that the client be able to resolve the simple hostname that the master server returns.
- Each master server has a logical port number that is user defined between 0-255. The default port number is 0.
- Each slave server within a master server has a logical port number beginning with 0 and incrementing up by 1 for each additional slave server.
- There may be multiple Master Servers running on a single machine, each with a unique port number.

Tree Structure

The MIM system must reside in a directory owned by the MIM DBA account, with the top level of the home directory structured as below:

- .limrc** – MIM system environmental variables are set in this file.
- .xmimrc** – Database definitions are read by server at startup.
- .cshrc** – csh startup environmental file
- .profile** – sh startup environmental file
- .netrc** – Used by FTP service for automatic server login.
- .wgetrc** – Resource file for wget process.
- certs** - Authentication certificate for wget.
- comp** – Directory used for dbcompare (for more information on dbcompare, see the "[dbcompare Utility](#)" chapter in the *Data and Development Guide*) when executing database comparisons.
- config** – Directory containing log and lock files for `load_updates.sh` and `cron_updates.sh`.

cron_updates.sh – Script run from cron that controls the downloading of packages and updating the server.

data – Directory containing default database files.

data.cust – Directory containing customer database files loaded from the Excel data loader.

data.gii – Directory containing the Equities database files.

dates – Directory containing date and event files used by the MIM system.

de-jakarta – Web server used by the Java API.

docs – Directory containing system files for creating database catalog of local database.

get_data.ksh – Script used to download packages from LIM.

license – Directory containing license file.

limtest – MIM server test account used by customer for data loading, usually referenced as /home/limtest.

load_updates.sh – Script used to load LIM supplied updates into the database.

mimdbcp – Directory containing system files to copy data from one database to another.

public_jobs – Directory containing saved jobs generated from XMIM.

templates – Directory containing template files that may be modified per installation.

tmp – Directory containing temporary files.

tools – Directory containing utilities such as print server, backups, perl, jre and ncftp.

updates – Directory used for downloading and processing updates.

xmim – Link to xmim40.sol.

xmim40.sol – Directory containing system files (bin, lib, include...).

xmimrc – Directory containing format definitions for viewing data (i.e., f8.2).

xmimsvr.acl – ACL file for controlling access to MIM server – default setting is “@” (all have access).

Selective Directory Tree Structures

certs

thawte.cer - Certification file for wget.

config/

load_updates.lck – Lock file created when load_updates.sh is running. Only one load_updates.sh allowed to run at a time.

load_updates.hst – History of load_updates.sh runs

load_updates.err – Contains error messages from last load_updates.sh run. Removed prior to each run.

cron_updates.lck – Lock file created when update downloads are occurring

data/, data.cust/, data.gii/

xmim.mim – Database schema file. Read into memory when server is started.

xmim.mimdb – Directory containing database files.

xmim.mimdb/daily – Directory containing daily data database files.

xmim.mimdb/tick - Directory containing real ticks (seconds) database files.

xmim.mimdb/intraday - Directory containing intraday data (minute/hourly) database files.

xmim.mimdb/millisecond - Directory containing millisecond data database files.

xmim.mimdb/tabledb – Directory containing relational components such as units of measure, entitlements and corrections.

lock – temporary file used by server.

templates/

.cshrc – default csh startup file for users and MIM DBA account.

.limrc – File containing MIM environmental variables.

.netrc – Example file used by FTP service for automatic server login.

.profile – sh startup environmental file.

.wgetrc – Startup file used by wget, runs FTP using HTTP proxy.

.xmimrc – Database definitions, read by server at startup

cron_updates_post.ksh - A sample script used by the publisher.

get_data.ksh – Customizable script for ftp/wget/uucp to retrieve updates.

sys.dtwmrc – Sample dtwm menu file for Solaris desktops.

xmimlock.acl – Example file for setting database locking permissions.

xmimserver – MIM server startup/shutdown file for rc0.d, rc2.d and init.d.

updates/

filelist – list of files previously downloaded.

logs – directory of log files from each update run.

processed – is not used.

unpacked – temporary holding area for unpacking updates.

temp – used for temp holding area as updates are downloaded.

xmim/

CHANGES – is the binary change log.

X11 – X11 lib definitions for local GUIs.

bin, lib, include – are the system binaries.

demo – contains the demo queries.

library – holds the MIM system macros.

Installation

The MIM server is a client/server system that integrates into existing networks. It requires tcp/ip networking with a Sun server accessible by the clients.

The client applications are as follows:

- The MIM Excel Add-In runs in Excel on the PC and uses the rpc protocol to access the MIM server.
- MIMIC and XMIM are Java applications running on the PC using the rpc protocol to access the MIM server.

Bandwidth for the PC applications is minimal.

In order to setup the MIM server on a network the following information is required:

- Static IP# for the Sun MIM server.
- Netmask for the IP#
- Hostname of the Sun server - The server will use its simple hostname to pass back to the clients to use in rpc calls.
- IP# of default gateway
- Sun server hostname is setup in DNS
- DNS domain name
- DNS server IP#s
- SMTP mail relay IP#
- Postscript network printers - IP#s of units close to user who will be using the XMIM GUI
- FTP capability for the SUN server to FTP out to ftp.lim.com for database updates. This will be a scripted job that can work through proxy servers.
- User names and passwords of XMIM users
- Method setup for remote administration when required.

Installation Procedures

1. Prerequisites:
 - There is a LIM DBA account setup - the installation is simplified if the username is "lim" (use bash, ksh or csh), the group id is "10" and the home directory is /home/lim.
 - The LIM DBA account is at least a 50gb partition.
 - Network is setup and working with DNS and SMTP.
 - User accounts setup for each user that will be using XMIM.
 - A basic understanding of UNIX system commands and administration.
2. To begin the installation, go to the LIM "[Downloads](#)" Web page, then go to the Web page for the latest "[Solaris MIM Server System and Binary](#)". Right-click on the link for the latest download (e.g., Version 4.6.20) and select **Copy Shortcut** from the menu. Paste the shortcut into a command window after the wget command.

```
wget http://customers.lim.com/download/mimserver/rel46.sys
```

3. Enter the following command:

```
gunzip -c rel46.sys|tar xvf -
```

4. Copy the download script and "dot" files from the templates directory:

```
cp -p templates/get_data.ksh  
cp -p templates/.profile .profile  
cp -p templates/.cshrc .cshrc  
cp -p templates/.limrc .limrc  
cp -p templates/.xmimrc .xmimrc
```

5. Modify setup files

Modify `$LIMHOME/.limrc` to match your configuration. This file is used by most scripts to set environments. Set the lines below to the appropriate values. Be sure to leave in the quotes surrounding `CUSTNAME` and use a name that identifies your company.

```
# This is where the MIM system resides (default /home/lim)
LIMHOME=/home/lim
# CUSTNAME will be used in the Subject of email notifications and should be
# descriptive for your Company, System and DB. Please replace Customer Name.
CUSTNAME="Customer Name - `uname -n`$PORT"
```

If `/home/lim` is not the base directory for the LIM applications you will have to modify some files. If you are using `/home/lim`, proceed to step 5.

You will need to replace `"/home/lim"` with the name of the LIM home directory.

The files to modify are:

```
$LIMHOME/.cshrc (if using csh as your shell)
(change setenv LIMHOME /home/lim)

$LIMHOME/jmim/jmim/config/config.sh
(change export XMIMHOME=/home/lim)
```

6. Boot/shutdown rc scripts

Copy the MIM server startup script to the rc directories. You must log in as the user root to perform this task.

```
su - root
cd /etc/init.d
cp -p $LIMHOME/templates/xmimserver .
cd /etc/rc0.d; ln ../init.d/xmimserver K07xmimserver
cd /etc/rc2.d; ln ../init.d/xmimserver S98xmimserver
```

If you are adding more than one server you will need to specify the port number. For example:



```
su - root
cd /etc/init.d
cp -p $LIMHOME/templates/xmimserver xmimserver4200
cd /etc/rc0.d; ln ../init.d/xmimserver4200 K07xmimserver4200
cd /etc/rc2.d; ln ../init.d/xmimserver4200 S98xmimserver4200
```

where `xmimserver4200` is the additional server.

If `/home/lim` is not the base directory or the user "lim" is not the owner of the installation, you will need to modify two lines with the appropriate values:

```
vi /etc/init.d/xmimserver
```

Make changes as needed to these two entries:

```
LIMHOME=/home/lim
LIMUSER=lim
```

Exit from root user:

```
logout
```

Exit from the lim user, then log back in to set new environment:

```
logout su - lim
```

7. Startup the server

The application will require a license in order to run. Please obtain the `hostid` and `hostname` of the system by typing:

```
hostid  
hostname
```

The License Key can be obtained by calling Customer Support at 1-800-LIM-XMIM (1-800-546-9646) +512-346-1618 or email support@lim.com and specifying the displayed `hostid` and `hostname`. The license will be emailed, faxed or verbally transmitted. To install the license, issue the following command and follow the instructions.

From the `$LIMHOME` directory type:

```
add_id
```

Once the license is installed, startup the MIM server. The server will start automatically when the system boots and shutdown when the system is halted.

```
start.server
```



To check the licenses that were installed, you may run the `list_ids` command-line application on the server. `list_ids` will return a list of the licenses that are available for the license file specified in the “`xmimrc`” file for the current user. To run `list_ids` enter:

```
list_ids
```

8. Setup and schedule the update process.

This update process will:

- On error, send mail to the user specified in the `.limrc` file.
- On a successful update, send mail to the user specified in the `.limrc` file.
- If there are no update files waiting to be run, it will run but not produce any output or email.

Modify the file `get_data.ksh`:

```
vi get_data.ksh
```

Comment/uncomment the lines for the appropriate update method: `ftp` or `wget`:

```
doftp  
# dowget
```

If this is an FTP account, create a `.netrc` file for accessing the LIM FTP server. You will need to obtain the login and password from LIM Customer Support replacing the XXXs with the appropriate values.

```
echo "machine ftp.lim.com login F8XXX password XXXXXXXX" > .netrc
chmod 600 .netrc
```

Test the FTP access. If the access fails, contact LIM Customer Support for help in troubleshooting. The most likely cause will be a proxy server that will require special scripting or network access problems.

```
ftp ftp.lim.com
```

If the FTP access works correctly you will be logged into your FTP account on the LIM FTP server. Quit FTP.

This script that runs the update process `cron_updates.sh` produces no output when there are no updates to pickup. The script should be scheduled to run once an hour.

Invoke the cron editor:

```
crontab -e
```

Add the following entry:

```
48 * * * * /home/lim/cron_updates.sh
```



`/home/lim` should be the value that has been set to LIMHOME.

9. Setup user accounts for each user that will be using the XMIM GUI.

For each user account that has been setup, copy the `$LIMHOME/templates/.cshrc` file to the user's home directory and replace the value of LIMHOME as needed. Create a crontab entry for each user by adding the following:

```
17 * * * * /home/lim/jmim/jmim/bin/sol/job_cron_trigger.sh
```



`/home/lim` should be the value that has been set to LIMHOME.

By default, date outputs are in US format. To change date outputs to UK format do the following for each user's profile:

Set the environmental variable `XMIM_LOCALE` to `EU`.

10. Setup printer access for XMIM users using HP Jetadmin (native PC applications will use printers defined on PC).

Install jetadmin from `$LIMHOME/tools/sol/sold615.pkg`



See `$LIMHOME/tools/sol/sold615.pkg,README` for installation instructions.

Setup default system printer and add entry to each users `.cshrc` to define default printer for user by setting `LPDEST` to desired printer name if it is not the system default.

MIM Server Permissions

There are four levels of security in the MIM system: server, application, database and data. The server level security was implemented in MIM server version 4.2. Previously no checks were made. The security check at the server connection point ensures that a user has proper authority to connect to MIM servers.

Free Pass

Connection to the MIM server is always allowed for the MIM server owner when connecting from the MIM server.

xmimsvr.acl File

The security check is controlled by the xmimsvr.acl file on the server. The client username, hostname and domain are compared to the list of authorized MIM users in the xmimsvr.acl file.

The xmimsvr.acl file is located on the server at:

For Unix:

```
$LIMHOME/xmimsvr.acl
```

or for PC:

```
(/home/lim/xmimsvr.acl)
```

There is one xmimsvr.acl file per server and each xmimsvr.acl file resides in the server's home directory.

Global Setting

The xmimsvr.acl file global setting is @ which allows anyone to have access to the server. The system administrator can edit the xmimsvr.acl file and change who has access to specific domains and server. If the xmimsvr.acl file does not exist then no one will have access to the server.

File Format

Users that need lock permissions must be added to a file called xmimsvr.acl where "acl" stands for access control list.

The syntax for an entry in the xmimsvr.acl file is:

```
[username]@[hostname | ipaddr[/netmask]]
```

where

- arguments on either side of the @ sign are optional including `username`, `hostname`, `ipaddr` and `netmask`.
- IP is in the format: `x.x.x.x[/n]`
- `"/n"` is the netmask. Without a netmask, an IP refers to a specific host. With the netmask, an IP can refer to a network.
- `hostname` might include a domain name or might be only a domain name. If only a domain name, there must be a `"."` in the name somewhere. An example would be `"@.com"`.
- the `username`, `hostname` and domain names are not case sensitive.

Examples

The following shows example entries for the `xmimsvr.acl` file:

```
lisa@matrix
lisa@matrix.lim.com
LiSA@maTrix.LIM.com
@matrix
@matrix.lim.com
@lim.com
@.com
lisa@12.43.227.00/24
lisa@12.43.227.45
lisa
@
```

MIM Database Write Permissions

In order to write data to the database, the database must first be locked. The system administrator can setup permissions so that individual users or groups can either have access to locking and writing to the database or be prevented from locking and writing to the database.

The system's administrator must set permissions on the file `xmim.mim` to determine who has the ability to lock and write to the database. The `xmim.mim` file contains the schema information for a MIM database. The file `.xmimrc` controls where the `xmim.mim` file(s) is located. The `.xmimrc` file is typically located in the user's home directory:

```
/home/lim/.xmimrc
```

The following shows an example entry for a `xmim.mim` file in the `.xmimrc` file:

```
database: /home/lim/data/xmim.mim
```

Permission Algorithms

Setting Permissions for No One

If Unix write permissions for the database file (`xmim.mim`) are not turned on for the database owner, no one can write to the database because of normal Unix restrictions.

Setting Permissions for Everyone

If Unix write permissions on the database file (`xmim.mim`) is turned on for "others", everyone can lock the database.

When the Server Owner is running Locally on a Server (Free Pass)

In this scenario, the server owner is working on a system where the server and client applications are on the same system. In this case, the server owner can lock the database.

Setting Permissions for Individuals

For all other circumstances, users that need lock database permissions must be added to a new file(s) called `xmimlock.acl`. The `xmimlock.acl` file(s) must reside in the same directory as the database schema (`xmim.mim`) file(s). There can be multiple databases, so for each `xmim.mim` schema file designating a database, there can be a `xmimlock.acl` file.

The syntax for an entry in the `xmimlock.acl` file is:

```
[username]@[hostname|ipaddr[/netmask]]
```

where

- arguments on either side of the `@` sign are optional including `username`, `hostname`, `ipaddr` and `netmask`.
- IP is in the format: `x.x.x.x[/n]`
- `"/n"` is the netmask. Without a netmask, an IP refers to a specific host. With the netmask, an IP can refer to a network.
- `hostname` might include a domain name or might be only a domain name. If only a domain name, there must be a `"."` in the name somewhere. An example would be `"@.com"`.
- the `username`, `hostname` and domain names are not case sensitive.
- The following shows some example entries in the `xmimlock.acl` file:

```
lisa@matrix  
lisa@12.43.227.00/24  
lisa@12.43.227.45  
geoffrey  
lim  
@
```

Note: After modifying the `"xmimlock.acl"` file, the MIM server does not need to be restarted in order for the changes to take affect.

Troubleshooting

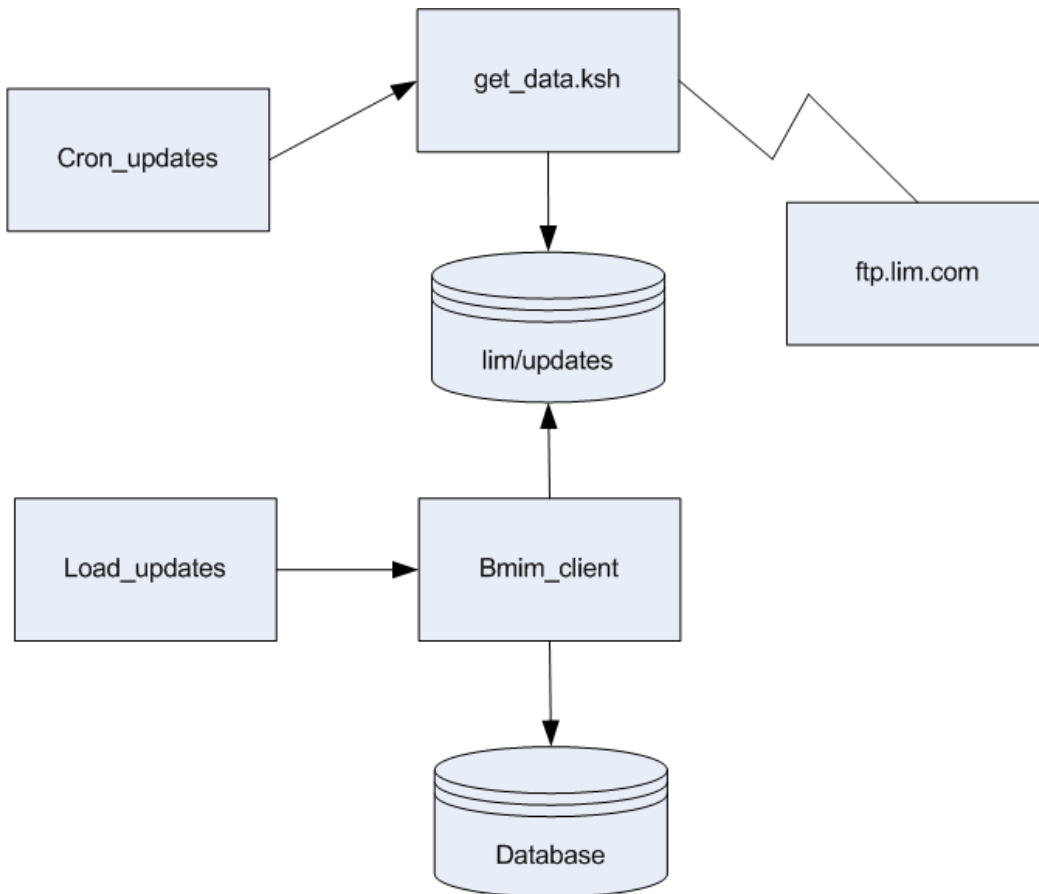
If you are still having trouble locking a database, take note of the following:

1. `lock_files` locks **all** databases in the current list of databases, so you may need more than one `xmimlock.acl` file even though you only wish to write to one database.
2. Nobody can write to a database if the database owner does not have Unix write permissions to the database schema file, `xmim.mim`. To set the Unix permissions for the owner to read/write and the group and others to read only, enter: `chmod 644 xmim.mim` the permissions will appear as: `-rw-r--r-- xmim.mim`

MIM Server Administration

Updating the Database

The following shows the MIM database update process:



The MIM database update process functions as follows:

- `Cron_updates.sh` is setup as a batch job in the MIM DBA user's crontab and is the controlling process for the update procedure.
- `Cron_updates.sh` will execute the `get_data.ksh` script to retrieve the update files from the LIM FTP server and deposit them in the LIM update holding area (`/home/lim/updates`).
- `Cron_updates.sh` will start up the `load_updates.sh` script, which is the LIM database update routine.
- `Load_updates.sh` will use `bmim_client` to access and update the LIM database.
- Upon completion, `cron_updates.sh` will notify the designated system administrator about the success of the update via email.

Starting/Stopping the MIM Server

All actions listed below should be run while logged into the MIM DBA account. These scripts have the MIM environmental variables set for `LIMHOME`, `PORT` and `SERVERHOST`. They will act only on the MIM server for this account.

This command stops the server:

```
server.info -k
```

This command starts the server:

```
start.server
```

To verify the status of the server execute:

```
server.info [-h] [-p server_number] [-k] [-r]
```

where (Optional entries are designated with brackets []):

<code>[-h]</code>	Help/Usage. Print this message and exit.
<code>[-p server_number]</code>	Server is at the port number <i>server_number</i> .
<code>[-k]</code>	Kill the server.
<code>[-r]</code>	Resynchronize the server with the <code>xmimrc</code> file.

By default the status will be returned for the server port defined in the `.limrc`. If the server is not running a message will be returned immediately stating `Cannot connect to server`.

Example output:

```
master server 0 on fourx is running
  process id:      23411
  owner:          lim
  log_level:      0
  log_file:       /home/lim/tmp/.xmim_server_0.log
  database 0:     /home/lim/data/xmim.mim
  license:        /home/lim/license/xmim.ids
  version:        Ver 4.1.16 Compiled Mon Oct 8 17:34:50 CDT 2001
  there are 2 slave servers registered
  slave server 0:
    host name:    fourx
    process id:   26009
    owner:        lim
    status:       running
    last dispatched: Wed Oct 17 05:35:06 2001
  slave server 1:
    host name:    fourx
    process id:   n/a
    owner:        n/a
    status:       not running
    last dispatched: n/a
```

System Monitoring

Update monitoring is aided by email notifications sent out when the MIM server runs the update process. There are four types of email notifications that are sent out to the email addresses defined in the `.limrc` file.

1. SUCCESSFUL – no errors encountered, update process complete
2. FAILURE – Errors encountered, update process was not able to continue. Errors must be located and corrected for normal database operation.
3. NOTICE – This message is a notification that a scheduled update process could not execute because another update process is already running, Only one update process at a time may run, no action is required.
4. WARNING (check log messages) – An error was encountered during the update process, but the update was able to complete. These errors are not critical and usually pertain to an individual series that did not update correctly. The error should be located and action taken to correct it.

Update logs are stored in `$LIMHOME/updates/logs`. These log files should be examined when errors occur to identify the error. If you need help in determining the error send the error message to support@lim.com for further assistance.


Disk monitoring is performed each time `cron_updates.sh` runs. Part of the output is a `df` of the MIM DBA partition. Do not let the system run out of disk space as this can corrupt the database.

Server logging is enabled by default at level 0 (lowest level). Log files are stored in `$LIMHOME/tmp` and can contain information about who is accessing the database, what is being accessed and types of access. Log files are stored in the common log format utilized by Web servers.

The `xmim_svr_info` program provides information about the server and the ability to stop the server.

```
xmim_svr_info [-h] [-s host] [-p server_number] [-k] [-r] [-n]
```

where (Optional entries are designated with brackets []):

[-h]	Help/Usage. Print this message and exit.
[-s <i>host</i>]	Specifies the host that the master server is running on. The default will be the local host.
[-p <i>server_number</i>]	Specifies the <i>server_number</i> of the desired master server. If this argument as well as the -k and -r options are not specified, information will be returned for every master server that is running on the given host. The default server number if the -k or -r options are used will be 0.
[-k]	Specifies to kill a given master server.
 Note that the user must be either the owner of the server process or the super-user to kill a server. The default will be to kill master server 0 on the local host.	
[-r]	Specifies to re-read the <code>.xmimrc</code> file which effectively causes the databases and any libraries/macros to be re-loaded. This will be effected only for one particular master server, the default being server 0 on the local host.
[-n]	Specifies to open a new log file with the standard log filename and begin logging to this new file. Simply renaming a log file will not result in logging to a new file because the file descriptor will still point to that file. Thus, to achieve archiving of log files, the old log file must be renamed and then the -n flag used to open a new log file, taking out a new file descriptor for it.

Printing

Printing is accomplished by one of two methods. If the user is using a native PC application then printing is defined on the user's local PC using standard Windows printing. If the user is using an X-application then printing is achieved through the UNIX machine. To accomplish this a tcp/ip networked postscript printer with a static IP number is required.

A copy of JETADMIN from HP is provided in `$LIMHOME/tools/sol` as a standard Solaris `pkgadd`. This program will be installed in `/opt/hpnp` and may be executed by running `/opt/hpnp/jetadmin`. Network printers may be added and setup as the default system printer. Users can set their default printer by setting the `LPDEST` environmental variable to the desired printer.

MIM Server Backups

What to Backup

It is important to backup the system on a regular basis. User accounts and the MIM database should be backed up on a reliable media. The database is comprised of three directories: data, dates and xmimrc. The database varies in size from 5gbs to 200gbs depending on the amount of data purchased and the amount of data loaded by the user. Most databases will be around 5-7gbs. Loading tick data or real-time ticks will generate larger database sizes and receiving the TAQ database from LIM requires about 200gbs of disk space.

Scheduling

The database should be backed up once a week on Sunday mornings when activity is the least. Databases should not be backed up while updating and should always be done as a full backup.

Restoring

If a database is to be restored, notify LIM (support@lim.com) and they will queue up the necessary updates to bring the database up to date.

If you do not have a backup system or method, LIM can provide a tape drive and scripts to backup the system. These scripts are located in:

```
$LIMHOME/tools/sol/dumpsys
```

Installing Test Servers

The following steps are required to setup a test database for loading customer data:

1. Create a new database and system for customer data only and read access to the LIM production database
2. Modify the MIM system for multiple databases
3. Create scripts to load data into the customer database

A few assumptions:

- The new test database will be running on port 99.
- The MIM DBA user name is `limtest`.
- The MIM DBA home directory will be `/home/limtest`.
- All remaining steps are carried out logged in as the user `limtest`.

Setting Up a Test Server for Data Loading

When loading non-LIM supplied data (customer data) it is a good idea to take advantage of the 4x multiple database technology and to always load the data first into a test database. There will be two databases used. One database will contain the data LIM ships and the other will be the customer data. The `.xmimrc` file will need modifying and some scripts created to define the new database and access to the production database. This will make life easier in that if something happens to the LIM database it can be replaced without effecting the data the customer has loaded and vice versa. It also makes it easy to install the production database by simply copying the new customer database and modifying the `.xmimrc` file in production.

Once complete both databases are accessed as if they were one. When loading customer data a new command called `database_narrow` is used in order to specify which database data is being loaded into.

Create a New Database for Customer Data Only

First create a new server instance running on port 99 owned by the user `limtest` in the `/home/limtest` directory as described in “[Installation](#)” earlier in this document. Obtain the system release from the LIM “[Download](#)” page or from LIM’s FTP site: `ftp.lim.com` and login as anonymous with your email address as the password. Extract the system into the `/home/limtest` directory:

Next, login as su or login as user `limtest`:

```
su - limtest
gunzip -c rel4x.sys | tar xvf -
```

Copy the setup files from the templates directory as described below:

Copy the MIM startup scripts to the bin directory, \$LIMHOME represents /home/limtest in the instructions below:

```
cp -p templates/start.xmim $LIMHOME/xmim/bin
cp -p templates/start.server $LIMHOME/xmim/bin
cp -p templates/start.bmim $LIMHOME/xmim/bin
cp -p templates/server.info $LIMHOME/xmim/bin
```

Copy the load_updates.cfg file to the config directory

```
cp -p templates/load_updates.cfg config
```

Copy the dot files to the limtest home directory:

```
cp -p templates/.cshrc .cshrc
cp -p templates/.limrc .limrc
cp -p templates/.xmimrc .xmimrc
```

Modify \$LIMHOME/.limrc to match your configuration. This file is used by several scripts to set environments. Set the two lines below to the appropriate values.

```
LIMHOME=/home/limtest
PORT="-p 99"
```

You will need to find LIMHOME and replace /home/lim with /home/limtest.

The files to modify are:

```
$LIMHOME/.xmimrc
$LIMHOME/.cshrc
In load_updates.cfg, replace "USERNAME lim" with "USERNAME limtest", replace "UPDATEDIR /home/lim/updates" with "UPDATEDIR /home/limtest/updates", replace "SERVERPORT 0" with "SERVERPORT 99".
```

```
$LIMHOME/config/load_updates.cfg
```

Copy the MIM server startup script to the rc directories. You must log in as the user root to perform this task.

```
su - root
cd /etc/rc2.d
cp -p $LIMHOME/templates/S98xmimserver S98xmimserver99
cd /etc/rc0.d; ln ../rc2.d/S98xmimserver99 K07xmimserver99
cd /etc/init.d; ln ../rc2.d/S98xmimserver99 xmimserver99
vi /etc/init.d/xmimserver99
```

Make changes to these two entries:

```
LIMHOME=/home/limtest
LIMUSER=limtest
```

Save changes and logout of root.

Create a link to the existing license file on the production server:

```
cd /home/limtest/license
ln -s /home/lim/license/xmim.ids .
```

Next change directory to mimdbcp and create two scripts:

```
cd /home/limtest/mimdbcp
touch makedb
```

```
touch make_cust
chmod 775 makedb
chmod 775 make_cust
```

Edit `makedb` and put the following lines in it:

```
rm -r ../data
mkdir ../data ../data/xmim.mimdb
cd ../data/xmim.mimdb
mkdir daily intraday tick
touch daily/cust.fac intraday/cust.tck
touch tick/cust.tck
cd ../../mimdbcp
start.server
bmim_client -p 99 make_cust
```

Edit `make_cust` and put the following lines in it (make sure the paths match with your actual paths). The name `Cust_Data` that is used throughout these examples could be the name of the customer's company. (It is an arbitrary name.)

```
lock_files
relation_add {
  name = Cust_Data;
  parent = TopRelation;
  type = category;
}
database_def_daily_file TopRelation:Cust_Data daily/cust.fac
database_def_intraday_file TopRelation:Cust_Data intraday/cust.tck
database_def_tick_file TopRelation:Cust_Data tick/cust.tck

relation_delete EconomicIndicators;
relation_delete Equities;
relation_delete Futures;
relation_delete Indices;
relation_delete MonetaryIndicators;
relation_delete Other;

unlock_files;
```

Next execute `makedb`, this will create an empty 4x DB designed for customer data and may be run at any time the database needs to be rebuilt and emptied out. This script will remove the data directory. You must save this directory under a different name if you wish to have it preserved.

Testing the New Database

You can test the database by trying a `print-relation` using the `bmim_client` command.

```
bmim_client -p 99
print_relation {relation = TopRelation; verbose;}
```



Use the `-p` option to tell BMIM to access Port 99

Press **Ctrl+d** to exit.

You should get a printout that looks similar to this:

```
Category:      TopRelation (Root of Relation Tree)
Category:      TopRelation:Cust_Data (Cust_data)
```

Where `cust_data` is the name that you named your test data.



When loading data to the new database, make sure you are logged in as user `limtest` and loading data using port 99 (`bmim_client -p 99`).

Modify MIM System for Multiple Databases

Now rename the databases, edit the `.xmimrc` file and start up the server in order to access the two databases as one. The database name “`data.company`” should be renamed to match your company name.

First rename the databases:

```
cd /home/limtest
mv data data.company
```

Next edit the `.xmimrc` file.

Change the following line to point to the production database. This will be a read-only instance:

```
echo "database:  $LIMHOME/data/xmim.mim"
to
echo "database: /home/lim/data/xmim.mim"
Insert this line above the production database entry:
echo "database:$LIMHOME/data.company/xmim.mim"
```

Example:

```
echo "database:$LIMHOME/data.company/xmim.mim"
echo "database: /home/lim/data/xmim.mim"
```

Save and exit.

Startup the server:

```
start.server
```

Now both the LIM and the customer databases are running under one 4x server.

Create Scripts to Load Data into Customer Database

In order to update the customer database it is very important that the following rules are followed:

- When loading data into the customer database the `database_narrow` command must be used at the beginning of each script before the database is locked.

- `database_narrow {database=~ /data.company/xmim.mim;}`
- Loading must be done as the user who owns the database.
- The home directory of the database owner is the top of the 4x tree.
- Category names should be mixed case.
- Relation names should be upper case.
- Column names are case independent.
- These rules will ensure that the data is correctly loaded into the proper database.

Implementing Data Loading into Production

Once the customer database is built and tested it is ready for production. To put it into production the `data.company` tree needs to be copied to the production directory and the production `.xmimrc` file needs to be modified to reflect the addition of another database. Scripts and procedures that are used to update the customer database also need to be copied to production and all ownerships set to the production MIM DBA owner.

Client Applications

There are four client applications to install (excluding the APIs). See the document [Installing LIM Client Applications](#) for instructions on setting up the XMIM, MIMIC and the Excel Add-in applications.



Jobs, previously a stand-alone product, is now installed as part of the XMIM installation.

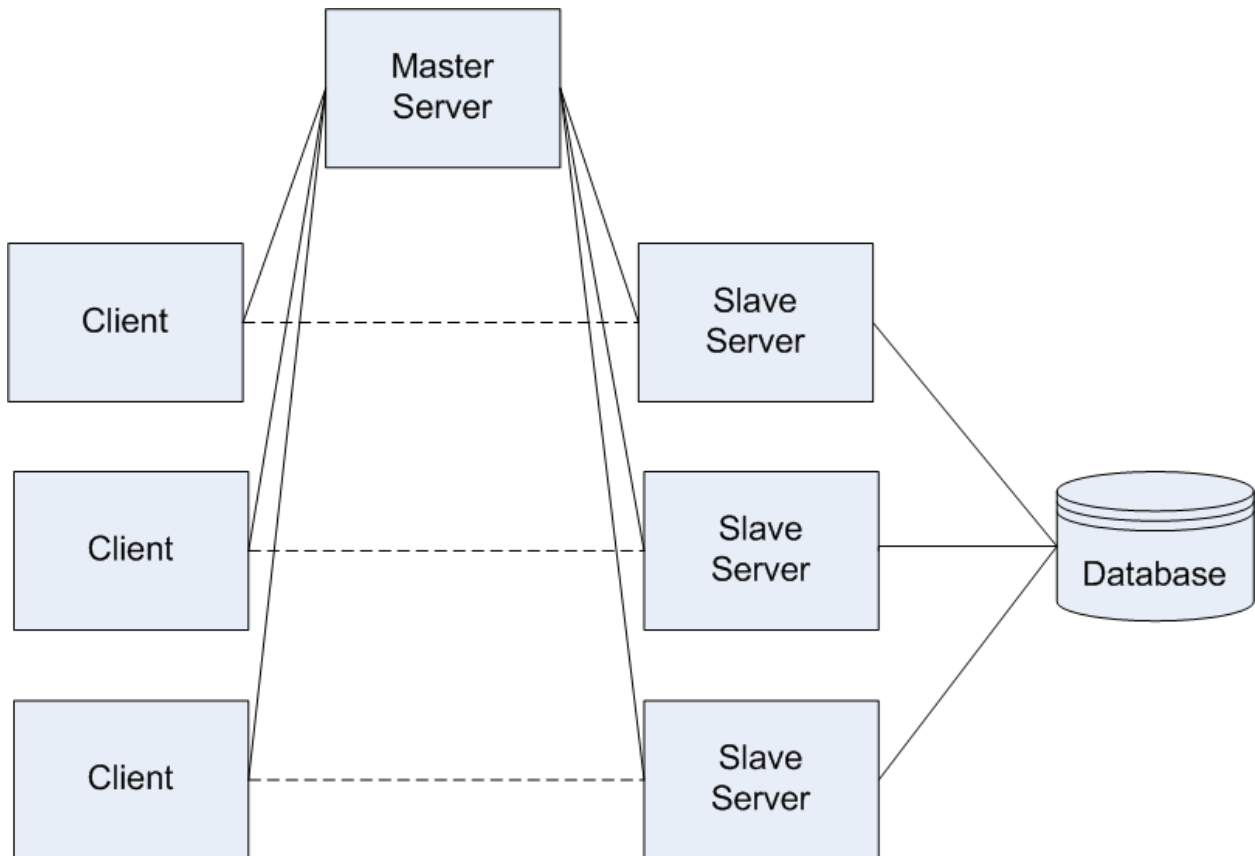
CHAPTER 2

MIM System Administration (Red Hat Enterprise Linux RHEL 4)

MIM Server Basics

Architecture

The following shows the architecture of the MIM server.



The MIM client/server process functions as follows:

- The process is made up of four components – (1) Master Server (2) Slave Servers (3) Clients (4) Database
- The master server is the controlling process and acts like a traffic cop. Requests are made from the clients and the master server divides the workload among the slave servers.
- The slave servers access the database performing the bulk of the work.
- Once the client process has been assigned a slave server, communications is between the client and slave server until the request is complete.
- The master server handles all schema requests. If a request is made for data, the master server will return the simple hostname and slave server port number to the client. The client will then establish communications with the slave server.
- It is very important that the client be able to resolve the simple hostname that the master server returns.
- All clients use the rpc protocol to communicate with the server over the network or localhost. Port numbers are either explicit or automatically assigned by the port mapper and are above 1024.

When using the port mapper, ports are in the range of 0-128 and have the following rules:

- Each master server has a logical port number that is user defined between 0-255. The default port number is 0.
- Each slave server within a master server has a logical port number beginning with 0 and incrementing up by 1 for each additional slave server.

Explicit ports will fall in the range 1056-127968 for every multiple of 32:

- The TCP port is mapped back to a RPC program number: $(\text{tcp port} - 1024) / 32 + 128$. The ports are used in 32 port blocks.
- There may be multiple master servers running on a single machine, each with a unique port number.

Tree Structure

The MIM system must reside in a directory owned by the MIM DBA account, with the top level of the home directory structured as below:

- .limrc** – MIM system environmental variables are set in this file.
- .xmimrc** – Database definitions, read by server at startup.
- .profile** – sh startup environmental file.
- .netrc** – Used by FTP service for automatic server login.
- .wgetrc** – Resource file for wget process.
- certs** - Authentication certificate for wget.
- config** – Directory containing log and lock files for load_updates.sh.

cron_updates.sh – Script run from cron that controls the downloading of packages and updating the server.

data – Directory containing default database files.

data.cust – Directory containing customer database files loaded from the Excel data loader.

data.gii – Directory containing the Equities database files.

dates – Directory containing date and event files used by the MIM system.

de-jakarta – Web server used by the Java API.

get_data.ksh – Script used to download packages from LIM.

license – Directory containing license file.

limtest – MIM server test account used by customer for data loading, usually referenced as /home/limtest.

j2se – Link to current version of Java Software Developers Tool Kit.

load_updates.sh – Script used to load LIM supplied updates into the database.

mimdbcp - Directory containing system files to copy data from one database to another.

public_jobs - Directory for customer created cron jobs.

templates – Directory containing template files that may be modified per installation.

tmp – Directory containing temporary files.

updates – Directory used for downloading and processing updates.

xmim – Link to xmim4.linux.

xmim4.linux – Directory containing system files (bin, lib, include...).

xmimrc – Directory containing format definitions for viewing data (i.e., f8.2)

xmimsvr.acl – ACL file for controlling access to MIM server – default setting is “@” (all have access).

Selective Directory Tree Structures

certs

thawte.cer - Certification file for wget.

config/

load_updates.lck – Lock file created when load_updates.sh is running. Only one load_updates.sh allowed to run at a time.

load_updates.hst – History of load_updates.sh runs.

load_updates.err – Contains error messages from last load_updates.sh run. Removed prior to each run.

cron_updates.lck – Lock file created when update downloads are occurring.

data/, data.cust/, data.gii/

xmim.mim – Database schema file. Read into memory when server is started.

xmim.mimdb – Directory containing database files.

xmim.mimdb/daily – Directory containing daily data database files.

xmim.mimdb/tick - Directory containing real ticks (seconds) database files.

xmim.mimdb/intraday - Directory containing intraday data (minute/hourly) database files.

xmim.mimdb/millisecond - Directory containing millisecond data database files.

xmim.mimdb/tabledb – Directory containing relational components such as units of measure, entitlements and corrections.

lock – Temporary file used by server.

mimdbcp

- README.guile** - Necessary file for mimdbcp.
- mimdbcp** - Command to run the mimdbcp utility.
- mimdbcp.cfg** - Configuration file for mimdbcp.

templates/

- .limrc** – File containing MIM environmental variables.
- .netrc** – Example file used by FTP service for automatic server login.
- .profile** - sh startup environmental file.
- .wgetrc** – Startup file used by wget, runs FTP using HTTP proxy.
- .xmimrc** – Database definitions, read by server at startup.
- cron_updates_post.ksh** - A sample script used by the publisher.
- get_data.ksh** – Customizable script for ftp/wget/uucp to retrieve updates.
- ksh.2.4.20** – AT&T ksh binary for kernel versions 2.4.20 and above.
- ksh.2.4.9** – AT&T ksh binary for kernel versions below 2.4.20.
- xmimserver** – MIM server startup/shutdown file for rc0.d, rc3.d and init.d.

updates/

- filelist** – List of files previously downloaded.
- logs** – Directory of log files from each update run.
- processed** – Not used.
- unpacked** – Temporary holding area for unpacking updates.
- temp** – Used for temp holding area as updates are downloaded.

xmim/

- CHANGES** – Binary change log.
- bin, lib, include** – System binaries.
- demo** – Demo queries.
- library** – Contains MIM system macros.

Installation

The MIM server is a client/server system that integrates into existing networks. It requires tcp/ip networking with a Linux server accessible by the clients.

- The Excel Add-In runs in Excel on the PC and uses the rpc protocol to access the MIM server.
- MIMIC is a Java application running on the PC using the rpc protocol to access the MIM server.
- API applications run on either the server or client systems.

Bandwidth required for the PC applications is minimal.

In order to setup the MIM server on a network the following information is required:

- Static IP# for the Linux MIM server.

- Netmask for the IP#
- Hostname of the Linux server - The server will use its simple hostname to pass back to the clients to use in rpc calls.
- IP# of default gateway
- Linux server hostname is setup in DNS
- DNS Domain Name
- DNS Server IP#s
- SMTP mail relay IP#
- FTP or HTTPS capability for the Linux server to download database updates from ftp.lim.com. This will be a scripted job that can work through proxy servers.
- Method setup for remote administration when required.

Installation Procedures

1. Prerequisites:
 - There is a LIM DBA account setup - the installation is simplified if the username is "lim" (use bash or ksh), the group id is "100" and the home directory is /home/lim.
 - The LIM DBA account is at least a 50gb partition.
 - Network is setup and working with DNS and SMTP.
 - A basic understanding of UNIX system commands and administration.
 - Red Hat Enterprise Linux RHEL 4
2. To begin the installation, go to the LIM "[Downloads](#)" Web page, then go to the Web page for the latest "[Linux MIM Server System and Binary](#)". Right-click on the link for the latest download (e.g., Version 4.6.20) and select **Copy Shortcut** from the menu. Paste the shortcut into a command window after the wget command.

For example:

```
wget http://customers.lim.com/download/mimserver_linux/rel46.sys
```

3. To extract the file, enter the following command:
4. Copy the download script and "dot" files from the templates directory:

```
gunzip -c rel46.sys|tar xvf -  
  
cp -p templates/get_data.ksh .  
cp -p templates/.profile .profile  
cp -p templates/.limrc .limrc  
cp -p templates/.xmimrc .xmimrc  
cp -p templates/.xmimrc .netrc
```

5. Modify the setup files:

Modify "\$LIMHOME/.limrc" to match your configuration. This file is used by most scripts to set environments. Set the lines below to the appropriate values. Be sure to leave in the quotes surrounding CUSTNAME and use a name that identifies your company.

```
# This is where the MIM system resides (default /home/lim)  
LIMHOME=/home/lim
```

```
# CUSTNAME will be used in the Subject of email notifications and should be
# descriptive for your Company, System and DB. Please replace Customer Name.
CUSTNAME="Customer Name - `uname -n`$PORT"
```

6. Boot/shutdown rc scripts and the KSH binary:

Copy the MIM server startup script to the rc directories. You must log in as the user root to perform this task. Replace the \$LIMHOME with the value as defined in Step 5.

```
su - root
cd /etc/init.d
cp -p $LIMHOME/templates/xmimserver .
```



If multiple MIM servers are being setup you will need to specify the xmimserver port number for the additional server. For example, to add another server:

```
cp -p $LIMHOME/templates/xmimserver xmimserver4200
```

where 4200 is an example port number.

If "/home/lim" is not the base directory or the user "lim" is not the owner of the installation, you will need to modify two lines with the appropriate values:

```
vi /etc/init.d/xmimserver
```

Make changes as needed to these two entries:

```
LIMHOME=/home/lim
LIMUSER=lim
```

Add the xmimserver to the startup system:

```
chkconfig --add xmimserver
```



If you are adding more than one server you will need to specify the port number. For example:

```
chkconfig --add xmimserver xmimserver4200
```

where xmimserver4200 is the additional server.

Copy the KSH binary to /bin/ksh. This shell is required to run scripts and update processes and must be a valid AT&T version for your kernel.



There are two versions in \$LIMHOME/templates/. One is for recent kernels – ksh.2.4.20 (i.e., Red Hat Enterprise RHEL 4), the other is for the older kernels – ksh.2.4.9 (i.e., RedHat 7.2).

```
cp -p $LIMHOME/templates/ksh.2.4.20 /bin/ksh
```

Exit from root user:

```
Logout
```

Exit from the lim user, then log back in to set new environment:

```
logout su - lim
```

7. Startup the server:

The application will require a license in order to run. Please obtain the `hostid` and `hostname` of the system by typing:

```
hostid hostname
```

The License key can be obtained by calling Customer Support at 1-800-LIM-XMIM (1-800-546-9646) +512-346-1618 or email support@lim.com and specifying the displayed `hostid` and `hostname`. The license will be emailed, faxed or verbally transmitted. To install the license, issue the following command and follow the instructions.

From the `$LIMHOME` directory type:

```
add_id
```

Once the license is installed, startup the MIM server. The server will start automatically when the system boots and shutdown when the system is halted.

```
start.server
```



To check the licenses that were installed, you may run the `list_ids` command-line application on the server. The `list_ids` command will return a list of the licenses that are available for the license file specified in the `xmimrc` file for the current user. To run `list_ids` enter:

```
list_ids
```

8. Setup and schedule the update process. This update process will:

- On error, send mail to the user specified in the `.limrc` file.
- On a successful update, send mail to the user specified in the `.limrc` file.
- If there are no update files waiting to be run, it will run but not produce any output or email.

Modify the file `get_data.ksh`:

```
vi get_data.ksh
```

Comment/uncomment the lines for the appropriate update method: `ftp`, `wget`, or `https`.

```
#doftp  
#dowget  
dohttp
```

Create a `.netrc` file for accessing LIM updates. You will need to obtain the login and password from LIM Customer Support replacing the XXXs with the appropriate values.

```
echo "machine ftp.lim.com login F8XXX password XXXXXXXX" > .netrc chmod 600 .netrc
```

This script that runs the update process `cron_updates.sh` produces no output when there are no updates to pickup. The script should be scheduled to run once an hour.

Invoke the cron editor:

```
crontab -e
```

Add the following entry:

```
48 * * * * /home/lim/cron_updates.sh
```



`/home/lim` should be the value that has been set to `LIMHOME`.

MIM Server Permissions

There are four levels of security in the MIM system: server, application, database and data. The server level security was implemented in MIM server version 4.2. Previously no checks were made. The security check at the server connection point ensures that a user has proper authority to connect to MIM servers.

Free Pass

Connection to the MIM server is always allowed for the MIM server owner when connecting from the MIM server.

xmimsvr.acl File

The security check is controlled by the `xmimsvr.acl` file on the server. The client username, hostname and domain are compared to the list of authorized MIM users in the `xmimsvr.acl` file.

The `xmimsvr.acl` file is located on the server at:

For Unix:

```
$LIMHOME/xmimsvr.acl
```

or for PC: `/home/lim/xmimsvr.acl`



There is one `xmimsvr.acl` file per server and each `xmimsvr.acl` file resides in the server's home directory.

Global Setting

The `xmimsvr.acl` file global setting is `@` which allows anyone to have access to the server. The system administrator can edit the `xmimsvr.acl` file and change who has access to specific domains and server. If the `xmimsvr.acl` file does not exist then no one will have access to the server.

File Format

Users that need lock permissions must be added to a file called `xmimsvr.acl` where “acl” stands for access control list.

The syntax for an entry in the `xmimsvr.acl` file is:

```
[username]@[hostname | ipaddr[/netmask]]
```

where

- arguments on either side of the @ sign are optional including `username`, `hostname`, `ipaddr` and `netmask`.
- IP is in the format: `x.x.x.x[/n]`
- “/n” is the `netmask`. Without a `netmask`, an IP refers to a specific host. With the `netmask`, an IP can refer to a network.
- `hostname` might include a domain name or might be only a domain name. If only a domain name, there must be a “.” in the name somewhere. An example would be “@.com”.
- the `username`, `hostname` and domain names are not case sensitive.

Examples

The following shows example entries for the `xmimsvr.acl` file:

```
lisa@matrix  
lisa@matrix.lim.com  
LiSA@maTrix.LIM.com  
@matrix  
@matrix.lim.com  
@lim.com  
@.com  
lisa@12.43.227.00/24  
lisa@12.43.227.45  
lisa  
@
```

MIM Database Write Permissions

In order to write data to the database, the database must first be locked. The system administrator can setup permissions so that individual users or groups can either have access to locking and writing to the database or be prevented from locking and writing to the database.

The system's administrator must set permissions on the file `xmim.mim` to determine who has the ability to lock and write to the database. The `xmim.mim` file contains the schema information for a MIM database. The file `.xmimrc` controls where the `xmim.mim` file(s) is located. The `.xmimrc` file is typically located in the user's home directory:

```
/home/lim/.xmimrc
```

The following shows an example entry for a `xmim.mim` file in the `.xmimrc` file:

```
database: /home/lim/data/xmim.mim
```

Permission Algorithms

Setting Permissions for No One

If Unix write permissions for the database file (`xmim.mim`) are not turned on for the database owner, no one can write to the database because of normal Unix restrictions.

Setting Permissions for Everyone

If Unix write permissions on the database file (`xmim.mim`) is turned on for "others", everyone can lock the database.

When the Server Owner is running Locally on a Server (Free Pass)

In this scenario, the server owner is working on a system where the server and client applications are on the same system. In this case, the server owner can lock the database.

Setting Permissions for Individuals

For all other circumstances, users that need lock database permissions must be added to a new file(s): `xmimlock.acl`. The `xmimlock.acl` file(s) must reside in the same directory as the database schema (`xmim.mim`) file(s). There can be multiple databases, so for each `xmim.mim` schema file designating a database, there can be a `xmimlock.acl` file.

The syntax for an entry in the `xmimlock.acl` file is:

```
[username]@[hostname|ipaddr[/netmask]]
```

where

- arguments on either side of the `@` sign are optional including `username`, `hostname`, `ipaddr` and `netmask`.
- IP is in the format: `x.x.x.x[/n]`
- `"/n"` is the `netmask`. Without a `netmask`, an IP refers to a specific host. With the `netmask`, an IP can refer to a network.
- `hostname` might include a domain name or might be only a domain name. If only a domain name, there must be a `"."` in the name somewhere. An example would be `"@.com"`.
- the `username`, `hostname` and domain names are not case sensitive.

The following shows some example entries in the `xmimlock.acl` file:

```
lisa@matrix  
lisa@12.43.227.00/24  
lisa@12.43.227.45  
geoffrey  
lim  
@
```



After modifying the `xmimlock.acl` file, the MIM server does not need to be restarted in order for the changes to take affect.

Troubleshooting

If you are still having trouble locking a database, take note of the following:

- `lock_files` locks **all** databases in the current list of databases, so you may need more than one `xmimlock.acl` file even though you only wish to write to one database.
- Nobody can write to a database if the database owner does not have Unix write permissions to the database schema file, `xmim.mim`. To set the Unix permissions for the owner to read/write and the group and others to read only, enter:

```
chmod 644 xmim.mim
```

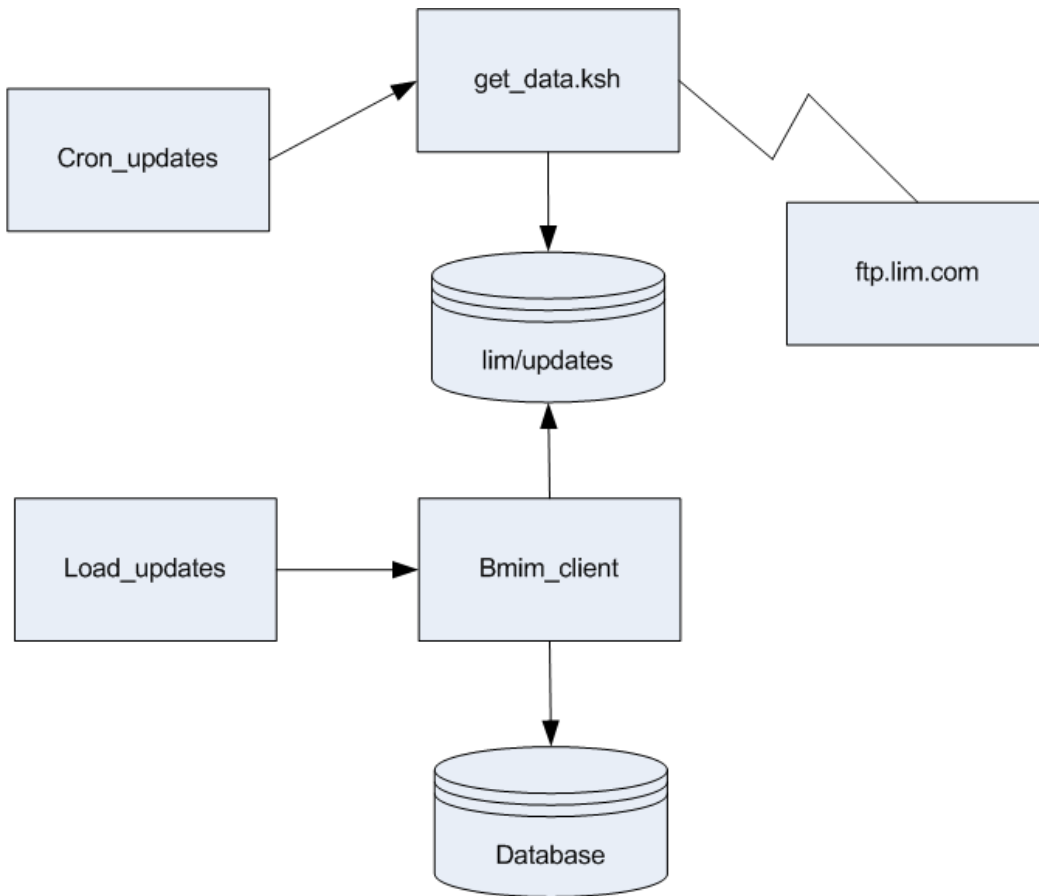
The permissions will appear as:

```
-rw-r--r-- xmim.mim
```

MIM Server Administration

Updating the Database

The following shows the MIM database update process:



The MIM update process functions as follows:

- `Cron_updates.sh` is setup as a batch job in the MIM DBA user's crontab and is the controlling process for the update procedure.
- `Cron_updates.sh` will execute the `get_data.ksh` script to retrieve the update files from the LIM FTP server and deposit them in the LIM update holding area (`/home/lim/updates`).
- `Cron_updates.sh` will start up the `load_updates.sh` script, which is the LIM database update routine.
- `Load_updates.sh` will use `bmim_client` to access and update the LIM database.
- Upon completion, `cron_updates.sh` will notify the designated system administrator about the success of the update via email.

Starting/Stopping the MIM Server

All actions listed below should be run while logged into the MIM DBA account. These scripts have the MIM environmental variables set for `LIMHOME`, `PORT` and `SERVERHOST`. They will act only on the MIM server for this account.

This command will stop the server:

```
server.info -k
```

This command will start the server:

```
start.server
```

To verify the status of the server execute:

```
server.info [-h] [-p server_number] [-k] [-r]
```

where (Optional entries are designated with brackets []):

<code>[-h]</code>	Help/Usage. Print this message and exit.
<code>[-p server_number]</code>	Server is at the port number <code>server_number</code> .
<code>[-k]</code>	Kill the server.
<code>[-r]</code>	Resynchronize the server with the <code>xmimrc</code> file.

By default the status will be returned for the server port defined in the `.limrc`. If the server is not running a message will be returned immediately stating `Cannot connect to server`.

Example output:

```
master server 0 on fourx is running
  process id:      23411
  owner:          lim
  log_level:      0
  log_file:       /home/lim/tmp/.xmim_server_0.log
```

```
database 0:      /home/lim/data/xmim.mim
license:        /home/lim/license/xmim.ids
version:        Ver 4.1.16 Compiled Mon Oct 8 17:34:50 CDT 2001
there are 2 slave servers registered
slave server 0:
    host name:      fourx
    process id:     26009
    owner:          lim
    status:         running
    last dispatched: Wed Oct 17 05:35:06 2001
slave server 1:
    host name:      fourx
    process id:     n/a
    owner:          n/a
    status:         not running
    last dispatched: n/a
```

System Monitoring

Update monitoring is aided by email notifications sent out when the MIM server runs the update process. There are four types of email notifications that are sent out to the email addresses defined in the .limrc file.

1. SUCCESSFUL – no errors encountered, update process complete
2. FAILURE – Errors encountered, update process was not able to continue. Errors must be located and corrected for normal database operation.
3. NOTICE – This message is a notification that a scheduled update process could not execute because another update process is already running, Only one update process at a time may run, no action is required.
4. WARNING (check log messages) – An error was encountered during the update process, but the update was able to complete. These errors are not critical and usually pertain to an individual series that did not update correctly. The error should be located and action taken to correct it.

Update logs are stored in `$LIMHOME/updates/logs`. These log files should be examined when errors occur to identify the error. If you need help in determining the error send the error message to support@lim.com for further assistance.


Disk monitoring is performed each time `cron_updates.sh` runs. Part of the output is a “df” of the MIM DBA partition. Do not let the system run out of disk space as this can corrupt the database.

Server logging is enabled by default at level 0 (lowest level). Log files are stored in `$LIMHOME/tmp` and can contain information about who is accessing the database, what is being accessed and types of access. Log files are stored in the common log format utilized by web servers.

The `xmim_svr_info` program provides information about the server and the ability to stop the server.

```
xmim_svr_info [-h] [-s host] [-p server_number] [-k] [-r] [-n]
```

where (Optional entries are designated with brackets []):

[-h]	Help/Usage. Print this message and exit.
[-s <i>host</i>]	Specifies the host that the master server is running on. The default will be the local host.
[-p <i>server_number</i>]	Specifies the <i>server_number</i> of the desired master server. If this argument as well as the -k and -r options are not specified, information will be returned for every master server that is running on the given host. The default server number if the -k or -r options are used will be 0.
[-k]	Specifies to kill a given master server.
	Note that the user must be either the owner of the server process or the super-user to kill a server. The default will be to kill master server 0 on the local host.
[-r]	Specifies to re-read the .xmimrc file which effectively causes the databases and any libraries/macros to be re-loaded. This will be effected only for one particular master server, the default being server 0 on the local host.
[-n]	Specifies to open a new log file with the standard log filename and begin logging to this new file. Simply renaming a log file will not result in logging to a new file because the file descriptor will still point to that file. Thus, to achieve archiving of log files, the old log file must be renamed and then the -n flag used to open a new log file, taking out a new file descriptor for it.

MIM Server Backups

What to Backup

It is important to backup the system on a regular basis. User accounts and the MIM database should be backed up on a reliable media. The database is comprised of three directories: data, dates and xmimrc. The database varies in size from 1gbs to 200gbs depending on the amount of data purchased and the amount of data loaded by the user. Most databases will be around 5-7gbs. Loading tick data or real time ticks will generate larger database sizes and receiving the TAQ database from LIM requires about 200gbs of disk space.

Scheduling

The database should be backed up once a week on Sunday mornings when activity is the least. Databases should not be backed up while updating and should always be done as a full backup.

Restoring

If a database is to be restored, notify LIM (support@lim.com) and they will queue up the necessary updates to bring the database up to date

Installing Test Servers

The following steps are required to setup a test database for loading customer data:

1. Create a new database and system for customer data only and read access to the LIM production database.
2. Modify the MIM system for multiple databases.
3. Create scripts to load data into the customer database.

A few assumptions:

- The new test database will be running on port 99.
- The MIM DBA user name is `limtest`.
- The MIM DBA home directory will be `/home/limtest`.
- All remaining steps are carried out logged in as the user `limtest`.

Setting Up a Test Server for Data Loading

When loading non-LIM supplied data (customer data) it is a good idea to take advantage of the 4x multiple database technology and to always load the data first into a test database. There will be two databases used. One database will contain the data LIM ships and the other will be the customer data. The `.xmimrc` file will need modifying and some scripts created to define the new database and access to the production database. This will make life easier in that if something happens to the LIM database it can be replaced without effecting the data the customer has loaded and vice versa. It also makes it easy to install the production database by simply copying the new customer database and modifying the `.xmimrc` file in production.

Once complete both databases are accessed as if they were one. When loading customer data a new command called `database_narrow` is used in order to specify which database data is being loaded into.

Create a New Database for Customer Data Only

First create a new server instance running on port 99 owned by the user `limtest` in the `/home/limtest` directory as described in “[Installation](#)” earlier in this document. Obtain the system release from the LIM “[Download](#)” Web page or from LIM’s FTP site: `ftp.lim.com` and login as anonymous with your email address as the password. Extract the system into the `/home/limtest` directory:

Next, login as `su` or login as user `limtest`:

```
su - limtest
gunzip -c rel.sys | tar xvf -
```

Copy the setup files from the templates directory as described below:

```
cp -p templates/start.server $LIMHOME/xmim/bin
cp -p templates/start.bmim $LIMHOME/xmim/bin
cp -p templates/server.info $LIMHOME/xmim/bin
```

Copy the dot files to the limtest home directory:

```
cp -p templates/.profile .profile
cp -p templates/.limrc .limrc
cp -p templates/.xmimrc .xmimrc
```

Modify `$LIMHOME/.limrc` to match your configuration. This file is used by several scripts to set environments. Set the two lines below to the appropriate values.

```
LIMHOME=/home/limtest
PORT="-p 99"
```

Copy the MIM server startup script to the rc directories. You must log in as the user root to perform this task.

```
su - root
cd /etc/init.d
cp -p $LIMHOME/templates/xmimserver xmimserver99
cd /etc/rc0.d; ln -s ../init.d/xmimserver99 K07xmimserver99
cd /etc/rc3.d; ln -s ../init.d/xmimserver99 iS98xmimserver99
vi /etc/init.d/xmimserver99
```

Make changes to these two entries:

```
LIMHOME=/home/limtest
LIMUSER=limtest
```

Save changes and logout of root.

Create a link to the existing license file on the production server:

```
cd /home/limtest/license
ln -s /home/lim/license/xmim.ids .
```

Next change directory to tmp and create two scripts:

```
cd /home/limtest/tmp
touch makedb
touch make_cust
chmod 775 makedb
chmod 775 make_cust
```

Edit `makedb` and put the following lines in it:

```
rm -r ../data
mkdir ../data ../data/xmim.mimdb
cd ../data/xmim.mimdb
mkdir daily intraday tick
touch daily/cust.fac intraday/cust.tck
touch tick/cust.tck
cd ../../mimdbcp
start.server
bmim_client -p 99 make_cust
```

Edit `make_cust` and put the following lines in it (make sure the paths match with your actual paths). The name `Cust_Data` that is used throughout these examples could be the name of the customer's company. (It is an arbitrary name.)

```
lock_files
relation_add {
  name = Cust_Data;
  parent = TopRelation;
  type = category;
}
database_def_daily_file TopRelation:Cust_Data daily/cust.fac
database_def_intraday_file TopRelation:Cust_Data intraday/cust.tck
database_def_tick_file TopRelation:Cust_Data tick/cust.tck

relation_delete EconomicIndicators;
relation_delete Equities;
relation_delete Futures;
relation_delete Indices;
relation_delete MonetaryIndicators;
relation_delete Other;

unlock_files;
```

Next execute `makedb`, this will create an empty 4x DB designed for customer data and may be run at any time the database needs to be rebuilt and emptied out. This script will remove the data directory. You must save this directory under a different name if you wish to have it preserved.

Testing the New Database

You can test the database by trying a print-relation using the `bmim_client` command.

```
bmim_client -p 99
print_relation {relation = TopRelation; verbose;}
```



Use the `-p` option to tell BMIM to access Port 99.

Press **Ctrl+d** to exit.

You should get a printout that looks similar to this:

```
Category:      TopRelation (Root of Relation Tree)
Category:      TopRelation:Cust_Data (Cust_data)
```

Where `Cust_data` is the name that you named your test data.



When loading data to the new database, make sure you are logged in as user `limtest` and loading data using port 99 (`bmim_client -p 99`).

Modify MIM System for Multiple Databases

Now rename the databases, edit the `.xmimrc` file and start up the server in order to access the two databases as one. The database name “`data.company`” should be renamed to match your company name.

First rename the databases:

```
cd /home/limtest
mv data data.company
```

Next edit the `.xmimrc` file.

Change the following line to point to the production database. This will be a read-only instance:

```
echo "database: $LIMHOME/data/xmim.mim"
to
echo "database: /home/lim/data/xmim.mim"
Insert this line above the production database entry:
echo "database:$LIMHOME/data.company/xmim.mim"
```

Example:

```
echo "database:$LIMHOME/data.company/xmim.mim"
echo "database: /home/lim/data/xmim.mim"
```

Save and exit.

Startup the server:

```
start.server
```

Now both the LIM and the customer databases are running under one 4x server.

Create Scripts to Load Data into Customer Database

In order to update the customer database it is very important that the following rules are followed:

- When loading data into the customer database the `database_narrow` command must be used at the beginning of each script before the database is locked.
- `database_narrow {database=~~/data.company/xmim.mim;}`
- Loading must be done as the user who owns the database.
- The home directory of the database owner is the top of the 4x tree.
- Category names should be mixed case.
- Relation names should be upper case.
- Column names are case independent.
- These rules will ensure that the data is correctly loaded into the proper database.

Implementing Data Loading into Production

Once the customer database is built and tested it is ready for production. To put it into production the `data.company` tree needs to be copied to the production directory and the production `.xmimrc` file needs to

be modified to reflect the addition of another database. Scripts and procedures that are used to update the customer database also need to be copied to production and all ownerships set to the production MIM DBA owner.

Client Applications

There are three client applications to install (excluding the APIs). See the document [Installing LIM Client Applications](#) for instructions on setting up the XMIM, MIMIC and MIM Excel Add-in applications.

CHAPTER 3

Entitlements

Introduction

Entitlements allow the system administrator to grant, revoke and manage privileges of the users. Using entitlements, the system administrator may exercise very fine control on any user's ability to read time series data. Write permissions are controlled by the lock files security controlled by the `xmimlock.acl` file. For more information on setting write permissions and writing to a database, see [Chapter 2, “MIM System Administration \(Red Hat Enterprise Linux RHEL 4\)”](#) or [Chapter 1, “MIM System Administration \(Solaris\)”](#) depending on your configuration. For an overview on security features see the “[MIM Server Security](#)” chapter in the *MIM Data and Development Guide*.

Entitlements are mainly administered using the Batch Mode process. The C API entitlement routines are documented and can be used to administer entitlements as well. Reference the “[Entitlements](#)” section in the “C/C++ API” chapter of the *MIM Data and Development Guide* for details on using the C API entitlement routines.

Concepts

The three key concepts are *Users*, *Groups* and *Permissions*.

User:	Users are individual users of the MIM system. On UNIX and NT platforms, the login name is used to represent a user (e.g., john, jane, judy).
Group:	A Group refers to a collection of users. Groups have names, e.g., developers (with users john and jane). A Group also points to a permission set that specifies the read privileges for that group.



A user may belong to multiple groups.

When a user seeks to access a time series, the user's groups are determined. If such groups exist, then the system checks to see if the user has the appropriate privileges.



Note for users with release prior to version 4.2: If no groups are found for the user, the system uses the privileges of a special **user** called "**anonymous**" if such a user is defined.

Finally, if the user does not belong to any defined group, the user is denied access to the system database.

Permissions:	Permissions consist of a set elements of the form {relation, column, read=[1 or 0]} . Each element of this set represents the read permissions for a relation/column.
--------------	--

Example 1:
`TopRelation TopColumn 0`

This permission denies read access on all relations and all their columns (i.e., all the possible time series) in the database.

Example 2:
`Equities Price 1`

This permission represents a read access to all the Price data of all the equities in the database.

Inheritance is supported such that, to find the permissions for a given relation column that does not have explicit permissions, the column and relation hierarchies are traversed up until a node is found that has associated permissions. For example, to find the permissions for Close of IBM, given a column hierarchy of TopColumn->Price->Close and a relation hierarchy of TopRelation->Equities->IBM, first the column hierarchy is traversed such that if IBM Close did not have associated permissions then IBM Price would be looked at and then IBM TopColumn. After that, the relation hierarchy is walked up: Equities Close, Equities Price, Equities TopColumn, TopRelation Close, TopRelation Price, TopRelation TopColumn.

Thus, the innermost loop is the column hierarchy and the outermost loop is the relation hierarchy. To determine the access privileges for a given relation and column, using the algorithm described above, the most specific relation/column/read permission is obtained. This allows the system administrator to exercise very fine control over the system. For example the administrator can override the permissions for a sub-tree of the relations/columns hierarchy by simply specifying a permission for that relation/column. For example, suppose that the administrator wants to grant read privileges for all the data except, say Futures. The following two permissions do the trick:
`TopRelation TopColumn 1 TopRelation Futures 0`

Free Pass

Entitlements has a “free pass” in place such that the person who starts the server also has permission to read any entitlements if they go through the same server host.

Enabling/Disabling Entitlements

By default, the entitlements functionality is turned off. The administrator may turn it on by inserting the following line in the `.xmimrc` file:

```
entitlements: yes
```

To turn the entitlements facility off:

```
entitlements: no
```

Even if the entitlements functionality is enabled, no entitlement checking is done if the entitlements database has not been created. In other words, if the entitlements database does not exist, then there are no access restrictions.

Usage Guidelines

By necessity, the method for processing entitlements was designed to minimize the impact of employing entitlements with respect to accessing data. Thus, an in-memory structure was used with entries for each permission tuple/group pairing. As the number of entries for this structure grows, the impact will be slower access to all data since the system will be required to check this structure prior to granting access for any data request. Thus, the following guidelines should be adhered to in using entitlements:

1. Minimize the number of groups used. The number of users per group does not affect performance so as many users as desired may be added without impact but the number of groups used should be kept small.
2. Minimize the number of permission tuples. Permission sets should be set for relations as high up in the relation hierarchy as possible, utilizing the inheritance feature for descendants. Exceptions can be specified as necessary to override inherited permission sets for specific relations.

Batch Processing Mode

Use the Batch Process mode to administer your entitlements. The Batch Process mode can set many entitlement changes at one time.

Example:

```
lim    TopRelation    TopColumn    1
group_c    Equitites:i:ib    FundamentalCat:Stocks    0
```

If the user belongs to `group_c` and also to `group lim`, the user will not be allowed to read `BookValue` of IBM because the user is restricted by the second entry.

Creating Entitlements Text File

The following outlines the instructions for processing entitlements in a batch process. Create a text file and include the following syntax:

```
#####
# Instructions for using batch mode...
#####
# specify a list of groups to add, one per line after "[groups]"
[groups]
lim
limdev
# specify a list of users to add to a group, one per line after "[users]"
[users]
stephen limdev
rg lim
# specify a list of permissions to add to a group.
# each line should be: group relation column read_flag
# 1 means (yes for read)
# 0 means (no for read)
[permissions]
lim TopRelation TopColumn 1
lim Equities TopColumn 0
limdev TopRelation TopColumn 1
limdev Equities:i:ib FundamentalCat:Stocks 0
# same rules apply for deleting relations, but different keywords:
# [delete_permissions]
# [delete_users]
# [delete_groups]
```

Where:

[password] – Note for release 4.2 and higher: Password is no longer used. Password is ignored if entered.

[groups] – Assign group names. The example above includes the group names “lim” and “limdev”.

[users] – Add the username then the group to assign the name to. For example, “stephen” is assigned to group “limdev”.



Note: A user can belong to multiple groups.

[permissions] – Specify a list of permissions to add to a group. Each line should be: group relation column read_flag.

The following shows the read/write permission settings:

1 means (yes for read)
0 means (no for read)

Example:

```
lim TopRelation TopColumn 1
lim Equities TopColumn 0
limdev TopRelation TopColumn 1
limdev Equities:i:ib FundamentalCat:Stocks 0
```

Adding read permission for the group `lim` to `TopRelation TopColumn` gives the group `lim` read permissions for everything in the database.

Adding the next line `lim Equities TopColumn 0` takes away read permissions for everything in the `Equities` category. Now group `lim` can read all relations except for equities.

The next line in the example, `limdev TopRelation TopColumn 1` gives read permissions to the group `limdev` for everything in the database.

The last line shows how you can revoke permissions by taking away read permissions for `limdev` for the columns in the `FundamentalCat:Stocks` category and the relations listed under the `ib` category in `Equities`. For example, a user in `limdev` would not have permission to read the `BookValue` of `IBM`.

[delete_permissions] – Use `delete_permissions` to remove permissions for a user or a group of users. The same rules apply to deleting permission as to adding permissions.

For example: To take away the write permissions for group `lim` on `TopRelation TopColumn`:

```
[delete_permissions]
lim TopRelation TopColumn 1
```

[delete_users] – Use `delete_users` to delete users from a group

Example: Deletes Stephen from the group `limdev`.

```
[delete_users]
stephen limdev
```

[delete_groups] – To delete a group, list the group names under `[delete_groups]`:

Example: Deletes the group `limdev`.

```
[delete_groups]
limdev
```

Example Entitlements Batch File (entitlements1.txt)

```
#####
# entitlements batch file...
#####
# specify a list of groups to add, one per line after "[groups]"
[groups]
lim
group_a
group_c
# specify a list of users to add to a group, one per line after "[users]"
[users]
lim lim
lisa group_a
lisa group_c
# specify a list of permissions to add to a group.
# each line should be: group relation column read_flag
# 1 means (yes for read)
# 0 means (no for read)
[permissions]
lim TopRelation TopColumn 1
group_a TopRelation TopColumn 1
group_c WID.LONESTAR_1 TopColumn 0
```

Sample BMIM Script to Run (ent1.bmim)

See the “[BMIM Scripting Language](#)” chapter in the “*MIM Data and Development Guide*” for detailed information on creating BMIM scripts.

The following BMIM commands are used for running entitlements:

1. `entitlements_clear` - The BMIM command `entitlements_clear` clears all previous entitlements information.
2. `entitlements_read` - The BMIM command `entitlements_read` loads new entitlements information from the batch input file.
3. `entitlements_write` - The BMIM command `entitlements_write` dumps entitlements to an output file.

```
# with entitlements1.txt, user lisa does not have read permission to read the WID.LONESTAR_1
#relation.
database_narrow {
    database = /home/lisa/data_complete/data/xmim.mim;
}
lock_files

# clear all previous entitlements information
entitlements_clear;
#load new entitlements information from batch file
entitlements_read /home/lisa/entitlements1.txt;
# dump entitlements to output file
entitlements_write /home/lisa/entitlements1_out.txt
facts_write {
    file = facts.bmim;
    relation = WID.LONESTAR_1;
```

```

    column = Storage;
    from_date = 01/01/2001;
}
unlock_files

```

Results of Running Sample BMIM Script

```

europa 8% ./bmim_client -p 6 -q matrix /home/lisa/ent1.bmim
Processing Bmim File: /home/lisa/ent1.bmim
Locking Files
Clearing Entitlements Information.
Reading Entitlements File: /home/lisa/entitlements1.txt
Writing Entitlements File: /home/lisa/entitlements1_out.txt
Writing Facts File: /home/lisa/xmim/4.1/src/facts.bmim
ERROR: write_facts line 26 in /home/lisa/ent1.bmim
| ERROR: User is not Entitled to read Storage of WID.LONESTAR_1 in database=/home/lisa/
data_complete/data/xmim.mim.
    User belongs to the following 2 Entitlement groups: group_a group_c
Unlocking Files
europa 9%

```

Sample Entitlements Batch Output File (entitlements1_out.txt)

```

#####
# Auto-Generated Entitlements Batch File # # # # # # # # #
#####
# generated: 09/16/02 09:03:22
# database schema = /home/lisa/data_complete/data/xmim.mim
#####
[groups]
# List of groups. Format is: groupname
group_c
group_a
lim
[users]
# List of users. Format is: username groupname
# where username can be any of the following formats:
# user, user@host, user@ip, or user@ip/netmask
lisa group_c
lisa group_a
lim lim
[permissions]
# List of permissions. Format is: group relation column readFlag
group_c WID.LONESTAR_1 TopColumn 0
group_a TopRelation TopColumn 1
lim TopRelation TopColumn 1

```

Processor Tag

In BMIM, a “#” usually designates a line that is commented out and will not run. The processor tag “#v4.2” followed by a space does not function as a commented out line but is recognized as part of the active script.

The processor tag is implemented so that BMIM scripts containing the new BMIM commands can be ignored with versions prior to MIM server version 4.2. The commands are only executed on version 4.2 and higher servers.

Example BMIM Script Using the Processor Tag

```
database_narrow {  
    database = /home/lim/data/xmim.mim;  
}  
lock_files  
#v4.2 entitlements_read entitlements.txt;  
unlock_files
```

Index

B

Backups

- MIM Server (Red Hat Enterprise Linux), 41
- MIM Server (Solaris), 18

Batch Processing Mode

- Entitlements, 54

C

Client Applications

- Red Hat Enterprise Linux, 47
- Solaris, 24

E

Entitlements, 49

- Batch Processing Mode, 54
- Enabling/Disabling, 52
- Example Batch File, 56
- Free Pass, 51
- Processor Tag, 57
- Sample BMIM Script, 56
- Text File, 54
- Usage Guidelines, 53

F

Free Pass, 10, 33, 51

I

Installation

- MIM Server (Solaris), 4
- Red Hat Enterprise Linux, 28

Installing Test Servers

- MIM Server (Red Hat Enterprise Linux), 42
- MIM Server (Solaris), 19

M

MIM Database Write Permissions (Red Hat Enterprise Linux), 35

- Permission Algorithms, 35
- Troubleshooting, 36

MIM Database Write Permissions (Solaris), 12

Permission Algorithms, 12

- Everyone, 12
- Free Pass, 12
- No One, 12

Troubleshooting, 13

MIM Server (Red Hat Enterprise Linux)

Permissions, 33

MIM Server (Red Hat Enterprise Linux)

Administration

- Starting/Stopping the Server, 38
- System Monitoring, 39

Architecture, 25

Backups, 41

Installing Test Servers, 42

MIM Server (Solaris)

Administration, 14

- Printing, 17
- Starting/Stopping the Server, 15
- System Monitoring, 16
- Updating the Database, 14

Architecture, 1

Backups, 18

Installation, 4

Installing Test Servers, 19

Permissions, 10

- Free Pass, 10
- xmimsvr.acl File, 10

Tree Structure, 2

MIM Server Administration (Red Hat Enterprise Linux), 37

MIM System Administration (Red Hat Enterprise Linux), 25

MIM System Administration (Solaris), 1

P

Permissions

- MIM Database (Red Hat Enterprise Linux), 35
- MIM Server (Red Hat Enterprise Linux), 33
- Free Pass, 33
- MIM Server (Solaris), 10

Printing

- MIM Server (Solaris), 17

Processor Tag

- Entitlements, 57

S

Starting/Stopping the Server

MIM Server (Red Hat Enterprise Linux), [38](#)

MIM Server (Solaris), [15](#)

System Monitoring

MIM Server (Red Hat Enterprise Linux), [39](#)

MIM Server (Solaris), [16](#)

U

Updating the Database

MIM Server (Red Hat Enterprise Linux), [37](#)

MIM Server (Solaris), [14](#)

X

xmimsvr.acl File, [33](#)

MIM Server (Red Hat Enterprise Linux), [33](#)

MIM Server (Solaris), [10](#)