

MESA Software

Installation Guide

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

Version 6.7.1
June 12, 2003

1	Introduction.....	1
1.1	Target Directory.....	1
1.2	Storage Directory.....	1
1.3	Disk Space.....	2
1.4	Uninstalling Previous Versions.....	2
2	MESA Internal Software Packages.....	4
2.1	MIR Central Test Node (CTN) Software.....	4
2.2	HL7IMEXA.....	4
2.3	PostgreSQL.....	4
2.4	Microsoft SQL Server.....	4
2.5	Adaptive Communication Environment (ACE).....	5
2.6	Xerces C.....	5
2.7	CGIHTML.....	5
3	Installation from Compiled Binaries.....	6
3.1	Database Installation.....	6
3.2	Perl Installation.....	8
3.3	Apache Installation.....	8
3.4	Install make.....	10
3.5	Install Java.....	10
3.6	MESA Installation.....	11
4	Installation from Source Code: Unix.....	12
4.1	Allocate Disk Space for Installation.....	12
4.2	Build the PostgreSQL System.....	13
4.3	Perl Installation.....	13
4.4	Build the Apache Web Server.....	13
4.5	Install the Sun Java Development Kit.....	13
4.6	Install the Xerces-C Software.....	13
4.7	Build the MESA Software.....	14
4.8	Build Secure Library and Applications.....	17
5	Install from Source Code: Windows NT, 2000 Server.....	19
5.1	Allocate Disk Space for Installation.....	19
5.2	Install SQL Server.....	19
5.3	Perl Installation.....	19
5.4	Build the Apache Web Server.....	19
5.5	Install the Sun Java Development Kit.....	20

5.6	Install the Xerces –C software	20
5.7	Build Environment.....	20
5.8	Build the MESA Software	20
5.9	Build Secure Library and Applications.....	24
6	Build Databases	26
6.1	Build the MESA Databases: Unix	26
6.2	Build the MESA Databases: Windows NT, 2000 Server	26
7	Install the MESA Storage Directory	27
7.1	Install the MESA Storage Directory: Unix	27
7.2	Install the MESA Storage Directory: Windows NT, 2000 Server.....	27
8	Runtime Notes	28
8.1	Runtime Notes: Unix	28
8.2	Runtime Notes: Windows NT, 2000 Server	28
9	Sample Installations	29
9.1	PostgreSQL Installation	29
10	Solaris Notes	30
10.1	Patches for Solaris 2.6.....	30

1 Introduction

This document describes the installation procedure for MESA software. The software is distributed as compiled binaries for select Unix platforms and Windows NT 4.0 systems. The software is also distributed with all source code for compiling on Unix and Windows NT systems.

The MESA package is written in C++ and Java and makes use of other packages that are written in C and C++. Section 2 of this document will describe the software packages that are part of MESA. Section 2 is an informative section and contains no instructions for building the software.

Section 3 of this document describes how to install the software from the pre-compiled binaries. Users who want or need to compile the software will find the complete compile/build instructions in section 4 for Unix systems and section 5 for Windows NT systems.

After you have installed the software using the instructions in section 3, 4 or 5, carry out the instructions in section 6. This will describe how to create database tables and files used to run the simulators.

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 and `D:\mesa` for Windows NT systems. The default can be changed during the build procedure as described in sections 4 and 5. 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 1.3). The location of the build directory is not important.

If you install the software using the binary installation, you may use a different target directory than the those mentioned above. You will need to configure the runtime environment as described in section 7.

The target directory contains these subdirectories:

bin	The compiled executable programs
data	
db	Files for controlling database operations
lib	Shared libraries loaded at runtime
logs	Logfiles for all MESA server applications
runtime	Configuration files for applications
testsuite	Scripts/data for test protocol

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 defaults for this is `/opt/mesa/storage` and `D:\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; 100 slices of CT data require much more space than one small Nuclear Medicine study.

1.3 Disk Space

Users installing the MESA software will need to be aware of the disk requirements for the system. On Solaris systems, you may need to install additional tools to build the software. The build environment should be present with Red Hat Linux 6.0 systems. The table below summarizes the disk requirements for various parts of 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 2.6	Postgres	5
Solaris 2.6	Java JDK	
Solaris 2.6	MESA Installation in target directory	180
Solaris 2.6	ACE software when compiled (can be deleted after MESA installation)	175
Solaris 2.6	MESA Source Code when compiled (can be deleted after installation)	350
Linux 6.0	Java JDK	
Linux 6.0	MESA Installation in target directory	100
Linux 6.0	MESA Source Code when compiled (can be deleted after installation)	100
Linux 6.0	MESA Source Code when compiled (can be deleted after installation)	250
Windows NT 4.0, 2000 Server	Perl	30
Windows NT 4.0, 2000 Server	SQL Server	200
Windows NT 4.0, 2000 Server	Java JDK	55
Windows NT 4.0, 2000 Server	MESA Installation in target directory	30
Windows NT 4.0, 2000 Server	MESA Source Code when compiled (can be deleted after installation)	175

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:

Unix	Windows NT 4.0, 2000 Server
cd \$MESA_TARGET/db	D: cd %MESA_TARGET%\db
make uninstall	perl drop_mesa_tables.pl <sa login> <sa password> <server>
cd \$MESA_TARGET	
rm -r *	Delete all directories in %MESA_TARGET%

The first two steps remove existing MESA databases. The parameters <sa login> and <sa password> for the Windows instructions refer to the system administrator account for SQL Server. The <server> parameter is passed to the isql command and identifies the server name. (In our institution, that can be TBALL or K1\MESA depending on which system is used for testing.) The last two steps remove the installed binaries, libraries and data files.

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. Sections 4 and 5 will describe other software packages (compilers) used to build MESA.

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 Microsoft SQL Server

This relational database is a commercial product from the Microsoft Corporation. This is the database used by the MESA tools for Windows NT 4.0 and 2000 Server installations. This is a licensed product from Microsoft and requires the user to obtain the appropriate software and licenses. This software is not provided with the MESA tools and must be purchased separately. The MESA tools support SQL Server 7.0 under Windows NT 4.0 and SQL Server 2000 under Windows 2000 Server.

2.5 Adaptive Communication Environment (ACE)

This software is no longer packaged with the MESA tools.

The Adaptive Communication Environment (ACE) provides a number of wrappers for operating system functions and a significant middle-ware implementation that is useful for network implementations. ACE is Open Source Software that is developed at Washington University and UC Irvine and is supported by a number of commercial contracts.

2.6 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

2.7 CGIHTML

CGIHTML is a library written in C that supports CGI applications. This library is written by Eugene Eric Kim (EEKIM@EEKIM.COM). This software is planned for future versions of MESA releases and is not required for this release.

3 Installation from Compiled Binaries

3.1 Database Installation

3.1.1 PostgreSQL Installation: Unix systems

Red Hat Linux 6.0 ships with PostgreSQL as an installation option. Please install this version of PostgreSQL (6.4.2).

Solaris does not come with this software. Please install the version 6.4.2 of PostgreSQL found on the MESA distribution web page.

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 (for Solaris) 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, cd to the directory */opt*.
4. We have supplied tar files with the PostgreSQL system compiled for Solaris 2.6. Identify the appropriate file and untar the contents. For example:

```
tar xf postgres642_sol2.6.tar
```

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

```
chown -R postgres:postgres /opt/postgres
```
6. 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:

```
initdb
```

You should only have to do this one time; this is done before you start the postgres server.
7. Log in under the *postgres* account and start the server application. Remember to use the environment variables listed in Table 3.1-1.

```
postmaster &
```

Table 3.1-1 Environment Variables for Postgres Account

System	Variable	Contains
Solaris	path	/opt/postgres/bin
	LD_LIBRARY_PATH	/opt/postgres/lib
	PGLIB	/opt/postgres/lib
	PGDATA	/opt/postgres/data
Red Hat Linux 6.0	path	should already be correct
	PGLIB	/usr/lib/pgsql
	PGDATA	/var/lib/pgsql

3.1.2 Microsoft SQL Server Installation: Windows NT 4.0, Windows 2000 Server

The MESA software is tested with Microsoft SQL Server 7.0 and SQL Server 2000. The MS SQL Server product provides a setup tool that guides you through the installation process. When the installation is complete, you will be able to interact with the server using interactive SQL and their GUI-based management tools. The MESA software communicates with the server using ODBC, and your PC must be configured to use that channel.

Before configuring the ODBC driver, you need to add a login to the Microsoft SQL Server system. This login is the login/password used by our software to connect to tables. You could use an existing account; the example we use is the account *ctn* with the password *ctn*. This is an account managed by SQL Server and not Windows. Use the SQL Server Enterprise Manager to add this new login (*ctn/ctn* if you follow our example).

Open the Control Panel folder and then open ODBC. For 2000 Server, you will find the ODBC control under Administrative Tools in the Control Panel. Select the "System DSN" tab. There will be a list of system data sources. There should be one called LocalServer which uses the SQL Server driver. This entry is created by the SQL Server installation procedure. Our server is configured with these values (set by the SQL Server setup program). You can choose a different login ID and password. The runtime notes will tell you how to communicate your values to the MESA software.

Variable	Value
Name	LocalServer
Description	<blank>
Which server	local
Verify authenticity by ...	SQL Server authentication
Login ID	ctn
Password	ctn

The MESA software runs on the same machine as the SQL Server, so we use the local connection. We do not use the trusted connection option, but you might decided to do so depending on how you want to configure your system.

The database software uses the environment variable `SQL_ACCESS` to determine the 3 parameters needed to establish the connection to the SQL database. The format of this variable is:

```
<server name>:<login>:<password>
```

We use this at our site: `LocalServer:ctn:ctn`

If you do not define this environment variable, the `LocalServer:ctn:ctn` values are used by default. You might choose to use a different login name or password or a different scheme. If you choose to use the Trusted Server feature of the system, you can leave the login and password values blank in the `SQL_ACCESS` variable: `"LocalServer::"`.

The SQL Server has security features that allow the administrator to restrict access to tables in the database. You may find that you need to open up access to get the MESA software to operate properly. If the MESA applications complain about access privileges, you will need to use the SQL Enterprise Manager to give you access rights to the databases (insert, delete). There are several methods for allowing access. One simple method is to activate the Manage pulldown (in SQL Enterprise Manager) and select logins. For the login that you are using, alias that login as `dbo` (stands for database owner) for the databases you are using. That should give you the privileges you need. We also suggest you read the SQL Server documents to understand their security features (they will certainly explain them better than we can).

3.2 Perl Installation

3.2.1 Perl Installation: Unix

Perl is shipped with Linux and requires no installation.

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

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

3.2.2 Perl Installation: Windows NT 4.0, 2000 Server

Perl is available for Win32 systems at <http://www.activestate.com>. We are using ActivePerl 5.6 on our systems. The Active State distribution comes in two versions. One requires the MSI installer and the other version has no uninstall feature. If either version is unacceptable to you, you can build the perl from source. Use the stable version (5.6 as of this document) that can be found at www.perl.org. We have installed this version in `D:\perl`.

The MESA distribution page contains a zip file with a version of perl that we compiled in our laboratory. You may also install this version of perl (5.6) by unzipping it in the appropriate directory.

3.3 Apache Installation

Not required for 6.x release of MESA software. Skip all of section 3.3.

Apache installation is completed in two steps. In the first step, you install an Apache web server without the MESA configuration and web files. This step would be necessary for Solaris systems but not for Linux systems (which should ship with an Apache web server as an

installation option). In the second step, you will add the configuration and web files to the installed server software.

3.3.1 Install Web Server

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

```
pkgadd -d pkgs/solaris/apache-1_3_3-sol26-sparc-local
```

3.3.2 Add Configuration and Web Files

1. Find the directory where the Apache server is installed. With the Solaris package described in 3.2.1, this is `/usr/local/apache`. With Red Hat Linux, the Apache installation directory is `xxx`.
2. As root, cd to the Apache installation directory:
 - `cd /usr/local/apache`
3. We have supplied tar files with the Apache configuration and web files prepared for Solaris 2.6 and Red Hat Linux 5.2 systems. Identify the appropriate file and untar the contents. For example:
 - `tar xf pkgs/apache133_sol2.6.tar`
4. The web server is started from the root account but will run as `nobody`. Start the web server:
 - `/usr/local/bin/apache/startserver &`

3.4 Install make

Some versions of Solaris may ship without make. Some of the post-install procedures require use of the make facility. A Solaris software installation package for gnu make is available at <http://www.sunfreeware.com> and is available through the MESA software web page. To install the software in `/usr/local/bin`, perform this step as root:

```
pkgadd -d make-3_76_1-sol26-sparc-local
```

3.5 Install Java

Starting with MESA Version 4.0, some user interfaces are supplied that are written in Java. These use the Java Swing classes, nominally available with Java version 1.2.

3.5.1 Install Java: Unix

We are using version 1.2.2 in our laboratory for both Solaris and Red Hat Linux. Obtain the Java Development Kit (JDK) version 1.2.2 from the Sun Java web site (<http://www.java.sun.com>) and install according to directions. We typically install in `/opt/jdk1.2.2`. Some versions of Solaris may come with Java installed, but this will be an earlier version without support for the Swing classes. You can install the newer version and have it coexist with the original version.

If you are installing from a binary installation, you will likely be able to install the Sun Java Runtime Environment (JRE) rather than the SDK. We have not tested that.

3.5.2 Install Java: Windows NT 4.0, 2000 Server

We are using version 1.3.0 in our laboratory for Windows NT systems. Obtain the Java Development Kit (JDK) version 1.3.0 from the Sun Java web site (<http://www.java.sun.com>) and install according to directions. We typically install in `D:\jdk1.3`.

3.6 MESA Installation

3.6.1 MESA Installation: Unix

The MESA software is run from a normal user account. Because you need to create a *postgres* account to manage the PostgreSQL software, you might choose to use the same account for the MESA software. We will assume you choose the user name *mesa* to operate the MESA software.

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 *mesa* account. The group is not important.
2. The path for the mesa account should include */opt/mesa/bin* and the path for postgres.
3. If your system uses a path to describe directories for runtime libraries, that path should include */opt/mesa/lib* and the path to Postgres shared libraries (*/opt/postgres/lib* on Solaris).
4. Create a directory */opt/mesa* owned by the mesa account.
5. Login under the mesa account and untar the MESA software:

- `cd /opt/mesa`
- `zcat mesa-sol2.6.tar.Z | tar xf -`

3.6.2 MESA Installation: Windows NT 4.0, 2000 Server

The 6.0.0 version of MESA software is distributed as a self-extracting zip file. This software can be installed using a normal account with no system privileges. All software is installed in the target directory (*%MESA_TARGET%*) and no files are installed in any system directories. No changes are made to the Registry.

Execute the self extracting archive for the MESA binary distribution. The software provides a default installation directory: *D:\mesa*. You may choose a different directory (and/or partition); just be sure to adjust the direct names given below and to set the runtime environment when you use the software.

4 Installation from Source Code: Unix

The MESA software is distributed with all source code and can be compiled on Unix systems. Building the software requires a proper build environment. Section 4.1 of this document will describe the build environment. Section 4.2 will describe the steps for building the MESA software.

MESA is compiled on Unix systems using gnu tools that are available without royalty and using the Sun Java Development Kit (JDK). The gnu tools can be retrieved from <ftp://prep.ai.mit.edu>. We have collected the source code from the MIT site and include the gnu packages on the MESA distribution web page.

Pre-built gnu tools are available for Sun Solaris users at <http://www.sunfreeware.com>. To install one of these tools you would use the `pkgadd` application that comes with Solaris. For example: `pkgadd -d pkgs/<gnu package>`.

The table below lists the packages we use and the order in which they are installed. As mentioned above, Solaris users can install these packages without a build step. Users on other Unix systems will have to refer to the installation instructions that come with the gnu software.

Package	ftp Site	Version
gnu make	ftp://prep.ai.mit.edu/gnu/make	3.76.1 or higher
gnu cc	ftp://prep.ai.mit.edu/gnu/gcc	2.95.3 (install both cc and g++)
gnu flex	ftp://prep.ai.mit.edu/gnu/flex	2.5.4a
Jave Development Kit	http://www.java.sun.com	1.2.2

It is possible that other compiler systems can be used to build the MESA software, but we have only tested with the gnu tools.

Note: A user on an SGI machine reports that he had to use a gnu install package rather than the default *install* that came with his machine. If using that install package is not an option, you can try changing the `INSTALL` macro in the environment files to `cp` rather than *install*. The gnu install is available from <http://freeware.sgi.com> in the *fileutils-3.16* package.

4.1 Allocate Disk Space for Installation

The MESA software requires space to build the software as well as space for the installation. After the software has been compiled and installed, you may delete the source code to recover space. The table below gives estimates on the space needed during various parts of the installation procedure. Some of the entries will make more sense as you read the rest of the procedure.

OS	Software	Notes	Estimate (MB)
	ACE	Uncompiled source	40
Linux	ACE	Compiled	80
Solaris	ACE	Compiled	140
Solaris	PostgreSQL	Installed	10
	MESA	Uncompiled source	20
Linux	MESA	Compiled (w/o ACE)	250
Linux	MESA	Installed (\$MESA_TARGET)	150
Solaris	MESA	Compiled (w/o ACE)	450
Solaris	MESA	Installed (\$MESA_TARGET)	400

4.2 Build the PostgreSQL System

Obtain the PostgreSQL release from the PostgreSQL home page (<http://www.postgresql.org>) or use the release included with MESA. Installation instructions are found at the PostgreSQL home page. The numeric values you choose for the postgres user account and postgres group are not significant. We are using version 6.4.2 of the PostgreSQL software. MESA assumes the software is installed in `/opt/postgres` on Solaris systems.

Red Hat Linux 6.0 that we use comes with PostgreSQL as an installation option. Install that software either during or after installation of the operating system.

4.3 Perl Installation

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.

4.4 Build the Apache Web Server

Not required for 6.x release of MESA software.

Obtain the Apache web server from the apache home page (<http://www.apache.org>) or use the release included with MESA. Installation instructions are found at the Apache home page. We use version 1.3.3 of the software. MESA assumes the software is installed in `/usr/local/apache`.

4.5 Install the Sun Java Development Kit

Read section 3.6 of this document. Because you are building the software from source, you will need the JDK rather than the JRE.

4.6 Install the Xerces-C Software

Obtain the xercesc release from the apache home page (<http://www.apache.org>). We use version 1.7.0. This software should be compiled in some directory that is accessible to the MESA source tree. You will want to compile this software one time (not in the MESA source tree) and then form a link to the MESA tree.

Installation instructions are available on the apache web page. We install the software in `/opt/xerces-c-src1_7_0`. The configuration scripts that we run are (Solaris):

```
./runConfigure -psolaris -gcc -xg++ -nsocket -tnative -rpthread (or for  
Linux)
```

```
./runConfigure -plinux -gcc -xg++ -minmem -nsocket -tnative -rpthread
```

When the software has been built, make a soft link to the MESA software. Make sure you use the names as specified below:

```
ln -s $XERCESSROOT $MESA_ROOT/external/xerces-c
```


4.7 Build the MESA Software

4.7.1 Extract the MESA Software

The MESA software is distributed in a tar file whose name will identify the version. For example, version 6.0.0 of the software is *mesa_6.0.0.tar.Z*. You will extract the software and build it from a normal user account. Because you need to create a postgres account to manage the PostgreSQL software, you might choose to use the same account for the MESA software. We will use a different account with the user name *mesa* to operate the MESA software.

The directory which serves as the root for the MESA source code is not important. Select a root directory and untar the MESA distribution:

- `zcat mesa_6.0.0.tar.Z | tar xf -`

You will see that the distribution contains several directories at the top level. These are described in the table below.

Directory	Description
apps	MESA applications
bin	Alternate installation directory
data	Reserved
dbscripts	Scripts to control database operations
env	Build environment for supported platforms
etc	Various templates, configuration files
external	External libraries (included, but not part of MESA development)
include	Master include directory for MESA packages.
javamesa	Java classes that support MESA operations
java_apps	Java applications
lib	Directory for compiled libraries
libsrc	Source code for the various MESA libraries
external	External software packages included with MESA
runtime	Configuration files for runtime options
scripts	Shell scripts for building the system
testapps	
testdata	The test data and scripts for the simulator tests

4.7.2 Create the MESA Target Directory

The default MESA target directory is /opt/mesa. Create the target directory using the *mesa* account. You may choose to use a different target directory by changing the environment file for your system. Please refer to section 4.6.4 and the environment variable MESA_TARGET described in table 4.6-2.

4.7.3 Build the ACE Software

The ACE software is no longer used in the MESA distribution. Skip this section.

The Adaptive Communication Environment (ACE) is a large, mature middle-ware package that MESA uses for some network operations. We are using version 5.1.10. Users familiar with ACE are free to substitute newer versions at their discretion.

MESA assumes that the ACE software is located in the MESA software tree (\$MESA_ROOT/external/ACE_wrappers). Our advice is to build the ACE software in a separate directory of your choice (xxx/yyy/ACE_wrappers) and then create a soft-link from the ACE directory to *external/ACE-wrappers*. The ACE software will probably not change with future release of MESA, so this will allow you to skip the step of rebuilding ACE with new versions of MESA.

Untar the ACE software (supplied as a separate tar file). As discussed above, untar the ACE software directly in *external* or untar it in another directory and create the proper soft-link to *external/ACE_wrappers*. ACE has its own installation instructions detailed in *ACE_wrappers/ACE-INSTALL.html*.

Some of the ACE installation steps discuss creating configuration files to match your operating system and compiler. We list the configuration files we used when building ACE in Table 4.6-1 below.

Table 4.7-1 ACE Configuration Files

OS	ACE Directory	ACE File	Linked to
Solaris	include/makeinclude	platform_macros.GNU	platform_sunos5_g++.GNU
	ace	config.h	config-sunos5.6.h
Linux	include/makeinclude	platform_macros.GNU	platform_linux_lxpthread.GNU
	ace	config.h	config-linux-lxpthread.h

You need only create the ACE library. This is accomplished by typing (gnu) make in the directory `ACE_wrappers/ace` after configuration is complete. The run-time library will be installed in a separate step. Remember to set the environment variable `ACE_ROOT` before you try to build the ACE library.

4.7.4 Create the MESA Configuration Environment

The `env` directory contains several directories. Each directory is targeted to a specific operating system (solaris, linux). In the OS-specific directories are triplets of files that define the compilation environment. Examples are:

<code>solaris.2.x.psql.gcc.noopt.env</code>	<code>make.solaris.2.x.psql.gcc.noopt</code>	<code>ctnmake.solaris.2.x.psql.gcc.noopt</code>
<code>linux.6.x.psql.gcc.noopt.env</code>	<code>make.linux.6.x.psql.gcc.noopt</code>	<code>ctnmake.linux.6.x.psql.gcc.noopt</code>
<code>linux.sparc.6.x.psql.gcc.noopt.env</code>	<code>make.linux.sparc.6.x.psql.gcc.noopt</code>	<code>ctnmake.linux.sparc.6.x.psql.gcc.noopt</code>

`make.os.options` is a file which is termed the *global Makfile*. This file is included by all other Makefiles when compiling the subroutine libraries and applications. It defines a global set of rules to be used during the make process. The best approach is to examine the existing files and modify them to fit your environment.

The second *global Makefile* (`ctnmake.os.options`) is used to control the build and installation of the CTN software. We include and use this file with the MESA distribution so that the CTN software can be configured and installed at the same time as the MESA software.

One example of an environment file is

```
env/solaris/solaris.2.x.psql.gcc.noopt.env. This file contains setenv
commands for the csh and should be used by users of the csh or tcsh as follows:
    source env/solaris/solaris.2.x.psql.gcc.noopt.env
```

If you wish to use a different shell, you may need to alter the syntax in the file and how you use the file.

The environment files contain variables that define path names for files and certain target directories. Some of the paths are relative to the root of the installation directory while others are relative to the root of the target directory. Table 4.6-2 below summarizes the environment variables set in the environment file.

Table 4.7-2 MESA Build Environment Variables

Environment Variable	Definition
----------------------	------------

MESA_MAKE	The pathname to the global makefile used when the MESA software is compiled.
MESA_ROOT	The root directory for MESA source code. This will be important for compilation, but not for run-time operations.
MESA_TARGET	The root directory for MESA installation. Default value is <code>/opt/mesa</code> . You may choose a different root by altering this variable.
DICOM_ROOT	The root directory for the CTN installation. Other CTN directories are defined from this point.
DICOM_BIN	The location of the compiled binaries for the system. For MESA, the default is <code>/opt/mesa/bin</code> .
DICOM_LIB	The location of compiled CTN libraries.
DICOM_INCLUDE	The location of the CTN common include files for the subroutine libraries.
DICOM_MAKE	The pathname to the global makefile used when the CTN software is compiled. This file is shared by the CTN and MESA software packages.
JDK_ROOT	The pathname to the installation directory for the Java Development Kit. We install in the directory <code>/opt/jdk1.2.2</code> .

4.7.5 Install the MESA Software

The MESA software includes a series of Makefiles that will build and install the MESA software. Developers who modify the software can build portions of MESA using a subset of the Makefiles. Users who want to build and install the software as distributed can do this from the *mesa* account:

1. `cd <MESA root directory>`
2. `source env/os/os-specific.env`
3. `make install`

The make system will build and install the MESA software in the target directory. The build process will stop if any errors are encountered. The build system should complete these steps:

1. Install the ACE software which was compiled in a previous step.
2. Compile/install HL7IMEXA software (HL7 library).
3. Compile/install MIR CTN software (DICOM library and applications).
4. Compile/install MIR JAVACTN software (support classes)
5. Compile/install MESA class libraries.
6. Compile/install JAVAMESA (support classes)
7. Compile/install MESA applications.
8. Compile/install Java applications.
9. Install test data and test scripts.
10. Install database scripts.

4.8 Build Secure Library and Applications

The MESA software uses the OpenSSL library to implement the TLS protocol. Several MESA applications are built with separate build instructions. You need build this part of the software only if you require those applications and you are granted access to the source code. Access to the source code depends on US export restrictions for software with encryption.

4.8.1 Install the OpenSSL Library

Obtain the OpenSSL library from www.openssl.org. We are using version 0.96.d. For Unix installation, the location of the source for the OpenSSL software is irrelevant; the MESA tools access the software after it is installed in its final destination.

Configure the OpenSSL software according to the instructions with that package. We configure the software so that the target directory is `/usr/local/openssl`. If you choose a different directory, you will need to modify the `$MESA_MAKE` file which expects that directory for the software.

To satisfy export regulations, we disable a number of ciphers. As of this release, no encryption is included. Therefore, the configuration specification is:

```
./config --prefix=/usr/local/openssl --  
openssldir=/usr/local/openssl no-bf no-case no-des no-dh no-  
dsa no-idea no-mdc2 no-rc2 no-rc4 no-rc5
```

Compile and install the software. To test the ciphers that are included, use the `openssl` command to list the supported ciphers:

```
openssl ciphers
```

You should see `NULL-SHA:NULL-MD5`

4.8.2 Install the MESA Secure Software

The MESA Secure Software uses the same mechanism as the standard MESA software. You need to install the standard software first (to build the libraries). Using the same environment file:

1. `cd <MESA root directory>`
2. `source env/os/os-specific.env`
3. `make install_secure`

If you need to rebuild the MESA Secure Software, use this command:

```
make rebuild_secure
```

5 Install from Source Code: Windows NT, 2000 Server

The MESA software is distributed with all source code and can be compiled on Windows NT 4.0 systems. Building the software requires a proper build environment. Section 5.6 of this document will describe the build environment. Section 5.7 will describe the steps for building the MESA software.

MESA is compiled on NT systems using Microsoft Visual C++ (MSVC++) and using the Sun Java Development Kit (JDK).

The table below lists the packages we use and the order in which they are installed.

Package	ftp Site	Version
MSVC++	Purchase from Microsoft	6.0
Microsoft SQL Server	Purchase from Microsoft	7.0
Java Development Kit	http://www.java.sun.com	1.3.0
Perl	http://www.activestate.com	5.6.0

It is possible that other compiler systems can be used to build the MESA software, but we have only tested with these tools.

5.1 Allocate Disk Space for Installation

The MESA software requires space to build the software as well as space for the installation. After the software has been compiled and installed, you may delete the source code to recover space. The table below gives estimates on the space needed during various parts of the installation procedure. Some of the entries will make more sense as you read the rest of the procedure.

OS	Software	Notes	Estimate (MB)
	ACE	Uncompiled source	40
NT	ACE	Compiled	160
	MESA	Uncompiled source	25
NT	MESA	Compiled (w/o ACE)	250
NT	MESA	Installed (\$MESA_TARGET)	30

5.2 Install SQL Server

Read the instructions provided in section 3.1.2 of this document.

5.3 Perl Installation

Read the instructions provided in section 3.2.2 of this document.

5.4 Build the Apache Web Server

Not required for 6.x release of MESA software.

Obtain the Apache web server from the apache home page (<http://www.apache.org>) or use the release included with MESA. Installation instructions are found at the Apache home page. We use version 1.3.3 of the software. MESA assumes the software is installed in.

5.5 Install the Sun Java Development Kit

Read section 3.6 of this document. Because you are building the software from source, you will need the JDK rather than the JRE.

5.6 Install the Xerces –C software

We are using version 1.7.0 of the Xerces-C software. Obtain this version of the software (version 2.x is currently not supported) from the apache web site (<http://xml.apache.org/xerces-c/>) and deposit the source code in a directory that is at the same level as the MESA distribution. The folder created by the distribution will have the version number in it (e.g. *xerces-c-src1_7_0*). Build the library (XercesLib, both Release and Debug) according to the installation instructions at the Apache web site.

5.7 Build Environment

The MESA software is compiled and installed using the *nmake* program supplied with the MSVC++ 6.0 development environment. The make files distributed with the MESA software compile and install the software. The software can be compiled from the MSVC++ integrated development environment, but that will not install the software and test scripts.

The *nmake* program will be invoked from a DOS command line window. In order for the build to work, your account needs to be configured as described below. The descriptions below assume that MSVC++, Perl and the Java JDK are assumed on the D: drive. If you install on a different drive, modify the paths as needed.

1. The environment variable *include* needs to contain the directories listed below:
 - a. D:\Program Files\Microsoft Visual Studio\VC98\atl\include
 - b. D:\Program Files\Microsoft Visual Studio\VC98\mfc\include
 - c. D:\Program Files\Microsoft Visual Studio\VC98\include
 - d. D:\jdk1.3\include
 - e. D:\jdk1.3\include\win32j
2. The environment variable *lib* needs to contain the directories listed below:
 - a. D:\Program Files\Microsoft Visual Studio\VC98\mfc\lib
 - b. D:\Program Files\Microsoft Visual Studio\VC98\lib
3. The environment variable Path needs to include the directories listed below:
 - a. D:\Perl\bin
 - b. D:\Program Files\Microsoft Visual Studio\Common\Tools\WinNT
 - c. D:\Program Files\Microsoft Visual Studio\Common\MSDev98\Bin
 - d. D:\Program Files\Microsoft Visual Studio\Common\Tools
 - e. D:\Program Files\Microsoft Visual Studio\VC98\bin
 - f. D:\jdk1.3\bin

5.8 Build the MESA Software

5.8.1 Extract the MESA Software

The MESA software is distributed in a tar file whose name will identify the version. For example, version 6.0.0 of the software is *mesa_6.0.0.tar.Z* or *mesa_6.0.0.zip*.

The directory which serves as the root for the MESA source code is not important; we use *D:\projects\mesa*. Select a root directory and extract the MESA distribution. You will see that the distribution contains several directories at the top level. These are described in the table below.

Directory	Description
apps	MESA applications
bin	Alternate installation directory
data	Reserved
dbscripts	Scripts to control database operations
env	Build environment for supported platforms
etc	Various templates, configuration files
external	External libraries (included, but not part of MESA development)
include	Master include directory for MESA packages.
javamesa	Java classes that support MESA operations
java_apps	Java applications
lib	Directory for compiled libraries
libsrc	Source code for the various MESA libraries
external	External software packages included with MESA
runtime	Configuration files for runtime options
scripts	Shell scripts for building the system
testapps	
testdata	The test data and scripts for the simulator tests

5.8.2 Create the MESA Target Directory

The default MESA target directory is D:\mesa. Create the target directory. You may choose to use a different target directory by changing the environment file for your system. Please refer to section 5.8.4 and the environment variable MESA_TARGET described in table 5.8-1.

5.8.3 Build the ACE Software

The ACE software is no longer used in the MESA tools. You should ignore this section.

The Adaptive Communication Environment (ACE) is a large, mature middle-ware package that MESA uses for some network operations. We are using version 5.1.10. Users familiar with ACE are free to substitute newer versions at their discretion.

The Win32 installation of MESA assumes the ACE software is located in a directory at the same level as the MESA distribution. Therefore, our projects directory has these three subdirectories:

- mesa
- ACE_wrappers
- xerces-c-src1_7_0

Build the ACE software in the ACE_wrappers directory or unzip the version provided on the MESA distribution web page. It must reside in this parallel directory. The ACE software will probably not change with future release of MESA, so this will allow you to skip the step of rebuilding ACE with new versions of MESA. ACE has its own installation instructions detailed in ACE_wrappers/ACE-INSTALL.html.

You need only create the ACE library (using the MSVC++ IDE).

5.8.4 Create the MESA Configuration Environment

The *env* directory contains several directories. Each directory is targeted to a specific operating system (solaris, linux). In the OS-specific directories are triplets of files that define the compilation environment for Unix systems. For Windows NT systems, a single batch file is used to define required environment variables. This file is *env\w32\w32_env.bat*.

The environment files contain variables that define path names for files and certain target directories. Some of the paths are relative to the root of the installation directory while others are relative to the root of the target directory. Table 5.7-1 below summarizes the environment variables set in the environment file.

Table 5.8-1 MESA Build Environment Variables

Environment Variable	Definition
CLASSPATH	Path name needed to build Java modules. During the build process, this includes two directories. For runtime operations, the CLASS_PATH will be different.
DICOM_ROOT	Points to the top of the MIR CTN source code distributed with the MESA tools.
JAR_DIRECTORY	Target directory for installing Java jar files built during install process. Must be %MESA_TARGET%\lib to match runtime scripts.
JDK_ROOT	The pathname to the installation directory for the Java Development Kit. We install in the directory D:\jdk1.3, but you may choose a different install directory.
MESA_BIN	Target directory for installing MESA binaries and dlls. Must be %MESA_TARGET%\bin to match runtime scripts.
MESA_ROOT	Points to the top of the MESA source code distribution.
MESA_TARGET	The root directory for MESA installation. Default value is D:\mesa. You may choose a different root by altering this variable.

5.8.5 Install the MESA Software

The MESA software includes a series of Makefiles that will build and install the MESA software. Developers who modify the software can build portions of MESA using a subset of the Makefiles. Users who want to build and install the software as distributed can do this from the *mesa* account:

1. cd <MESA root directory>
2. env\w32\w32_env.bat
3. nmake/f w32_install.mak install

The make system will build and install the MESA software in the target directory. The build process will stop if any errors are encountered. The build system should complete these steps:

1. Install the ACE software which was compiled in a previous step.
2. Compile/install HL7IMEXA software (HL7 library).
3. Compile/install MIR CTN software (DICOM library and applications).
4. Compile/install MIR JAVACTN software (support classes)
5. Compile/install MESA class libraries.
6. Compile/install JAVAMESA (support classes)
7. Compile/install MESA applications.
8. Compile/install Java applications.

9. Install test data and test scripts.
10. Install database scripts.

5.9 Build Secure Library and Applications

The MESA software uses the OpenSSL library to implement the TLS protocol. Several MESA applications are built with separate build instructions. You need build this part of the software only if you require those applications and you are granted access to the source code. Access to the source code depends on US export restrictions for software with encryption.

5.9.1 Install the OpenSSL Library

Obtain the OpenSSL library from www.openssl.org. We are using version 0.96.d. The MESA build parameters assume this software is installed in a directory parallel to the MESA software and is named openssl-0.9.6d.

Configure the OpenSSL software according to the instructions with that package. To satisfy export regulations, we disable a number of ciphers. As of this release, no encryption is included. Therefore, the configuration specification is:

```
Perl Configure no-bf no-des no-dh no-idea no-mdc2 no-rc2 no-rc4 no-rc5 VC-WIN32
```

Compile and install the software. We are not using assembly language, so our remaining build steps are:

```
ms\do_nt
nmake -f ms\ntdll.mak
```

As the software builds, we found various problems/errors in the build procedure when we disabled ciphers. To fix these, we commented out the code in the following files:

```
crypto/des/read_pwd.c – commented out all code
apps/speed.c – commented out code in application; application now just exits
crypto\ripemd\rmdtest.c – has a test for a Macro, but that failed.
```

You also need to comment out references to functions that are found in ms/LIBEAY32.def.

These are

```
PEM_read_DSA_PUBKEY
PEM_read_bio_DSA_PUBKEY
PEM_write_DSA_PUBKEY
PEM_write_bio_DSA_PUBKEY
d2i_Netscape_RSA
d2i_Netscape_RSA_2
d2i_RSA_NET
d2i_RSA_NET_2
i2d_Netscape_RSA
i2d_RSA_NET
```

To test the ciphers that are included, use the openssl command to list the supported ciphers:

```
openssl ciphers
```

You should see NULL-SHA:NULL-MD5

5.9.2 Install the MESA Secure Software

The MESA Secure Software uses the same mechanism as the standard MESA software. You need to install the standard software first (to build the libraries). Using the same environment file:

4. `cd <MESA root directory>`
5. `source env/os/os-specific.env`
6. `make install_secure`

If you need to rebuild the MESA Secure Software, use this command:

```
make rebuild_secure
```

6 Build Databases

The installation steps defined in sections 3, 4 and 5 install the MESA software in the target directory. You need to complete the steps in this section to build the MESA databases.

6.1 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 6.1-1 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 postgres account to perform a one-time initialization of the postgres system:

- set environment variables as described in Table 5.1-1
- `initdb`

Use the postgres account to start the postgres database server. Make sure the PGLIB and PGDATA environment variables are set. This database server needs to be running for all MESA simulators.

- `postmaster &`

Use the *mesa* account to build the databases:

1. `cd /opt/mesa/db`
2. set the PGUSER environment variable defined in Table 5.1-1
3. `make database`

6.2 Build the MESA Databases: Windows NT, 2000 Server

The installation procedure assumes you will build databases needed for all MESA tests. The first step is to create 10 databases for the various MESA actors/simulators. Use the SQL Enterprise Manager to create the following 10 databases:

adt	imgmgr
ordplc	wkstation
ordfil	rpt_repos
mod1	rpt_manager
mod2	syslog

Each database should be at least 5 MB in size. For each database, add the SQL server account configured in section 5.2 above as a `db_owner`. This will allow you to add and remove tables. Once the databases have been built, you need to create the tables in the databases. Use a DOS window and set the directory to `%MESA_TARGET%\db`. Execute the perl script:

```
perl create_mesa_tables.pl <login> <password> <server>
```

In the example that we use, *login* and *password* are both “ctn”. As mentioned previously, the login that you choose will need *db_owner* rights to create tables in the database. The *server* parameter is passed with the `-S` switch to `isql`; it is the name you have given your SQL server.

7 Install the MESA Storage Directory

A separate directory is maintained for storage of what can be large files (images). We use the environment variable `MESA_STORAGE` to point to this directory. The default value is `/opt/mesa` or `D:\mesa\storage` but you may choose a different directory.

7.1 Install the MESA Storage Directory: Unix

The files in the storage directory are stored in one tar file: `mesa_storage_5.0.0.tar.Z`. (A later version may be available on the distribution server). To create the storage directory:

1. set the environment variable `$MESA_STORAGE`
2. `mkdir $MESA_STORAGE`
3. `cd $MESA_STORAGE`
4. `zcat mesa_storage_5.0.0.tar.Z | tar xf -`

7.2 Install the MESA Storage Directory: Windows NT, 2000 Server

The files in the storage directory are stored in self extracting zip archive: `mesa_storage_5.0.0.exe`. To create the storage directory:

1. Execute the self extracting archive
2. Select the destination for the storage. The default is `D:\mesa\storage`.
3. Remember to set the `MESA_STORAGE` environment variable as described in the runtime section of this document.

8 Runtime Notes

After installation is complete, you should not need to have the MESA source directory to run the test scripts.

8.1 Runtime Notes: Unix

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, 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/jdk1.2.2 (or another value of your choosing)
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
```

8.2 Runtime Notes: Windows NT, 2000 Server

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.

Variable	Value
MESA_TARGET	D:\mesa (or another value of your choosing)
MESA_STORAGE	D:\mesa\storage (or another value of your choosing)
PATH	should include %MESA_TARGET%\bin and the path to perl
JDK_ROOT	D:\jdk1.3 (or another value of your choosing)
JAR_DIRECTORY	%MESA_TARGET%\lib
MESA_SQL_LOGIN	ctn (or other login of your choosing)
MESA_SQL_PASSWORD	ctn (or other password of your choosing)
SQL_ACCESS	LocalServer:ctn:ctn (consistent with values above)
MESA_OS	WINDOWS_NT

9 Sample Installations

This section of the document lists commands and options we used in installing some of the packages. All of the packages we use have complete installation instructions, so you may only need this section if you have problems with the installation.

9.1 PostgreSQL Installation

These steps are performed as the user *postgres*.

```
mkdir /opt/postgres
mkdir /x/p
cd /x/p

tar xf $MESA_ROOT/pkgs/source/postgresql-6_4_2.tar
cd postgresql-6.4.2/src
./configure --prefix=/opt/postgres
/usr/local/bin/make >& make.log
/usr/local/bin/make install >& make.install.log
```


10 Solaris Notes

10.1 Patches for Solaris 2.6

This section is obsolete. The MESA tools are supported for Solaris version 7 and higher. Solaris 2.6 is not supported.

We run this software on a Solaris 2.6 machine with a number of patches. The patches are listed below. We normally install these from a large patch kit that we receive from Sun (not as a number of individual patches).

```
Patch: 105295-01  Obsoletes:  Packages:
Patch: 105642-02  Obsoletes:  Requires:  Incompatibles:  Packages:  SUNWkvm
Patch: 105642-05  Obsoletes:  Requires:  Incompatibles:  Packages:  SUNWkvm
Patch: 105757-01  Obsoletes:  Requires:  Incompatibles:  Packages:  SUNWcsu
Patch: 105755-03  Obsoletes:  Requires:  Incompatibles:  Packages:  SUNWcsu
Patch: 105746-01  Obsoletes:  Requires:  Incompatibles:  Packages:  SUNWcsu
Patch: 105736-01  Obsoletes:  Requires:  Incompatibles:  Packages:  SUNWcsu
Patch: 105724-01  Obsoletes:  Requires:  Incompatibles:  Packages:  SUNWcsu
Patch: 105722-01  Obsoletes:  Requires:  Incompatibles:  Packages:  SUNWcsu
Patch: 105718-02  Obsoletes:  Requires:  Incompatibles:  Packages:  SUNWcsu
Patch: 105705-01  Obsoletes:  Requires:  Incompatibles:  Packages:  SUNWcsu,
SUNWhea
Patch: 105686-02  Obsoletes:  Requires:  Incompatibles:  Packages:  SUNWcsu
Patch: 105621-02  Obsoletes:  Requires:  Incompatibles:  Packages:  SUNWcsu,
SUNWarc
Patch: 105615-02  Obsoletes:  Requires:  Incompatibles:  Packages:  SUNWcsu
Patch: 105568-02  Obsoletes:  Requires:  Incompatibles:  Packages:  SUNWcsu
Patch: 105562-01  Obsoletes:  Requires:  Incompatibles:  Packages:  SUNWcsu,
SUNWnisu
Patch: 105518-01  Obsoletes:  Requires:  Incompatibles:  Packages:  SUNWcsu
Patch: 105516-01  Obsoletes:  Requires:  Incompatibles:  Packages:  SUNWcsu
Patch: 105490-01  Obsoletes:  Requires:  Incompatibles:  Packages:  SUNWcsu,
SUNWcsr, SUNWarc, SUNWbtool, SUNWhea, SUNWtoo, SUNWosdem
Patch: 105405-01  Obsoletes:  Requires:  Incompatibles:  Packages:  SUNWcsu,
SUNWarc
Patch: 105401-04  Obsoletes:  105524-01  Requires:  Incompatibles:  Packages:
SUNWcsu, SUNWarc, SUNWnisu
Patch: 105397-02  Obsoletes:  Requires:  Incompatibles:  Packages:  SUNWcsu
Patch: 105393-01  Obsoletes:  Requires:  Incompatibles:  Packages:  SUNWcsu
Patch: 105216-01  Obsoletes:  Requires:  Incompatibles:  Packages:  SUNWcsu
Patch: 105210-01  Obsoletes:  Requires:  Incompatibles:  Packages:  SUNWcsu,
SUNWarc
Patch: 105401-16  Obsoletes:  105524-01  Requires:  Incompatibles:  Packages:
SUNWcsu, SUNWcsr, SUNWarc, SUNWnisu
Patch: 105181-09  Obsoletes:  105214-01, 105636-01, 105776-01, 106031-02,
106308-01  Requires:  Incompatibles:  Packages:  SUNWcsu, SUNWcsr, SUNWcar,
SUNWhea
Patch: 105562-03  Obsoletes:  Requires:  Incompatibles:  Packages:  SUNWcsu,
SUNWnisu
Patch: 105210-17  Obsoletes:  Requires:  Incompatibles:  Packages:  SUNWcsu,
SUNWarc
```

Patch: 105216-03 Obsoletes: Requires: 105401-07 Incompatibles: Packages: SUNWcsu

Patch: 105621-08 Obsoletes: 105686-02, 105845-01, 106064-01, 106075-01
Requires: Incompatibles: Packages: SUNWcsu, SUNWcsr, SUNWarc, SUNWhea, SUNWnisu

Patch: 105393-07 Obsoletes: 106033-01 Requires: 105621-04 Incompatibles: Packages: SUNWcsu

Patch: 105615-04 Obsoletes: Requires: Incompatibles: Packages: SUNWcsu

Patch: 105665-03 Obsoletes: Requires: Incompatibles: Packages: SUNWcsu

Patch: 106049-01 Obsoletes: Requires: Incompatibles: Packages: SUNWcsu

Patch: 106257-04 Obsoletes: Requires: Incompatibles: Packages: SUNWcsu

Patch: 106271-04 Obsoletes: Requires: Incompatibles: Packages: SUNWcsu, SUNWnisu

Patch: 105755-06 Obsoletes: Requires: Incompatibles: Packages: SUNWcsu

Patch: 106301-01 Obsoletes: Requires: Incompatibles: Packages: SUNWcsu

Patch: 106439-02 Obsoletes: Requires: Incompatibles: Packages: SUNWcsu

Patch: 106448-01 Obsoletes: Requires: Incompatibles: Packages: SUNWcsu

Patch: 105490-05 Obsoletes: Requires: Incompatibles: Packages: SUNWcsu, SUNWcsr, SUNWarc, SUNWbtool, SUNWhea, SUNWtoo, SUNWosdem, SUNWxcu4

Patch: 106226-01 Obsoletes: Requires: Incompatibles: Packages: SUNWcsu

Patch: 105755-07 Obsoletes: Requires: Incompatibles: Packages: SUNWcsu

Patch: 106592-02 Obsoletes: Requires: Incompatibles: Packages: SUNWcsu

Patch: 105181-03 Obsoletes: Requires: Incompatibles: Packages: SUNWcsr, SUNWcar, SUNWhea

Patch: 105847-01 Obsoletes: Requires: Incompatibles: Packages: SUNWcsr

Patch: 105845-01 Obsoletes: Requires: Incompatibles: Packages: SUNWcsr

Patch: 105836-01 Obsoletes: Requires: Incompatibles: Packages: SUNWcsr

Patch: 105797-01 Obsoletes: Requires: Incompatibles: Packages: SUNWcsr

Patch: 105786-01 Obsoletes: Requires: Incompatibles: Packages: SUNWcsr

Patch: 105780-01 Obsoletes: Requires: Incompatibles: Packages: SUNWcsr

Patch: 105778-01 Obsoletes: Requires: Incompatibles: Packages: SUNWcsr

Patch: 105776-01 Obsoletes: Requires: Incompatibles: Packages: SUNWcsr

Patch: 105742-01 Obsoletes: Requires: Incompatibles: Packages: SUNWcsr

Patch: 105720-01 Obsoletes: Requires: Incompatibles: Packages: SUNWcsr

Patch: 105693-01 Obsoletes: Requires: Incompatibles: Packages: SUNWcsr

Patch: 105604-01 Obsoletes: Requires: 105181-02 Incompatibles: Packages: SUNWcsr, SUNWcar, SUNWhea

Patch: 105600-01 Obsoletes: Requires: Incompatibles: Packages: SUNWcsr

Patch: 105572-02 Obsoletes: Requires: Incompatibles: Packages: SUNWcsr

Patch: 105564-02 Obsoletes: Requires: Incompatibles: Packages: SUNWcsr

Patch: 105529-01 Obsoletes: Requires: Incompatibles: Packages: SUNWcsr

Patch: 105528-01 Obsoletes: Requires: Incompatibles: Packages: SUNWcsr

Patch: 105486-01 Obsoletes: Requires: Incompatibles: Packages: SUNWcsr

Patch: 105379-03 Obsoletes: Requires: Incompatibles: Packages: SUNWcsr

Patch: 105214-01 Obsoletes: Requires: Incompatibles: Packages: SUNWcsr

Patch: 105654-03 Obsoletes: Requires: Incompatibles: Packages: SUNWcsr

Patch: 105379-05 Obsoletes: Requires: Incompatibles: Packages: SUNWcsr

Patch: 105786-06 Obsoletes: Requires: Incompatibles: Packages: SUNWcsr

Patch: 105720-06 Obsoletes: Requires: Incompatibles: Packages: SUNWcsr

Patch: 105797-05 Obsoletes: Requires: Incompatibles: Packages: SUNWcsr

Patch: 105600-06 Obsoletes: Requires: 105181-05 Incompatibles: Packages: SUNWcsr, SUNWhea

Patch: 105604-05 Obsoletes: Requires: 105181-05 Incompatibles: Packages: SUNWcsr, SUNWcar, SUNWhea

Patch: 105651-02	Obsoletes:	Requires:	Incompatibles:	Packages: SUNWcar, SUNWhea	
Patch: 105636-01	Obsoletes:	Requires:	Incompatibles:	Packages: SUNWcar	
Patch: 105222-01	Obsoletes:	Requires:	Incompatibles:	Packages: SUNWcar	
Patch: 105633-01	Obsoletes:	Requires:	Incompatibles:	Packages: SUNWxwplt	
Patch: 105618-01	Obsoletes:	Requires:	Incompatibles:	Packages: SUNWxwplt	
Patch: 105400-01	Obsoletes:	Requires:	Incompatibles:	Packages: SUNWxwplt, SUNWplowl	
Patch: 105284-16	Obsoletes:	Requires:	Incompatibles:	Packages: SUNWmfrun	
Patch: 105464-01	Obsoletes:	Requires:	Incompatibles:	Packages: SUNWxwopt, SUNWxwman	
Patch: 105669-02	Obsoletes:	Requires:	Incompatibles:	Packages: SUNWdtbas	
Patch: 105630-01	Obsoletes:	Requires:	Incompatibles:	Packages: SUNWdtbas	
Patch: 105160-01	Obsoletes:	Requires:	Incompatibles:	Packages: SUNWdtbas	
Patch: 105669-04	Obsoletes:	Requires:	Incompatibles:	Packages: SUNWdtbas	
Patch: 106242-01	Obsoletes:	Requires:	Incompatibles:	Packages: SUNWdtbas	
Patch: 105800-01	Obsoletes:	Requires:	Incompatibles:	Packages: SUNWadmap	
Patch: 105800-03	Obsoletes:	Requires: 106125-05	Incompatibles:	Packages: SUNWadmap	
Patch: 106193-03	Obsoletes:	106350-01	Requires:	Incompatibles:	Packages: SUNWadmap
Patch: 105472-01	Obsoletes:	Requires:	Incompatibles:	Packages: SUNWatfsu	
Patch: 105566-01	Obsoletes:	Requires:	Incompatibles:	Packages: SUNWdtdmn	
Patch: 105566-05	Obsoletes:	Requires:	Incompatibles:	Packages: SUNWdtdmn, SUNWdtdst	
Patch: 105497-01	Obsoletes:	Requires:	Incompatibles:	Packages: SUNWoldst	
Patch: 106222-01	Obsoletes:	Requires:	Incompatibles:	Packages: SUNWoldst	
Patch: 105377-03	Obsoletes:	Requires:	Incompatibles:	Packages: SUNWbcp	
Patch: 105492-01	Obsoletes:	Requires:	Incompatibles:	Packages: SUNWcg6	
Patch: 105798-02	Obsoletes:	Requires:	Incompatibles:	Packages: SUNWcpr	
Patch: 105558-01	Obsoletes:	Requires:	Incompatibles:	Packages: SUNWdtdst	
Patch: 105338-04	Obsoletes:	Requires:	Incompatibles:	Packages: SUNWdtdst, SUNWdthev, SUNWdtma	
Patch: 105558-03	Obsoletes:	Requires:	Incompatibles:	Packages: SUNWdtdst	
Patch: 105743-01	Obsoletes:	Requires:	Incompatibles:	Packages: SUNWfns	
Patch: 105795-01	Obsoletes:	Requires:	Incompatibles:	Packages: SUNWhmd, SUNWhmdu	
Patch: 105795-05	Obsoletes:	Requires: 105181-05	Incompatibles:	Packages: SUNWhmd, SUNWhmdu	
Patch: 105362-05	Obsoletes:	Requires:	Incompatibles:	Packages: SUNWm64, SUNWm64cf, SUNWm64w	
Patch: 105362-13	Obsoletes:	Requires:	Incompatibles:	Packages: SUNWm64, SUNWm64cf, SUNWm64w	
Patch: 105552-01	Obsoletes:	Requires:	Incompatibles:	Packages: SUNWnisu	
Patch: 105403-01	Obsoletes:	Requires:	Incompatibles:	Packages: SUNWnisu	
Patch: 105552-02	Obsoletes:	Requires:	Incompatibles:	Packages: SUNWnisu	
Patch: 106235-02	Obsoletes:	Requires:	Incompatibles:	Packages: SUNWpcu, SUNWpsu	
Patch: 105741-02	Obsoletes:	Requires:	Incompatibles:	Packages: SUNWpd	
Patch: 105610-02	Obsoletes:	Requires:	Incompatibles:	Packages: SUNWpd	
Patch: 105580-03	Obsoletes:	Requires:	Incompatibles:	Packages: SUNWpd, SUNWpdu	
Patch: 105741-05	Obsoletes:	Requires:	Incompatibles:	Packages: SUNWpd	
Patch: 105580-08	Obsoletes:	Requires:	Incompatibles:	Packages: SUNWpd, SUNWpdu	

```
Patch: 105637-01 Obsoletes: Requires: Incompatibles: Packages: SUNWpmu
Patch: 106125-05 Obsoletes: Requires: Incompatibles: Packages: SUNWswmt
Patch: 105426-01 Obsoletes: Requires: Incompatibles: Packages: SUNWtnfc
Patch: 106040-10 Obsoletes: 105189-03 Requires: Incompatibles: Packages:
SUNWxim, SUNWxi18n
Patch: 105407-01 Obsoletes: Requires: Incompatibles: Packages: SUNWvolu
Patch: 105189-01 Obsoletes: Requires: Incompatibles: Packages: SUNWxi18n
Patch: 105361-02 Obsoletes: Requires: Incompatibles: Packages: SUNWxilvl
Patch: 105360-08 Obsoletes: Requires: Incompatibles: Packages: SUNWffb,
SUNWffbcf, SUNWffbw, SUNWffbxg
```

11 XML Notes

11.1 Installation of XSLT Processor

The MESA software uses the Xalan XSLT processor from the Apache foundation (xml.apache.org). We include the jar files in the MESA distribution to simplify the installation process for MESA users. This section documents how to get the jar files for users who are installing from scratch or who want newer versions of the Xalan software.

1. Retrieve the Java-2 Xalan software from the apache site:

<http://xml.apache.org/xalan-j/downloads.html>

We use the “second binary distribution” in `xalan-j_2_5_1-bin-2jars.zip` or `xalan-j_2_5_1-bin-2jars.tar.gz`.

2. x
3. x