

MESA Software

Installation Guide Unix Binary Option

**Electronic Radiology Laboratory
Mallinckrodt Institute of Radiology
510 South Kingshighway Blvd.
St. Louis, MO 63110
314.362.6965 (Voice)
314.362.6971 (Fax)**

Revision 9.8.0
5-December-2005

Copyright © 2005: Washington University

Contents

1	Introduction.....	2
1.1	Target Directory	2
1.2	Storage Directory.....	3
1.3	Disk Space	3
1.4	Uninstalling Previous Versions	4
2	MESA Internal Software Packages	5
2.1	MIR Central Test Node (CTN) Software	5
2.2	HL7IMEXA.....	5
2.3	PostgreSQL.....	5
2.4	Xerces C	5
2.5	Apache Tomcat Web Server.....	6
3	Installation from Compiled Binaries	7
3.1	Account Information	7
3.2	Database Installation	7
3.3	Perl Installation.....	9
3.4	Java Installation	10
3.5	Apache Tomcat Installation.....	10
3.6	OpenLDAP Installation	10
3.7	Agfa/Philips DVT.....	11
3.8	MESA Installation	11
4	Build Databases	13
4.1	Build the MESA Databases: Unix.....	13
5	Install the MESA Storage Directory.....	14
6	Runtime Notes	15
A	RIS Mall Installation (Linux)	16

1 Introduction

This document describes the installation procedure for MESA software for select Unix systems. Users reading this document have chosen to install the MESA software using pre-compiled binaries. Installation instructions for Win32 binaries or from source code are found in other documents.

Check mark	Task #	Task	Section Reference (this document)
	1	Verify operating system compatibility for MESA test tools	
	2	Decide upon the target directory structure you will use (or use defaults)	
	3	Verify available disk space	
	4	Verify previous versions of MESA test tools are not installed	
	5	Install/configure PostgreSQL database	
	6	Install PERL	
	7	Install PERL module - date	
	8	Install the Sun Java 2 Software Development Kit	
	9	Install MESA Test Tool software and scripts (includes necessary CTN software)	
	10	Build SQL database tables	
	11	Install the MESA Storage files	
	12	Create Environment Variables	
	13	Begin executing test scripts by referring to IHE Test Plan documents	Other documents

1.1 Target Directory

MESA is designed to be installed in a target directory specified at build time. We use the environment variable MESA_TARGET for this directory; the default value is */opt/mesa* for Unix

systems. Once the installation procedure is complete, all software depends only on executables, configuration files, data files in the target directory and files in the storage directory (see 1.2).

The target directory contains these subdirectories:

bin	The compiled executable programs
db	Files for controlling database operations
lib	Shared libraries loaded at runtime
logs	Logfiles for all MESA server applications
runtime	Configuration files for applications
mesa_tests	Scripts/data for test protocol
webmesa	A holding directory used for the second stage of the installation process.

Some parts of the MESA software are designed for installation with the Jakarta Tomcat web server. Those pieces and the installation directory will be defined in this document.

1.2 Storage Directory

The MESA storage directory is used to store messages and datasets received from peer applications. For example, the MESA Image Manager will store images received from modalities in this directory. The default for this is `/opt/mesa/storage`.

The location of this directory is recorded in the environment variable `MESA_STORAGE`. We separate this from the target directory to give you more control over disk partitions on your system. The space required for this directory depends on the tests you perform.

1.3 Disk Space

Users installing the MESA software will need to be aware of the disk requirements for the system. If you want to use the default target directory and you have insufficient disk space in that partition, you can use soft links in the file system.

Operating System	Software	Space (MB)
Solaris 7	Postgres	
Solaris 7	Java JDK	
Solaris 7	MESA Installation in target directory	
Redhat 8.0	Java JDK	
Redhat 8.0	MESA Installation in target directory	

1.4 Uninstalling Previous Versions

There are no incremental upgrades to the MESA software. Each installation writes over the top of the previous installation (unless you choose to retain the old version in a separate directory). To remove a previous version, follow these steps:

- `cd $MESA_TARGET/db`
- `make uninstall`
- `cd $MESA_TARGET`
- `rm -r *`

The first two steps remove existing MESA databases. The last two steps remove all files from the previous installation and prepare you for the current installation.

2 MESA Internal Software Packages

The MESA libraries and applications are built on top of several other software packages. This section only describes those packages that are compiled into the MESA software.

2.1 MIR Central Test Node (CTN) Software

The MIR Central Test Node (CTN) software is a public domain DICOM implementation that was originally funded by the RSNA and is maintained by MIR. This software is written in the “C” programming language and provides libraries that support various parts of the DICOM Standard or natural applications of the DICOM Standard. These include Information Object creation, encoding and parsing, network operations, DIMSE services and network operations. The CTN software also provides a number of simple test programs (transmit this image) and more complete applications (image server).

An extension to the CTN software called JAVACTN is used by the MESA software. A small set of Java classes are used to support some DICOM operations, such as Information Object manipulation. This software uses the Java Native Interface (JNI) to provide access to the CTN C libraries.

The CTN software requires some updates to satisfy the MESA design goals. An updated version of the CTN software is included with the MESA distribution.

2.2 HL7IMEXA

HL7IMEXA is a table-driven HL7 message builder/parser that has its origins at Columbia University. The Columbia implementation was called HL7IMEX. The software was modified by Allen Rueter of MIR and renamed HL7IMEXA. This software is written in the “C” language and supports message building and parsing, but no network operations. Minor changes are made to this package to support the C++ environment used by MESA and to correct some runtime problems.

2.3 PostgreSQL

PostgreSQL is an SQL-based relational database from the University of California at Berkeley (UCB). This is the latest in a series of database engines and is currently supported by UCB. It is of interest to the MESA developers because it uses a standard SQL interface and may be used without royalty by both university groups and corporations. The PostgreSQL software is integrated into the CTN software, as are a number of other relational database products. The MESA and CTN software use this database for Unix systems.

2.4 Xerces C

Xerces C is a library maintained by the Apache foundation used to parse XML data. Quoting from the Apache web page:

Xerces (named after the Xerces Blue butterfly) provides world-class XML parsing and generation. Fully-validating parsers are available for both Java and C++, implementing the W3C XML and DOM (Level 1 and 2) standards, as well as the de facto SAX (version 2) standard. The parsers are highly modular and configurable. Initial support for XML Schema (draft W3C standard) is also provided

2.5 Apache Tomcat Web Server

The Apache Tomcat web server is maintained by the Apache foundation (www.apache.org). This web server is designed to support Java servlets and provides the mechanism we use for web applications.

3 Installation from Compiled Binaries

3.1 Account Information

The MESA software is run from a normal user account. In the examples below, we will assume you have created an account with the login mesa. You are free to use a different user account. The software requires root privileges for installation; no root privileges are required to run the software.

3.2 Database Installation

Red Hat Linux 8.x, 9.x ships with PostgreSQL as an installation option. You should be able to use the version of PostgreSQL that ships with the system.

Solaris does not come with this software. Please install the version 7.3.4 of PostgreSQL found on the MESA distribution web page.

3.2.1 Database Installation – Solaris

The PostgreSQL database is installed in a directory separate from the MESA software. This software is a self-contained package with a server application for managing the database. The compiled software is distributed in a tar file whose name identifies both the version of PostgreSQL and the version of the operating system that we use.

The installation steps are:

1. Create a postgres group. The value for the group identifier is not important.
2. Create an account with the user name postgres. The numeric userid for that account is arbitrary. This account should be in the postgres group. Table 3.1-1 lists specific values that should be in the path variables for the postgres account and other environment variables.
3. We install the software in /opt/postgres. As root, create the directory /opt/postgres and then “cd” to that directory.

```
mkdir /opt/postgres
```

```
cd /opt/postgres
```

4. We have supplied tar files with the PostgreSQL system compiled for Solaris 7. Identify the appropriate file and untar the contents. For example:

```
tar xf postgres734_sol7.tar
```

5. Change the ownership and group of the postgres files:

```
chown -R postgres:postgres /opt/postgres
```


- There is a one-time setup step required for postgres. Login under the postgres account and make sure the proper environment variables are set. Initialize the database as follows:

```
mv /opt/postgres/data /opt/postgres/data.old
initdb
```

You should only have to do this one time; this is done before you start the postgres server.

- Log in under the postgres account and start the server application. Remember to use the environment variables listed in Table 3.1-1.

```
postmaster -i &
```

The `-i` switch allows us to use a JDBC connection for the MESA RID Information Source.

Table 3.2-1 Environment Variables for Postgres Account (Solaris)

System	Variable	Contains
Solaris	path	/opt/postgres/bin
	LD_LIBRARY_PATH	/opt/postgres/lib
	PGLIB	/opt/postgres/lib
	PGDATA	/opt/postgres/data

3.2.2 Database Installation – Redhat Linux

The PostgreSQL database is installed in a directory separate from the MESA software. This software is a self-contained package with a server application for managing the database. The software under Redhat Linux is available as an RPM and can be installed using the standard RPM procedures.

The installation steps are:

- Install the Redhat package for the server SQL Database Server. That will install PostgreSQL. You can skip the extra package for mysql. This can be installed when the system is first installed or later. If installing after the fact, try

```
/usr/bin/redhat-config-packages
```

- Edit the file `/var/lib/pgsql/data/pg_hba.conf`. There is a line that is commented out that allows all local database connections on the machine. Assuming you trust your local users, uncomment this line

```
host    all    all    127.0.0.1    255.255.255.255    trust
```

3. Edit the file `/etc/rc1.d/K15postgresql`. We need to pass the `-i` option to the postmaster. In the lone line that starts the postmaster, the `-o` switch is used to pass parameters to the postmaster binary. In the quoted area before `"-p ${PGPORT}"`, add `-i`. That is:

```
-o '-i -p ${PGPORT}' start ...
```

Stop and then start the postmaster:

```
./K15postgresql restart
```

Table 3.2-2 Environment Variables for Postgres Account (Redhat Linux)

System	Variable	Contains
Red Hat Linux 8.0, 9.0	path	should already be correct
	PGLIB	/usr/lib/pgsql
	PGDATA	/var/lib/pgsql

3.3 Perl Installation

Starting with version 8.0.0 of the MESA tools, Perl modules that are not part of the standard perl distribution are used. Read this section about Perl installation, and also refer to section 3.3.1.

Perl is shipped with Linux and requires no installation.

A Perl installation package for Solaris is available at <http://www.sunfreeware.com> and is included on the MESA distribution web page. To install the software in `/usr/local/bin`, perform this step as root:

```
pkgadd -d perl-5_005_02-sol26-sparc-local
```

For other operating systems, obtain Perl from <http://www.perl.com>. Install perl according to the directions in the distribution.

3.3.1 Additional Perl Modules

Version 8.0.0 of the MESA tools uses additional Perl modules. These can be found on the web distribution page. Install as follows:

Date/Manip.pm	No need to install; included in <code>mesa_tests/actors/rad/common/scripts</code> . If you find a conflict with your perl installation (perl already has this module), remove the MESA copy.

3.4 Java Installation

Install the Sun Java 2 Software Development Kit (J2SDK) for version 1.4.2 or higher. The JRE is not sufficient. You need to obtain this from the Sun web server: www.java.sun.com. We are not allowed to redistribute the J2SDK.

The table below lists the installation directories that we use for testing. These values are not required, and you might choose different directories.

Table 3.4-1 Java Installation Directories

System	Director
Solaris	/opt/j2sdk1.4.2_02
Red Hat Linux 8.0, 9.0	/usr/java/j2sdk1.4.2_02

3.5 Apache Tomcat Installation

The Apache Tomcat server is used by the MESA Information Source actor to supply summary information and documents in response to RID queries. Install version a 5.0.x version that is at 5.0.28 or higher.

The table below lists the installation directories that we use for testing. These values are not required, and you might choose different directories.

Table 3.5-1 Tomcat Installation Directories

System	Director
Solaris	/opt/jakarta-5.0.28
Red Hat Linux 8.0, 9.0	/opt/jakarta-5.0.28

3.6 OpenLDAP Installation

OpenLDAP is required for PWP tests. As of this document (version 9.0.0), we have only tested OpenLDAP with Linux installations. You may already have some of these RPMs installed on your system.

1. Query your Linux system for existing OpenLDAP packages:

```
rpm -qa | grep -i openldap
```

2. Obtain and install the missing RPMs from the following list:
 - a. openldap
 - b. openldap-clients
 - c. openldap-servers
3. The table below lists the versions of these packages tested in our laboratory. We found the RPM installer did not want to install some RPMs because the version numbers were different. We forced the installer to ignore the versions as follows:

```
rpm -- install -- force package
```

This table lists a version of LDAP RPMs tested on a Redhat system.

openldap-2.0.27-11
openldap-clients-2.0.27-11
openldap-servers-2.0.27-8

3.7 Agfa/Philips DVT

The Agfa/Philips DVT is used to examine DICOM composite objects. This is only available on Windows versions. If installing on Unix systems, you will use the David Clunie DICOM3TOOLS (included in the release package).

3.8 MESA Installation

The pre-compiled MESA software is included in a tar file whose name includes the operating system version. The instructions below assume that you install in the default installation directory: `/opt/mesa`. You may choose a different directory; just be sure to adjust the directory names given below and to set the runtime environment when you use the software.

1. Create the directory `/opt/mesa`; make the owner mesa (or whatever account you have chosen).
2. Login under the mesa account and untar the MESA software:

```
cd /opt/mesa
zcat mesa-sol7.tar.Z | tar xf -
```

3.8.1 Complete Installation – Tomcat (e.g., RID Profile)

Complete the installation by copying MESA files to the Tomcat directory. You need to set the environment variables TOMCAT_HOME and MESA_TARGET appropriately. In the directory \$MESA_TARGET/webmesa/mesa-iti:

1. Edit the file web_unix_postgresql.xml (use notepad).
2. Find the two different initialization parameters whose param-name is logPath. Change the first part of the param-value from /opt/mesa to the directory where the MESA software is installed (i.e., the value for \$MESA_TARGET). Leave the segment that says /logs/info_src.log as is.
3. Find the two different initialization parameters whose param-name is jdbcConnectURL. Examine the values for user in the URL; the default value is postgres. You should be able to leave this value as is. If you have trouble connecting to the database, change the value to the user ID of the person who created the databases.
4. Find the one initialization parameter whose param-name is imagePath. Change the param-value to the partition and directory where you have chosen to install the MESA storage data. The default value is /opt/mesa/storage.
5. To complete this installation step:

```
cd $MESA_TARGET/webmesa/mesa-iti
make -f Makefile.iti install
```

4 Build Databases

The installation steps defined in section 3 install the MESA software in the target directory. You need to complete the steps in this section to build the MESA databases.

4.1 Build the MESA Databases: Unix

The MESA installation contains a Makefile that will be used to build the MESA databases for all simulators. Even if you do not require a specific database (because you are substituting your application for a MESA simulator), you should make all of the databases.

The postgres software requires some environment variables to be set when you build databases or use database applications. These variables are listed in the table below:

Table 4.1-1 PostgreSQL Environment Variables

Variable	Value (Solaris)	Value (Red Hat Linux 6.0)
PGLIB	/opt/postgres/lib	/usr/lib/pgsql
PGDATA	/opt/postgres/data	/var/lib/pgsql
PGUSER	postgres	Postgres

Use the mesa account to build the databases:

1. set the PGUSER environment variable defined in Table 4.1-1
2. Create the database tables and load with test values:

```
cd /opt/mesa/db
make database
```

If you have trouble creating the database, here are some things to check;

1. The Postgres initdb command has been as a one time setup. This is described in previous sections of this document.
2. The proper environment variables are set (Table 4.1-1).
3. The Postgres server (postmaster) is running and is owned by the postgres account.

4.1.1 Load Databases with Test Data

The ITI tests require some initial data in the SQL database. The installation procedure automatically loads the tables. To check the installation:

```
psql info_src
>select * from reports;
```

5 Install the MESA Storage Directory

The MESA storage directory is separate from the MESA_TARGET directory. We distribute two separate tar files for installation. Use the table below to determine which tar files are appropriate for your installation. You only need to install the radiology storage files if you are testing radiology integration profiles.

Table 5.1-1 MESA Storage Files

MESA Tests	Storage File Name
Radiology, Cardiology	rad_storage_8.3.0.tar.Z
IT Infrastructure	iti_storage_8.3.0.tar.Z

To create the storage directory:

1. Set the environment variable \$MESA_STORAGE
2. Create the top level storage directory
3. mkdir \$MESA_STORAGE
4. Install the Radiology storage files (if necessary)

```
cd $MESA_STORAGE
```

```
zcat rad_storage_8.3.0.tar.Z | tar xf -
```

5. Install the IT Infrastructure storage files

```
cd $MESA_STORAGE
```

```
zcat iti_storage_8.3.0.tar.Z | tar xf -
```

6. Edit the file \$MESA_STORAGE/ecg/ecg20304.xml. This file has URLs that refer to the server info-src. Replace the server name with the name of the system running the Tomcat server in all URLs, or make a hostname entry to map the name info-src to that system.

6 Runtime Notes

There are environment variables that need to be set when you run the test scripts. These are discussed in previous sections of the document and listed again in the table below. Note that these are for the account that runs the test scripts (mesa), not the postgres account.

Variable	Value
MESA_TARGET	/opt/mesa (or another value of your choosing)
MESA_STORAGE	/opt/mesa/storage (or another value of your choosing)
PGUSER	postgres
PATH	should include \$MESA_TARGET/bin and the path to perl
LD_LIBRARY_PATH	should include \$MESA_TARGET/lib
JDK_ROOT	/opt/j2re1.4.2_02 (or another value of your choosing)
JAVA_HOME	Same value as JDK_ROOT
JAR_DIRECTORY	/opt/mesa/lib (\$MESA_TARGET)/lib
MESA_OS	One of SOLARIS, LINUX (check with Project Manager for other values)

The LD_LIBRARY_PATH that we use for Solaris systems is:

```
LD_LIBRARY_PATH=/opt/SUNWspro/lib:/usr/openwin/lib:/usr/lib:/usr/dt/lib:/usr/ucblib:/opt/postgres/lib:/opt/mesa/lib
```


A RIS Mall Installation (Linux)

A.1 Prerequisites

The RIS Mall consists of a series of PHP scripts which connect to the PostgreSQL server and which are executed by the Apache server. In order to make use of the RIS Mall, MESA needs to be installed, along with PostgreSQL, PHP and Apache (as described below). In addition, the php-pgsql package must be installed (there is an RPM by that name).

A.2 PostgreSQL

Please note that PostgreSQL version 7.3 or higher is required. The notes below are similar to the ones described above for PostgreSQL installation.

1. Make sure postgres is running (psql -l should yield something other than "cannot connect")
2. Make sure the PGUSER variable is set to *postgres*. This may be accomplished by putting the following line in the *.cshrc* file:

```
setenv PGUSER postgres
```

3. Edit postgres config file *pg_hba.conf*. This may be set according to security policy. This should work:

```
local    all             all                                     trust
host     all             all             127.0.0.1          255.255.255.255   trust
host     all             all             0.0.0.0            0.0.0.0           reject
```

The file may be found in */var/lib/pgsql/data* on Red Hat systems and */etc/postgresql* on Debian systems. Consult postgres documentation if you cannot find it. You will need to restart postgresql after making changes to this file.

4. Confirm that you can read from the database. Running "psql -l" should yield a list of available databases.
5. Allow tcp/ip connections on port 5432. To check if postgresql is accepting port 5432 connections, type

```
telnet localhost 5432
```

If the connection is refused, you need to edit *postgresql.conf* file (same directory as *pg_hba.conf*). Set:

```
tcpip_socket = true
```

Restart postgresql after making changes.

6. Make the MESA databases (for MESA tests and RIS Mall)

```
cd $MESA_TARGET/db
make database
```

A.3 Apache

1. Make sure apache (httpd) server is started. To check, type

```
telnet localhost 80
```

If connection is refused, apache needs to be started.

2. Edit apache.conf file in \$MESA_TARGET/webmesa/ris_mall
 - a. configure MESA_TARGET correctly
 - b. configure ris_mall alias correctly
 - c. configure security permissions

these configuration issues are explained in that file.

3. Edit the main apache configuration file. This is typically httpd.conf. On a Red Hat system it is usually found in */etc/httpd/conf*; on Debian it is */etc/apache*

Add the following to the end of the httpd.conf file:

```
include "/opt/mesa/webmesa/ris_mall/apache.conf"
```

Of course, be sure the path is to the file created in step 2 above.

After editing httpd.conf restart apache to have the changes take effect.

A.4 PHP

1. Make sure PHP has postgres support. In order to determine if postgres support is loaded, you may go to the page *http://localhost/ris_mall*.

If there are no errors after you log in, postgres support is enabled. Alternatively, you may create the file test.php with the following text:

```
<?php phpinfo(); ?>
```

in the doc directory. Opening that file in the browser will tell you what modules are loaded. If pgsqll appears as a section in that page, postgres support is loaded; proceed to step 3.

2. If postgres support is not enabled, make sure that the php-pgsql package is loaded. You may need to edit php.ini (*/etc/php.ini* for Red Hat, */etc/php4/apache/php.ini* for Debian) and put the following line in:

```
extension=pgsql.so
```

In order to perform further debugging on PHP, change the following lines in php.ini:

```
display_startup_errors = On
log_errors = On
error_log = /home/maw/php.error (or any other filename; be
sure file exists)
```

This will write all errors PHP encounters to the error_log file for further analysis.

3. Ris_mall requires the register_globals to be enabled. In php.ini (see above) set

```
register_globals=On
```

apache will need to be restarted after this.

A.5 RIS Mall

1. Configure the destination configuration files in \$MESA_TARGET/webmesa/ris_mall/config. They begin with hl7* and correspond to destinations for the adt, order filler and order placer actors. For tests on the local machine, it is necessary to just have the entries with localhost as the destination.

After editing these files, it is necessary to recreate the ris_mall databases (this does not take as long as creating all of the MESA databases). To do so,

```
cd $MESA_TARGET/db
make clear-webmesa
```

This is also useful for clearing out data from ris_mall tables during testing.

2. start the ris mall servers.

```
cd $MESA_TARGET/bin
./start_ris_mall_servers.csh
```

The servers may be stopped with the command

```
./stop_ris_mall_servers.csh
```

It is useful to confirm that the DICOM server is running by performing the command,

```
$MESA_TARGET/bin/dicom_echo localhost 4250
```

3. It may be necessary to change the firewall configuration in order to allow access to the RIS Mall services from remote hosts. Make sure that ports 4100, 4200 and 4250 are not blocked.
4. At this point the RIS Mall should be available for use. To access it, access the web page:

```
http://localhost/ris_mall
```

For further information, consult the RIS Mall User's Guide available on the main RIS Mall page.