

EMSYSTEM.DOC

(last update Sep 8, 1989)

This document is a meager attempt to provide basic information concerning the use and setup of the set of electron micrograph image analysis programs running at Purdue University.

TABLE OF CONTENTS

- A. INTRODUCTION**
- B. LOGICAL ASSIGNMENTS**
- C. DOCUMENTATION**
- D. RUNNING PROGRAMS IN THE EM SYSTEM**
- E. BUILDING A SYSTEM**
- F. SOURCE CODE AND OBJECT LIBRARIES**
- G. INTERACTIVE RASTER GRAPHICS**

A. INTRODUCTION

EM is a system of Electron Micrograph image analysis programs, initially patterned after routines developed at the MRC Laboratory of Molecular Biology during the 1960's through 1970's. Substantial modifications have lead to the present system implemented at Purdue University. Images of several types of biological specimens can be analyzed with these programs, including two- and three-dimensional crystals, and icosahedral viruses. Programs for processing helical particles are implemented at Brandeis University, but not at Purdue. In addition, several programs have been designed for building models to help simulate particle images and aid in testing software.

This documentation outlines the system at Purdue which initially was implemented on a VAX 11/750 (8 Mbytes of core memory). It is now running (Summer 1989) on a VAX 8550 (about 6 times faster than the 750) with 32 Mbytes of core memory. The extra speed and memory of the 8550 enabled us to reprogram many segments which now run as much as 100 times faster than the old system on the 750. In some instances portions of the code retain some of the older algorithms, enabling them to run on smaller machines with limited memory. Future changes probably will NOT accommodate such flexibility because it is assumed that most future computers will be even more powerful and flexible than the presently configured 8550 system. Even so, the FORTRAN code is generally simple enough so that it may easily be implemented on smaller VMS systems.

The interactive graphics routines are implemented on a 1280 by 1024 resolution LEXIDATA Model 3400 color raster graphics device. Most of the graphics independent software is easily transportable to another VAX/VMS system. Wherever possible, programs are designed to be graphics device independent. In programs which implement interactive graphics routines, some attempt has been made to keep these routines fairly simple and confined within one or a few separate program segments. The program LX_DRIVER has been designed to handle many of the most common graphics manipulations such as panning, zooming and changing color look-up tables, so these functions may be performed exclusive of all the other application software. A similar program could be written at other sites to accommodate the use of a different graphics device.

The remainder of this document introduces important aspects of the EM system at Purdue and gives suggestions about how the system may be implemented elsewhere.

B. LOGICAL ASSIGNMENTS

A number of standard LOGICAL ASSIGNMENTS are generally made in each user's LOGIN.COM file to facilitate location of EM system files, programs, etc. Users wishing to write new applications software also can then easily access subroutines and object libraries without having to keep personal copies of previously developed, general purpose source code.

Presently at Purdue, each user LOGIN.COM command file contains at least the following six LOGICAL ASSIGNMENTS which identify the directories where object libraries, FORTRAN source code, documentation, executable tasks and command files for building the system are kept.

```
$ ASSIGN [TSB.BLD]      SYS$BLDLIB
$ ASSIGN [TSB.DOC]      SYS$DOCLIB
$ ASSIGN [TSB.EXE]      SYS$EXELIB
$ ASSIGN [TSB.FOR]      SYS$EMLIB
$ ASSIGN [TSB.LEX]      SYS$LXLIB
$ ASSIGN DUA0:          SYS$TMP
```

Thus, any reference made, for example, to SYS\$EMLIB:, is interpreted as [TSB.FOR]. These logical assignment statements frees users from having to globally EDIT command files such as EM*.BLD and also the FORTRAN source which uses INCLUDE files such as EM.CMM, KEYS.CMM and LEXI.CMM (see Section E.). Users outside Purdue have the option of storing EM system files in one or more directories of their own choosing as long as they then make the

appropriate LOGICAL ASSIGNMENTS in their LOGIN.COM files so SYS\$EMLIB, SYS\$BLDLIB, SYS\$DOCLIB, SYS\$EXELIB and SYS\$LXLIB point to the appropriate directories. SYS\$TMP is the 500 Mbyte RA81 system disk drive (DUA0) on the VAX 8550, used mainly for scratch space and system routines and files. Several EM programs make extensive use of this scratch space, using the SYS\$TMP logical assignment. If general user scratch disk space is unavailable, then SYS\$TMP could be used to identify a special user directory, set aside specifically for writing scratch files.

C. DOCUMENTATION

Several levels of documentation are available to help the user understand how to implement the EM system and to help programmers write new applications software and modify existing routines.

1. Descriptions of useful *.DOC files.

SYS\$DOCLIB:EMPROGS.DOC

This file contains single-line descriptions of most of the EM system programs as well as miscellaneous other programs that may be useful. Suggested program pathways are outlined for several different image analysis schemes at the end of this file.

SYS\$DOCLIB:EMFORMATS.DOC

This contains full descriptions of the different file formats used for the four main classes of data (IMAGE, FOURIER TRANSFORM, MAP and STRUCTURE FACTOR) used in the image analysis software.

SYS\$DOCLIB:"PROGRAM".DOC

Documentation concerning the implementation of each program is found in *.DOC files ("PROGRAM".DOC, where "PROGRAM" refers to the root name of the executable task). Program documentation can be accessed via the master driver program, EM, by typing a "-" character followed by the string "PROGRAM" for the desired program. For programs whose name is prefixed by the characters "EM", you may omit the "EM" (i.e. typing "IMGDSP" or "EMIMGDSP" is equivalent). Please note that MANY programs have little or no documentation (SORRY!).

SYS\$DOCLIB:TSBLIB.DOC

This documents how many of the general purpose and image analysis (e.g. IMG_*, FFT_*, SF_*, LL_* and MAP_*) subroutines contained in the object library SYS\$EMLIB:TSBLIB.OLB are used. This documentation is especially important for programmers wishing to write new applications software using these standard routines.

SYS\$LXLIB:LXLIB.DOC

This documents the Purdue LX_* LEXIDATA graphics subroutines used to call the LEXIDATA IDOS software. LX_* routines adopt a convention in which the origin of data is in the lower left corner (x,y = 0,0). See further comments in Section G. (USE OF INTERACTIVE RASTER GRAPHICS).

2. NOTES TO PROGRAMMERS

- a. Additional "documentation" is often included in the FORTRAN source code, usually at the beginning of the code, but also at appropriate locations throughout the code. Be forewarned, however, that a fair amount of undocumented code exists owing to the fact that some has been inherited from other, well-meaning "hackers". In many instances I have attempted to add documentation in portions of the code.
- b. An attempt has been made throughout the EM system code to use a set of parameters and variables with standardized but unique names, passed through COMMON blocks defined in the INCLUDE files EM.CMM and KEYS.CMM in SYS\$EMLIB and LEXI.CMM in SYS\$LXLIB. It is highly recommended (PLEASE !) that new software adopt the use of these specific parameters. Check EM.CMM, KEYS.CMM and LEXI.CMM for descriptions of the parameters. Wherever possible, the parameters have been assigned names which are unique and informative, and most importantly, are not likely to be used in any other context by unsuspecting programmers.

D. RUNNING PROGRAMS IN THE EM SYSTEM

SYS\$EXELIB:EM.EXE

EM is the main driver program for all the image analysis as well as many other programs in the SYS\$EXELIB directory. This program can be run from any USER directory entering the following command, followed by <RETURN>:

\$RUN SYS\$EXELIB:EM

Documentation on how the program EM works may be found in EM.DOC in the SYS\$DOCLIB directory. This program is mainly intended for new users, or users outside the image analysis group who may be less familiar with the EM programs. For "experts", it is generally simpler to set up a command file of LOGICAL SYMBOLS which establish a set of simple acronyms for running various programs. This command file should be executed whenever the user logs onto the computer.

As an example, a file called RUN.COM could be set up in which statements such as the following might be included:

```
$ EM:==          "RUN SYS$EXELIB:EM"
$ FFT:==        "RUN SYS$EXELIB:EMFFT"
$ GRAPH*ICS:==  "RUN SYS$LXLIB:LX_GRAPHICS"
$ ICO:==        "RUN SYS$EXELIB:EMICO"
$ IMG:==        "RUN SYS$EXELIB:EMIMG"
$ MAP:==        "RUN SYS$EXELIB:EMMAP"
```

Thus, typing FFT, followed by <RETURN> will invoke the running of the program called "EMFFT". (Fewer keystrokes = fewer mistakes!)

E. BUILDING A SYSTEM

A number of command files (EM*.BLD) reside in the SYS\$BLDLIB directory. These have been developed to help facilitate "automatic" generation of portions of or the entire EM system. Before these command files can be successfully executed, three object libraries must be created (at Purdue these are called SYS\$EMLIB:TSBLIB.OLB, SYS\$LXLIB:LXLIB.OLB and [TSB.NEWFV]SDFLIB.OLB).

Batch-job control command files for rebuilding all EMICO*, EMIMG*, EMFFT*, EMLL*, EMSF*, EMMAP* and MBLD* programs are contained in the appropriate *.BLD files in the SYS\$BLDLIB directory. Execution of any of these command files will regenerate a portion of the EM system. To regenerate the entire system of programs, the EMSYSTEM.BLD command file can be invoked. Of course, these can only be run after the three object libraries mentioned above exist.

TSBLIB.BLD may be used to rebuild all the object code in the SYS\$EMLIB:TSBLIB.OLB library. Alternatively, portions of the library may be rebuilt with the command files TSBLIB_IMG.BLD, TSBLIB_FFT.BLD, TSBLIB_MAP.BLD, TSBLIB_SF.BLD, and TSBLIB_MISC.BLD in the SYS\$BLDLIB directory.

SYS\$LXLIB:LXLIB.BLD is used to rebuild the object code in the SYS\$LXLIB:LXLIB.OLB library. Note that this command file is stored in SYS\$LXLIB.

F. SOURCE CODE AND OBJECT LIBRARIES

FORTRAN source code for most of the EM programs is kept in the SYS\$EMLIB directory in files named EM*.FOR and EM*.SUBS. Source code for many of the general purpose subroutines for manipulating IMAGE, FFT, MAP, LL and SF data are kept in the SYS\$EMLIB in files such as FFTLIB1.SUBS, FFTLIB2.SUBS, ICOLIB.SUBS, IMGLIB.SUBS, LLLIB.SUBS, MAPLIB.SUBS and SFLIB.SUBS. Source code for a number of general purpose subroutines are in MISCLIB.SUBS. See SYS\$DOCLIB:TSBLIB.DOC for documentation about how specific routines are used. Other general purpose routines for VT screen management may be found in VTLIB.SUBS and SCRLIB.SUBS in SYS\$EMLIB.

Three object libraries contain most of the compiled versions of subroutines used by the EM programs. They are SYS\$EMLIB:TSBLIB.OLB, SYS\$LXLIB:LXLIB.OLB and [TSB.NEWFV]SDFLIB.OLB. LXLIB.OLB contains the LX_* and many other graphics subroutines (see F.). SDFLIB.OLB contains many of the icosahedral processing routines. These object libraries are created according to the specifications given in Section D. (BUILDING A COMPLETE SYSTEM).

A series of useful routines for handling interactive data processing are found in the SCR_*.SUBS files in the SYS\$EMLIB. The original versions of these routines were written and documented by Eric Meyer in Jun-Aug, 1987. Look at the program EMIMG for examples of the use of these routines.

SYS\$EMLIB:EM.CMM and SYS\$LXLIB:LEXI.CMM

These files contain declarations for general purpose parameters and variables used in the EM system software. Most programs and subroutines begin (immediately following the IMPLICIT NONE statement) with one or more of the following INCLUDE statements:

```
INCLUDE 'SYS$EMLIB:EM.CMM'  
INCLUDE 'SYS$EMLIB:KEYS.CMM'  
INCLUDE 'SYS$LXLIB:LEXI.CMM'
```

These INCLUDE statements result in the contents of SYS\$EMLIB:EM.CMM and SYS\$LXLIB:LEXI.CMM to be included with the remaining source code. Thus, the parameters and variables

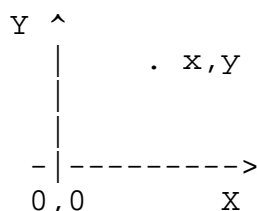
declared in EM.CMM and LEXI.CMM need not (and CANNOT) be declared in the source code (should you ignore this warning, the compiler will not-so-kindly remind you of your error!). The two *.CMM files specify several COMMON blocks enabling the declared parameters and variables to be passed between program segments.

G. INTERACTIVE RASTER GRAPHICS

1. Description

Interactive graphics are performed in the Purdue EM system on a 1280 by 1024 resolution Model 3400 LEXIDATA color raster graphics device. The IDOS software supplied by LEXIDATA places the screen origin at the upper left corner of the display, however, the EM programs are all based on a lower left origin system. Consequently, a series of higher level FORTRAN subroutines, SYS\$LXLIB:LXLIB.SUBS, were developed to allow the programmer to write applications graphics software which adopts the lower-left origin convention. These LX_* subroutines call the LEXIDATA IDOS FORTRAN routines and make the necessary origin conversion transparent to the user. It is STRONGLY RECOMMENDED that these subroutines be used in new applications software rather than using the IDOS routines directly. Many LX_* routines are virtually identical to the appropriate DS* LEXIDATA routines. Please note that NOT ALL of the IDOS routines have LX_* counterparts, since some still need to be written.

2. Diagram of graphics coordinate system



COORDINATE AXIS CONVENTION FOR LEXIDATA GRAPHICS
ORIGIN IS AT LOWER LEFT OF SCREEN (MEMORY)

3. Important files for graphics applications and descriptions of contents.

SYS\$LXLIB:LXLIB.SUBS

This file contains most of the FORTRAN LX_* graphics applications routines which are used to call the IDOS routines supplied by LEXIDATA. They use a lower-left corner origin convention. The LX_* subroutines call the LEXIDATA IDOS routines

whose object code is stored in the SYSTEM library LEXLIB.OLB in the directory SYS\$SYSROOT:[SYSMGR.LEX].

The file LXDATA.FOR in the SYSTEM directory is also needed when linking programs that use the LX_* routines.

SYS\$LXLIB:LXLIB.DOC

Documents the functions of and manner of invoking each LX_* graphics subroutine.

SYS\$LXLIB:LXLIB.OLB

Object library of the LX_* and other graphics subroutines.

SYS\$BLDLIB:LXLIB.BLD

Command file for rebuilding the object code in LXLIB.OLB for all the LX_* type and other graphics subroutines.

SYS\$LXLIB:LEXI.CMM

File included with programs using LEXIDATA graphics subroutines. This contains several COMMON blocks which serve to pass a variety of constants and variables used by the graphics routines. An INCLUDE statement should be added to the beginning of any routine requiring parameters stored in these COMMON blocks.