Lecture notes:
How to manage the EGS4 system

Alex F Bielajew
Institute for National Measurement Standards
National Research Council of Canada
Ottawa, Canada
K1A 0R6

Tel: 613-993-2715
FAX: 613-952-9865
e-mail: `alex@irs.phy.nrc.ca`

National Research Council of Canada Report PIRS-0391

> *Documentation is like sex:*
> *When it's good, it's very, very good, and*
> *when it's bad, it's better than nothing...*
> *Anon.*

1 Introduction

This lecture describes how to get the EGS4 distribution, install it and Mortran3 compile, Fortran compile, link and run an EGS4 user code on a unix system. The system should install and run without modification on Sun/Solaris_1.x.x (a BSD 4.3-based UNIX) and SGI/IRIX (a SVR4 UNIX) systems. The examples provided herein were performed on a Sun. For systems other than Sun or SGI's, all of the scripts will require some modifications. It is recommended that any modifications to these scripts follow the same design philosophy—keeping them as general as possible with the capability of running on other architectures left intact. This prevents the codes from becoming system dependent and liberates this constraint for future hardware acquisitions.

We shall describe in detail the inputs to two multi-purpose unix scripts called `egs4_compile` and `egs4_run` which perform the bulk of the work for EGS4 users. PEGS4 datafile processing, standalone Mortran3 processing and batch processing are described as well.

2 Installation and maintenance

The person responsible for installing and maintaining the EGS4 system should read this section.

2.1 How to get EGS4 for unix machines

This is an excerpt from a standard blurb that is sent to those making inquiries about EGS4:

```
*******************************************************************************
***********************                                 ***********************
***********************   H O W   T O   G E T   E G S   ***********************
***********************                                 ***********************
*******************************************************************************

HOW_TO_GET_EGS.txt - BLIF 92/12/09

Procedures for obtaining the EGS4 code:
=======================================

*** In all cases ***
After you have received your EGS4 distribution, a letter (on company
letterhead, if possible) must be sent to Dr. Nelson (address below)
with a CC to me (Alex Bielajew, address below) so that you may be
registered as a user and you must include a paragraph or two describing
your intended applications for EGS4.  Dr. Nelson also distributes the
```

```
manuals and you will receive one if you request it in your letter.


*** Note ***
All the distributions below include the PRESTA macros and routines.

For UNIX systems:
=================

Includes:
- Standard EGS4 distribution
- Scripts and utilities
- EGS_Windows  (a graphical interface, SunView,  SunPHIGS_1.2)
- EGS_Windows2 (a graphical interface, OpenLook, SunPHIGS_2.0)
- example user codes
- Physics extensions slated for next release of EGS
- Scripts run without modification on Sun's and SGI/4D machines.
- Binaries supplied for Sparc and Iris architectures
- Notes re port to AIX machines available on request
Requires:
- Fortran compiler (with NAMELIST option)
- PHIGS (1.2 or 2.0) (optional, for graphical interface)

Preferred method of distribution is by anonymous ftp.

Anonymous ftp sites:

nrcnet0.nrc.ca   (132.246.160.2)
academic.lbl.gov (128.3.12.48)

cd to pub/egs4 and browse for what you need.
Read the *README*'s and take what you need.
Please do this during off-peak hours if possible.
nrcnet0.nrc.ca   is in the Eastern Time Zone,
academic.lbl.gov is in the Pacific Time Zone.

Minimum distribution for 2.0 is:

pub/egs4/Unix_Version_2.0/INSTALL.guide      - Installation guide for 2.0
pub/egs4/Unix_Version_2.0/INSTALL.all        - Installation script for 2.0
pub/egs4/Unix_Version_2.0/egs4unix_2.0.tar.Z - Basic 2.0 distribution
```

2.2 Installation issues

This the the `INSTALL.guide` supplied with the system:

```
===================================================
INSTALL.guide - Alex F Bielajew (aka BLIF) 92/11/02
```

This guide is intended to help you navigate your way through the installation of the EGS4 system on a unix machine.

First decision: How will I maintain the system?

The installation of the EGS4 system relies on the existence of a root directory from where the EGS4 filesystem will be maintained. This directory is called the HEN_HOUSE. There will be a unix environment variable called HEN_HOUSE which is used by all the scripts and utilities.

I HIGHLY recommend that HEN_HOUSE be the root directory of an account called, say, "egs4". This account should be logged in to only for installation and maintenance purposes. While this introduces a little more administrative work, it is the safest and the system you are left with is standard. If you get into trouble and need help, I will find my way around a standard system more easily. This is the way I run my system.

Here are the possibilities for HEN_HOUSE:
(~x mean home directory of user with login name "x")

```
1) Single-user
   2 accounts:
   "egs4" for maintenance, and
   "user" for compiling and running user codes
   MOST HIGHLY RECOMMENDED
   =======================
   HEN_HOUSE = ~egs4
   Pro's: Most safe, standard, and secure.
          "egs4" can not compile or run user codes keeping the
          integrity of the egs distribution.
   Con's: "user" has to maintain another account.
          "egs4" can not compile or run user codes resulting in some
          minor inconvenience.

2) Single-user
   2 accounts:
   "egs4" for maintenance, compiling and running user codes, and
   "user" for compiling and running user codes
   HEN_HOUSE = ~egs4/egs4
   Pro's: Moderately safe, standard, and secure.
```

```
       "egs4" can compile or run user codes which can be convenient.
Con's: "user" has to maintain another account.
       "egs4" can compile or run user codes, introducing the risk that
              the egs filesystem can be modified by a user code or
              procedure.

3) Single-user
   1 account:
   "user" for maintenance, compiling and running user codes
   LEAST RECOMMENDED
   =================
   HEN_HOUSE = ~user/egs4
   Pro's: Easy maintenance.
   Con's: "user" can modify inadvertently the EGS4 filesystem.
          Divergence from the distribution standard.

4) Multi-user, n = 1,2,3...N users
   N + 1 accounts:
   "egs4"   for maintenance, and
   "user_n" for compiling and running user codes
   MOST HIGHLY RECOMMENDED
   =======================
   HEN_HOUSE = ~egs4
   Pro's: Most safe, standard, and secure.
          "egs4" can not compile or run user codes keeping the
          integrity of the EGS4 distribution.
   Con's: One extra account to maintain.
          "egs4" can not compile or run user codes resulting in some
          minor inconvenience.

5) Multi-user,  n = 1,2,3...N users
   N + 1 accounts:
   "egs4"   for maintenance, compiling and running user codes, and
   "user_n" for compiling and running user codes
   HEN_HOUSE = ~egs4/egs4
   Pro's: Moderately safe, standard, and secure.
          "egs4"   can compile or run user codes which can be convenient.
   Con's: "user_n" has to maintain another account.
          "egs4"   can compile or run user codes, introducing the risk that
                   the EGS4 filesystem can be modified by a user code or
                   procedure.

6) Multi-user, N users
   N accounts:
   "user_1" for maintenance, compiling and running user codes, and
   "user_n" for compiling and running user codes
   LEAST RECOMMENDED
   =================
```

```
   HEN_HOUSE = ~user_1/egs4
   Pro's: Easy maintenance.
   Con's: "user_1" can modify inadvertently the EGS4 filesystem.
          Divergence from the distribution standard.
          "user_1" has to give read access to his ~user_1/egs4 area.
```

If you choose the HIGHLY RECOMMENDED approach, your hierarchy of files for the basic distribution will look like:

```
  .
  .
  .
  |----egs4                    (Maintenance area                  )
  |     |---appendix           (Documentation from SLAC           )
  |     |---bin                (Standard binary directory         )
  |     |    |---architecture_1 (Architecture_1 binaries e.g. sparc)
  |     |    |---architecture_2 (Architecture_2 binaries e.g. iris )
  |     |              .
  |     |              .
  |     |              .
  |     |---docs               (Documentation area                )
  |     |---dosrz              (Example user code                 )
  |     |---ensrc_spectra      (Energy spectrum files             )
  |     |---examin             (   "      "    "                  )
  |     |---ibm_scripts        (Command procedures for IBM/CMS    )
  |     |---inhom              (   "      "    "                  )
  |     |---inhomp             (   "      "    "                  )
  |     |---lib                (Standard library directory        )
  |     |    |---architecture_1 (Architecture_1 library e.g. sparc )
  |     |    |---architecture_2 (Architecture_2 library e.g. iris  )
  |     |              .
  |     |              .
  |     |              .
  |     |---mortran3           (Mortran3 compiler                 )
  |     |---nrcc_extras        (Old VAX routines                  )
  |     |---pegs4              (Standard PEGS4 area               )
  |     |     |---data         (Example data sets                 )
  |     |---slac_extras        (SLAC examples                     )
  |     |---tutor1             (   "      "    "                  )
  |     |---tutor2             (   "      "    "                  )
  |     |---tutor3             (   "      "    "                  )
  |     |---tutor4             (   "      "    "                  )
  |     |---tutor5             (   "      "    "                  )
  |     |---tutor6             (   "      "    "                  )
  |     |---tutor7             (   "      "    "                  )
  |     |---vax_scripts        (Command procedure for VAX/VMS     )
  |     |---xvgr_stuff         (Plotting/graphing utilities       )
```

```
  |     |---xyzdos               (  "      "    "                  )
  |
  |---egs4user_1              (User_1's  home area          )
  |       |---egs4            (User_1's  EGS4 area          )
  |            |---pegs4      (User_1's PEGS4 area          )
  |            |     |---data (User_1's PEGS4 datafiles     )
  |            |---usercode1  (User_1's EGS4 application # 1)
  |            |---usercode2  (User_1's EGS4 application # 2)
  |                .
  |                .
  |                .
  |---egs4user_2              (User_2's  home area          )
  |       |---egs4            (User_2's  EGS4 area          )
  |            |---pegs4      (User_2's PEGS4 area          )
  |            |     |---data (User_2's PEGS4 datafiles     )
  |            |---usercode1  (User_2's EGS4 application # 1)
  |            |---usercode2  (User_2's EGS4 application # 2)
  |                .
  |                .
  |                .
  |
  .
  .
  .
```

```
================================================================================
================================================================================
How do I read the tape?
================================================================================
================================================================================
```

If you obtained the files by ftp, skip to the next section.

The following files have been placed on the tape.

```
INSTALL.all                       (compulsory)
INSTALL.guide                     (compulsory)
egs4unix_2.0.tar.Z                (compulsory)
egs4unix_2.0_iris_binaries.tar.Z  (optional)
egs4unix_2.0_sparc_binaries.tar.Z (optional)
EGS_Windows_1.2.tar.Z             (optional)
EGS_Windows_2.0.tar.Z             (optional)
egs4unix_1.0.tar.Z                (for archive purposes)
egs4unix_1.1.tar.Z                (for archive purposes)
```

about 47 MB's of data in total.

Read the next section to see if you want to extract the optional and archive files.

go to the directory where EGS4 is to be installed

On a Sun system with a DC6150 tape drive
==

Type either.

```
tar -xvf /dev/rst0
```

to get all the files on the tape, or

```
tar -xvf /dev/rst0 ./INSTALL.all ./INSTALL.guide ./egs4unix_2.0.tar.Z
```

to get the most basic distribution, about 2 MB of data.

To get optional and archive files type

```
tar -xvf /dev/rst0  [filename]
```

where filename is one of the optional or archive files listed above.

On a Sun system with an 8mm Exabyte 8200 tape drive
===

Type either.

```
tar -xvf /dev/rst1
```

to get all the files on the tape, or

```
tar -xvf /dev/rst1 ./INSTALL.all ./INSTALL.guide ./egs4unix_2.0.tar.Z
```

to get the most basic distribution, about 2 MB of data.

To get optional and archive files type

```
tar -xvf /dev/rst1  [filename]
```

where filename is one of the optional or archive files listed above.

For systems other than Sun's:
============================

You should be able to read this tape (barring hardware difficulties) on any UNIX system using the dd image copy facility.

```
(I've actually only done this on a Silicon Graphics machine running IRIX
reading a DC6150 Sun tape.)

The command I used to read the whole tape was:

dd if=/dev/tape conv=swab,noerror,sync | tar xf -

If you learn of how to do this on other systems, please let me know
and I will add the information to this file.

===============================================================================
===============================================================================
How do I install the system?
===============================================================================
===============================================================================

1) You have obtained the distribution either through ftp transfer or on tape.
2) You have decided where HEN_HOUSE should be.
3) You are logged on to the account that has write access to the HEN_HOUSE
4) You have the ONLY the following files in your HEN_HOUSE:

   (*** Warning***                                                          )
   (If you have other files in this filesystem they may be overwritten by the)
   (installation procedure.                                                 )

   INSTALL.guide - This file, which you have read completely.
                   Make sure you have execute permission on this file.

   INSTALL.all   - a csh script which auto installs the entire system and
                   and builds the essential binaries if need be.
                   Make sure you have execute permission on this file.
                   If not, change it with:
                   chmod 744 INSTALL.all

   egs4unix_2.0.tar[.Z] (The ftp files are compressed.)
                 - This is the standard distribution and is required.
                   You require at least 17 MB to unload this file

   The following files are optional:

   egs4unix_2.0_iris_binaries.tar[.Z]  - binaries for an SGI/iris machine.
                                         You require 16 MB to unload this file.
   egs4unix_2.0_sparc_binaries.tar[.Z] - binaries for a Sun/SPARC machine.
                                         You require 17 MB to unload this file.
   EGS_Windows_1.2.tar[.Z]             - Sun/SunView/PHIGS1.2 graphical tool
                                         with solids modeling.
                                         You need SunPHIGS1.2 to make this work.
                                         You require 38 MB to unload this file.
```

```
EGS_Windows_2.0.tar[.Z]            - Sun/OpenLook/PHIGS2.0 graphical tool
                                     without solids modeling.
                                     You need SunPHIGS2.0 to make this work.
                                   - You require 65 MB to unload this file
```

The required storage space for each distribution listed above is an estimate only. It may be slightly different on your machine. If you are very tight on disk space you can put the tarfiles on another filesystem but you will have to modify "INSTALL.all" to make it work. You may also throw away the tarfiles afterwards or compress than and keep them for archival purposes.

5) Run the INSTALL.all script by typing:

```
csh -f INSTALL.all
```

If you are on an iris or sparc machine the script should be allowed to run to completion. It has been tested on both these machines. If you are running another architecture you are, regrettably, in for some work.... Please attempt the installation by modifying the installation scripts and communicating your changes to me so that others do not have to suffer. There are some TeX'ed notes in $HEN_HOUSE/lib/rs6000 for IBM RS6000 machines. Some of these notes pertain to the egs4unix_1.1 installation and I have not brought them up to date and don't intend to until forced to install the system on one of these machines.

6) INSTALL.all does the following:

a) Types INSTALL.guide on your screen
b) extracts files from egs4unix_2.0.tar.
   If any of your tarfiles are compressed, it uncompresses them and recompresses them when it is finished with them.
c) INSTALL.all modifies the first line of the file:

   Cshrc_additions_for_egs4

   which is a file that should be "sourced" by all EGS4 users in their ~/.cshrc file with the line

```
source (HEN_HOUSE)/Cshrc_additions_for_egs4
```

   where the value for HEN_HOUSE should be given explicitly, eg

```
source ~egs4/Cshrc_additions_for_egs4
```

   This should go in the section of .cshrc that is relevant for both interactive and batch shells, as all shells need to know about the definitions in this file. This file defines a set of mnemonics and "set variables" that make it easy to navigate through the EGS4 system. Most importantly, it defines the HEN_HOUSE location if the user has not already defined it.

d) If you are on an iris or sparc machine and you have the file:

egs4unix_2.0_iris_binaries.tar

and/or

egs4unix_2.0_sparc_binaries.tar

INSTALL.all extracts files from the appropriate tarfile. If you do not have the appropriate tarfile, INSTALL.all attempts to build the essential binaries.

e) If you have the files:

EGS_Windows_1.2.tar

or

EGS_Windows_2.0.tar

INSTALL.all extracts files from these archives as well.

f) Types to your screen the compulsory readme files given in step 8 below.

g) If INSTALL.all ruins successfully, it executes

```
chmod 444 INSTALL.all
```

so that it can not be run again. (Of course, you can change it back!)

7) Modify your ~/.cshrc as described in step 6c) above and tell all EGS4 users to do the same.

8) Read the following files in order:

```
$HEN_HOUSE/PHILOSOPHY        (compulsory)
$HEN_HOUSE/000_README_OR_DIE (compulsory)
$HEN_HOUSE/README.general    (compulsory)
$HEN_HOUSE/README.filelist   (compulsory)
$HEN_HOUSE/CHANGES_from_1.1  (if you were an egs4unix_1.1 user)
$HEN_HOUSE/*/README*         (as you start to use the tools in  $HEN_HOUSE/*/)
```

9) If you are on a sparc and you want to install the iris binaries, type

```
   uncompress egs4unix_2.0_iris_binaries.tar # If compressed
   tar -xvf egs4unix_2.0_iris_binaries.tar

9) If you run into trouble, contact me and I will try to help.

   If the trouble is caused by skipping sections 1)--8) I tend to get surly
   and will recommend that you RFM.

   If you have suggestions I am always pleased to entertain them.

   These ideas get incorporated into higher versions if I find them of
   general use.

   If you have made changes of general interest I would be delighted
   to include them in higher versions and acknowledge your help.

================================================================================
================================================================================
How do I run different architectures in parallel?
================================================================================
================================================================================

There is no capability for this yet. It will come in future releases.
However, if you have defined your hostnames properly,

Cshrc_additions_for_egs4

will set up a set of mnemonics for using your remote machines.

To set up sparc hostnames sleepy and sneezy
and iris hostnames doc and dopey:

mkdir $HEN_HOUSE/lib/sparc/hostnames
touch sleepy
touch sneezy

mkdir $HEN_HOUSE/lib/iris/hostnames
touch doc
touch dopey

================================================================================
================================================================================
How do I get help?
================================================================================
================================================================================
```

1) An EGS4 listserv account has been set up.

This account is meant to field all questions regarding EGS4, irrespective of systems.

You post your questions to this account and the wide community of subscribers has a chance to answer your query.

To sign on, send an e-mail message to:

`listserv@slacvm.slac.stanford.edu` (Internet), or `listserv@slacvm` (Bitnet)

saying:

`SUBscribe EGS4-L "Your full name"`

To post questions/comments/answers, send e-mail to:

`egs4-l@slacvm.slac.stanford.edu` (Internet), or `egs4-l@slacvm` (Bitnet)

2) Questions specific to this distribution -

e-mail to:

`rdonahue@academic.lbl.gov` (Rick Donahue)

or,

`alex@irs.phy.nrc.ca` (Alex Bielajew)

Remember, we do this for free, so be nice.

3) Questions regarding EGS4 in general:

Name	e-mail	telephone	fax
Alex F Bielajew	alex@irs.phy.nrc.ca	613 992 2197	613 952 9865
Rick Donahue	rdonahue@academic.lbl.gov	510 486 5597	510 486 6608
Hideo Hirayama	hh02@kekcc.kek.jp		
W Ralph Nelson	wrnrp@slacvm.slac.stanford.edu	415 926 2047	415 926 3569
Dave W O Rogers	dave@irs.phy.nrc.ca	613 992 2197	613 952 9865
Sue Walker		206 545 1141	206 545 1347

2.3 Example of an installation on a Sparc machine

Following is an example of an ftp retrieval and minimal installation on a Sparc. These procedures should be executed from the root directory of the EGS4 maintenance account/area. The output from the installation scripts is more verbose than indicated. Only the essentials were kept. Lines that require user input have "`# User input #`" appended to them. Additional comment lines start with a "`#`".

```
#
# Start of ftp session
#
> ftp nrcnet0.nrc.ca  # User input #

Name (nrcnet0.nrc.ca:egsboss): anonymous  # User input #

331 Guest login ok, send e-mail address as password.

Password: yourname@your.internet.address  # User input #

#
# Ignore the following warning
#
230-*************************************************************************
230-*                                                                       *
230-*          This anonymous FTP directory is for use of NRC employees only   *
230-*                                                                       *
230-*                    Unauthorised Access Or Use Is Prohibited              *
230-*                                                                       *
230-*************************************************************************

ftp> cd pub/egs4/Unix_Version_2.0  # User input #
ftp> ls -l                         # User input #

-r--r--r--  1 3221        3905 Dec  8  1992 EGS4README.ftp
-r--r--r--  1 3221     6487775 Dec  8  1992 EGS_Windows_1.2.tar.Z
-r--r--r--  1 3221     9023879 Dec  8  1992 EGS_Windows_2.0.tar.Z
-r--r--r--  1 3221        7947 Dec  8  1992 INSTALL.all
-r--r--r--  1 3221       18840 Dec  8  1992 INSTALL.guide
-r--r--r--  1 3221     2544515 Dec  8  1992 egs4unix_2.0.tar.Z
-r--r--r--  1 3221     4126079 Dec  8  1992 egs4unix_2.0_iris_binaries.tar.Z
-r--r--r--  1 3221     5083203 Dec  8  1992 egs4unix_2.0_sparc_binaries.tar.Z

ftp> get INSTALL.guide       # User input #
ftp> get INSTALL.all         # User input #
ftp> bin                     # User input #
ftp> get egs4unix_2.0.tar.Z  # User input #
ftp> quit                    # User input #
#
```

```
# End of ftp session
#

> ls -l  # User input #

-rw-rw-r--  1 egsboss      7947 Aug  5 09:18 INSTALL.all
-rw-rw-r--  1 egsboss     18840 Aug  5 09:18 INSTALL.guide
-rw-rw-r--  1 egsboss   2544515 Aug  5 09:19 egs4unix_2.0.tar.Z
-rw-rw-r--  1 egsboss         0 Aug  5 09:17 install

#
# Start of installation
#
> chmod 755 INSTALL.*  # User input #
> INSTALL.all          # User input #
===================================================
INSTALL.guide - Alex F Bielajew (aka BLIF) 92/11/02
===================================================
.
.
.
Lots of text
.
.
.
Continue with installation (y or n[default])? y  # User input #

Will assume that /home/nrc/egsboss
is the HEN_HOUSE and modify files accordingly.

Is this OK (y or n[default]) ? y  # User input #

Attempting full installation of the EGS4 system...
==================================================

Architecture type --- sparc
===========================

/home/nrc/egsboss

Uncompressing  egs4unix_2.0.tar...

Extracting files from egs4unix_2.0.tar ...
.
.
.
Long list of files
.
```

```
.
.
Recompressing egs4unix_2.0.tar ...

Mortran3 installation
=====================

Fortran compile Mortran3 source code...
======================================

Directory /home/nrc/egsboss/bin does not exist. Creating it.

Directory /home/nrc/egsboss/bin/sparc does not exist. Creating it.
.
.
.
Compiler messages
.
.
.
Finished compiling the Mortran3 compiler...

Compilation completed successfully.

Convert raw file to hexadecimal...
==================================

Converting raw data (mornew77.raw) into hex data (mortran3.dat)...

Finished converting raw data (mornew77.raw) into hex data (mortran3.dat)...

Stripping the initial comment line from mortran3.dat...

Finished...

-rw-rw-r--  1 egsboss     84022 Aug  5 09:33 /home/nrc/egsboss/bin/sparc/mortran3.dat

Testing the Mortran3 compiler by compiling check77.mortran...
=============================================================

User code....................../home/nrc/egsboss/mortran3/check77.mortran
Debug option..............................off
Compile option............................n

Creating .mortjob.mortran...

Mortran3 compiling .mortjob.mortran...
```

```
Fortran compiling check77.fortran...
.
.
.
Compiler messages
.
.
.
Compilation completed successfully.

Running check77.exe... Results should be sensible!
=================================================

 THE FIRST TEN INTEGERS AND THEIR SQUARES:

 I=  1 ( ODD INTEGER)   1
 I=  2 (EVEN INTEGER)   4
 I=  3 ( ODD INTEGER)   9
 I=  4 (EVEN INTEGER)  16
 I=  5 ( ODD INTEGER)  25
 I=  6 (EVEN INTEGER)  36
 I=  7 ( ODD INTEGER)  49
 I=  8 (EVEN INTEGER)  64
 I=  9 ( ODD INTEGER)  81
 I= 10 (EVEN INTEGER) 100

 Warning: nonstandard floating-point arithmetic mode
 was enabled in this program and was never cleared.

Finished Mortran3 installation
==============================

Compiling PEGS4 ...

User code......................./home/nrc/egsboss/pegs4/pegs4.mortran
Debug option.............................off
Compile option...........................n

Creating .mortjob.mortran...

Mortran3 compiling .mortjob.mortran...

Fortran compiling pegs4.fortran...
.
.
.
Compiler messages
.
```

```
.
.
Compilation completed successfully.

Compiling neatify, a utility routine ...

User code....................../home/nrc/egsboss/neatify.mortran
Debug option..............................off
Compile option............................n

Creating .mortjob.mortran...

Mortran3 compiling .mortjob.mortran...

Fortran compiling neatify.fortran...
.
.
.
Compiler messages
.
.
.
Compilation completed successfully.

Philosophy - BLIF 92/10/29
==========================
.
.
.
Lots of text
.
.
.
================================
000_README_OR_DIE: BLIF 92/04/14
================================

I don't want to be melodramatic, but this is important.

If you are running SunOS 4.1.1 (and earlier, maybe) and you use tmpfs
(ask a sys admin if you don't know what this means), Fortran fails to
initialize DATA statements correctly.  I have a patch in
$HEN_HOUSE/bin/sparc/bug.patches that can counter this. An example of
how to compile with this is given in $HEN_HOUSE/standard.configuration.

==============================
README.general - BLIF 92/11/03
==============================
```

```
.
.
.
Lots of text
.
.
.
=======================================
README.filelist - BLIF version 92/10/29
=======================================
.
.
.
Lots of text
.
.
.
Are you a former egs4unix_1.1 user (y or n[default])? y  # User input #

================
CHANGES_from_1.1
================
.
.
.
Lots of text
.
.
.
Successful full installation of the EGS4 system
===============================================
```

3 Compiling and running EGS4 usercodes in the unix system

Those concerned with compiling and running EGS4 codes should read this section.
All EGS4 users should add the following line to their `.cshrc` file.

```
.
.
.
source /home/nrc/egsboss/Cshrc_additions_for_egs4
.
.
.
```

This line should be positioned in the `.cshrc` file so that it is read by all shells, both interactive and non-interactive. In the above example, it is assumed that the directory "`/home/nrc/egsboss`" is the root directory for maintenance of EGS4.

The most important function of `Cshrc_additions_for_egs4` is to define the environment variable `HEN_HOUSE` which is the root directory for maintenance of EGS4, `/home/nrc/egsboss` in this example. When the EGS4 system is upgraded, the EGS4 manager only has to change this environment variable and all users automatically access the upgraded system.

The other function for `Cshrc_additions_for_egs4` is to define mnemonics for short-cuts in accessing the EGS4 scripts and in navigating through the EGS4 filesystem. The most important of these are described below.

For the remainder of this section, the following notation is used:

`[a|b|c]` means one of options `a`, b or `c` MUST be chosen.
{`a|[b]|c`} means that choice does not have to be made, `b` is chosen by default.

There is one other environment variable that the user may define optionally. If the user defines the environment variable `EGS_PERT`, then the prompts and messages from most of the scripts are kept to a minimum.

3.1 Compiling using `egs4_compile`

In order to compile an EGS4 usercode, the user must have a directory called `~/egs4/usercode`, where `usercode.mortran` resides. The tilde, “~” is csh notation for a user's home directory. If `usercode.mortran` does not exist on `~/egs4/usercode`, then `egs4_compile` looks for it in `$HEN_HOUSE/usercode`. If it finds it there, then it is copies over to the user's `~/egs4/usercode` area. This is useful if the user wants to compile a standard code that resides in the `$HEN_HOUSE`.

The compilation is accomplished with the statement:

```
$HEN_HOUSE/egs4_compile [m|f|fl|clean] [usercode] {[a]|x} {debug|[noopt]|opt1|opt2|opt3|opt4}
```

where the various options are:

```
m     - Mortran3 compile usercode.mortran
f     - Fortran compile and link usercode, assumes "m" step executed previously
mf    - Mortran3 compile, Fortran compile and link usercode
clean - clean all unnecessary files from the usercode area

a  - work on the source file as one big file
x  - split the source file into separate routines
     speeding up development times

debug - invoke debugger
noopt - no optimization
opt1  - optimization level 1
opt2  - optimization level 2
opt3  - optimization level 3
opt4  - optimization level 4
```

Some useful mnemonics(aliases) made by `Cshrc_additions_for_egs4` are:

```
m  == $HEN_HOUSE/egs4_compile m
f  == $HEN_HOUSE/egs4_compile f
mf == $HEN_HOUSE/egs4_compile mf

mb  == $HEN_HOUSE/egs4_batch_compile m     # same as above but in batch
fb  == $HEN_HOUSE/egs4_batch_compile f     # same as above but in batch
mfb == $HEN_HOUSE/egs4_batch_compile mf    # same as above but in batch
```

Another file that is associated with the compilation is `~/egs4/usercode/usercode.configuration`. This file tells the Mortran3 compiler how to arrange Mortran3 source files for presentation to the Mortran3 compiler. If `~/egs4/usercode/usercode.configuration` is not found, then a standard configuration file, `$HEN_HOUSE/standard.configuration` is employed.

Following is an example of the compilation of the standard user code, `tutor1.mortran` that employs `$HEN_HOUSE/standard.configuration`. Lines that require user input have "`# User input #`" appended to them. Additional comment lines start with a "`#`". Assuming that the user has a directory named `~/egs4/tutor1` that is initially empty, this is how the compilation proceeds (The user is not an `EGS_PERT` and so there is much information that is echoed.):

```
> mf tutor1  # User input #

/home/nrc/egsboss/egs4_compile
+-----------------------------------------------------------------+
* egs4_compile - Sun/SGI/unix/csh Version 1992/10/28/19:30        *
*                                                                 *
*           Sun/SGI/unix/csh script file to compile a user code *
*           with the EGS4 code system.                            *
*                                                                 *
*           First time user, inquiries, BUGS?                     *
*           Contact alex@irs.phy.nrc.ca                           *
*           and/or                                                *
*           Alex F Bielajew                                       *
*           Institute for National Measurement Standards          *
*           National Research Council of Canada                   *
*           Ottawa, CANADA                                        *
*           K1A OR6                                               *
*           Tel: 613-993-2197                                     *
*           FAX: 613-952-9865                                     *
+-----------------------------------------------------------------+

Option..................................Mortran3/Fortran compile and link
User code...............................tutor1
Mode....................................All, tutor1 and EGS4
Debug option............................OFF
```

```
Optimization..............................OFF

Type <Control-C><Return> (soft exit), or <Return> (continue) :   # User input #

Could not find tutor1.mortran on /home/nrc/egsuser/egs4/tutor1

Will attempt to copy tutor1.mortran from system directory

Type <Control-C><Return> (exit), or <Return> (continue) :        # User input #

Copying Mortran3 source /home/nrc/egsboss/tutor1/tutor1.mortran to /home/nrc/egsuser/egs4/tutor1

Creating .mortjob.mortran...

Using /home/nrc/egsboss/standard.configuration...

user/egs4 configuration and hierarchy
-------------------------------------
/home/nrc/egsboss/egs4mac.mortran(EGS4 standard macros)
/home/nrc/egsboss/nrcc4mac.mortran(NRCC standard macros)
/home/nrc/egsboss/lib/sparc/machine.mortran(Code conversion macros and RNG)
------------------------------------------------------------------------
tutor1.mortran(User code - macros and source)
------------------------------------------------------------------------
/home/nrc/egsboss/egs4blok.mortran(EGS4 standard block data)
/home/nrc/egsboss/egs4.mortran(EGS4 standard subroutines)
------------------------------
end of user/egs4 configuration

Finished creating .mortjob.mortran

Type <Control-C><Return> (soft exit), or <Return> (continue) :   # User input #

Mortran3 compiling .mortjob.mortran...

Fortran compiling and linking tutor1_sparc.f ...

No /home/nrc/egsuser/egs4/bin/sparc binary directory.

Type <Control-C><Return> (soft exit), or
     <Return> (create directory and continue) :   # User input #
.
.
.
Fortran compilation messages
.
.
.
```

```
Compilation completed successfully.
```

Under the direction of `$HEN_HOUSE/standard.configuration` the Mortran3 source files:

```
$HEN_HOUSE/egs4mac.mortran           (EGS4 standard macros)
$HEN_HOUSE/nrcc4mac.mortran          (NRCC standard macros)
$HEN_HOUSE/lib/sparc/machine.mortran (Code conversion macros and RNG)
~/egs4/tutor1/tutor1.mortran         (User code - macros and source)
$HEN_HOUSE/egs4blok.mortran          (EGS4 standard block data)
$HEN_HOUSE/egs4.mortran              (EGS4 standard subroutines)
```

were concatenated into a single file called `~/egs4/usercode/.mortjob.mortran` and presented to the Mortran3 compiler. The file `machine.mortran` contains architecture-specific code conversion macros. This file can be employed to make usercode's machine-independent.

After the compilation, the following files are left remaining in `~/egs4/usercode`:

```
.mortjob.mortran (concatenated file for compilation)
tutor1.mortlst   (Mortran3 compile listing)
tutor1.mortran   (User code)
tutor1.sparc_exe -> ~/egs4/bin/sparc/tutor1.exe (Executable)
tutor1_sparc.f   (Fortran produced by Mortran3 compilation)
```

The outputs of the Mortran3 compilation are the Mortran3 compile listing `tutor1.mortlst`, the Fortran source produced by Mortran3, `tutor1_sparc.f` and the executable produced by the Fortran compilation, `tutor1.sparc_exe`. Note that the compilation script keeps track of the architecture where the compilation was performed and employs this information into the output filenames so that the area `~/egs4/usercode` can be used by more than one unix machine (via NFS mounts) with a single Mortran3 source code. Also, the executable, `tutor1.sparc_exe` is actually a link into the user's EGS4 binary repository area `~/egs4/bin/sparc` for this architecture which was created by `egs4_compile`. This foreshadows a future version egs4unix that will allow parallel execution of usercodes in a mixed-architecture environment.

Other codes require additional Mortran3 source files. The `dosrz.configuration` file assembles the following source files:

```
$HEN_HOUSE/egs4mac.mortran           (EGS4 standard macros)
$HEN_HOUSE/nrcc4mac.mortran          (NRCC standard macros)
$HEN_HOUSE/presta.macros             (NRCC PRESTA macros)
$HEN_HOUSE/lib/sparc/machine.mortran (code conversion macros and RNG)
$HEN_HOUSE/ranmar.macros             (long-sequence RNG for the SUN)
$HEN_HOUSE/ranmar.correlations       (ranmar related macros for correlations/restarts)
~/egs4/dosrz/dosrz.mortran           (user code - macros and source)
$HEN_HOUSE/ranmar.init               (initialization routine for ranmar)
$HEN_HOUSE/prnter.mortran            (printer controller code source)
$HEN_HOUSE/srcrz.mortran             (NRCC cylindrical/planar geometry)
```

```
$HEN_HOUSE/ensrc.mortran            (NRCC input energy distribution)
$HEN_HOUSE/nrccauxp.mortran         (NRCC standard subroutines modified for PRESTA)
$HEN_HOUSE/presta.mortran           (NRCC PRESTA subroutines)
$HEN_HOUSE/egs4blok.mortran         (EGS4 standard block data)
$HEN_HOUSE/egs4.mortran             (EGS4 standard subroutines)
```

In addition to the standard files that were linked with `tutor1`, `dosrz` uses the PRESTA macros and routines, `presta.macros`, `nrccauxp.mortran` and `presta.mortran`, files associated with the long-sequence random number generator, `ranmar.macros`, `ranmar.correlations` and `ranmar.init`, a printer controller, `prnter.mortran`, a source routine for cylindrical-planar geometry, `srcrz.mortran`, and source code for modeling an input energy spectrum, `ensrc.mortran`.

The absolute minimum configuration for an EGS4 usercode is:

```
$HEN_HOUSE/egs4mac.mortran          (EGS4 standard macros)
~/egs4/usercode/usercode.mortran    (user code - macros and source)
$HEN_HOUSE/egs4blok.mortran         (EGS4 standard block data)
$HEN_HOUSE/egs4.mortran             (EGS4 standard subroutines)
```

The ordering is critical because user-defined macros can replace ones that were defined earlier. There are plenty of examples in the distribution. For your own usercodes, start with either `$HEN_HOUSE/standard.configuration` or one of the examples `$HEN_HOUSE/*/*.configuration`.

Although the use of configuration files may seem somewhat cumbersome, it is a powerful and general way of allowing flexibility in compilation. Moreover, the entire layout of the compilation is contained in just one file. There are more compact ways of stringing files together for Mortran3 compilation, (the `%U` Fortran unit switches internal to Mortran3) but these have fallen into disuse. As indicated previously, the compile script creates the concatenated file `.mortjob.mortran`. On more than one occasion I have accidentally erased my usercode and recovered it by extracting is out of `.mortjob.mortran`. Sometimes redundancy can be a good thing!

3.1.1 Compilation strategies

```
mf usercode x noopt       # Initial stage, resolution of compilation errors
mf usercode a debug       # Next stage, resolution of execution-time errors
mf usercode x noopt       # Next stage, refinement of outputs, algorithms
mf usercode a opt1/2/3/4  # Production stage. Set optimization as high as
                          # possible.
clean usercode            # Make it tidy after production is finished.
```

The above is an excerpt from the `README.general` file distributed with the system. This strategy has been found to be the most efficient way of developing a usercode.

In the initial stage of compilation, typing and syntax errors are being trapped by the compilers. Therefore, one wants the compilations to be as quick as possible and the fastest route is via the `x noopt` switch set. (`noopt` is the default.) The `x` switch plays an important role. For the Mortran3 compilation, the entire concatenated `.mortjob.mortran` is always compiled according to the configuration file. This is done because Mortran3 code segments are "non-local" in character and a change to a macro in one code segment can change how another code segment compiles. However, Fortran does not have this feature and the `x` switch causes the Fortran output (initially in one big file) to be split into separate subroutines, functions, block data, and main routines. These are then Fortran-compiled individually and the source and object code is stored in a separate workarea. Although this takes a little time on the first compilation, subsequent compilations are executed much more quickly as Fortran code segments that have not changed are not-recompiled.

After the parsing errors have been resolved, execution-time errors must be corrected. In this case the `a debug` switches are invoked. Symbolic debugging code is introduced to allow the use of `dbx` and array bounds are also checked during execution. The `a` is employed because the debuggers must reference one source file only.

After execution time errors are resolved, it is time for refinement of the algorithms and small changes (to make outputs prettier, for example). In this case `x noopt` keeps compilation times down as splitting is invoked again and the Fortran compilation is not optimised.

Finally the production stage arrives. The `a opt1/2/3/4` switches should be invoked with optimisation as high as possible in practical terms. The `a` should be employed because some optimisers can only optimise a single piece of code. Also, this switch allows the code segments to be positioned in an optimal way. For example, if `subroutine X` makes repeated calls to `function Y` in the middle of a heavily used loop, position `function Y` close to `subroutine X` in the source code. The object codes for these routines will be placed in close proximity by the compiler saving cache flushes during execution. A rule of thumb for optimisation levels: `opt1` gains 0–50%, `opt2` gains 0–20%, `opt3` gains 0–5% and `opt4` gains 0–3%, depending on the code and the compiler. However, higher levels of compilation takes more CPU time. For production runs that last only an hour, an `opt2` compilation that takes 5 minutes may be sufficient. If the production run is expected to last a month, then squeezing out that extra 1% using `opt4` may be worth a 30 minute compile time.

3.2 Running using `egs4_run`

Having successfully compiled a usercode, execution is accomplished with the statement:

```
$HEN_HOUSE/egs4_run [usercode] [input_file|""] [pegs_data_file] {debug|""}
```

Some useful mnemonics(aliases) made by `Cshrc_additions_for_egs4` are:

```
ex  == $HEN_HOUSE/egs4_run
exb == $HEN_HOUSE/egs4_batch_run     # same as above but in batch
```

The meanings of the arguments are as follows:

`usercode` Name of user code executable: Either `~/egs4/bin/sparc/[debug_]usercode.exe` if it exists, or else `$HEN_HOUSE/bin/sparc/usercode.exe`. Note that if debug mode is set the executable is called `debug_usercode.exe`. There are no defaults for this argument.

`input_file` Input file name: `~/egs4/usercode/input_file.egs4inp`. This is a filename associated with Fortran unit 5 which may or not be employed by the usercode. If it is not, then this argument must be supplied as a set of empty double quotes. If the unit number is employed by the usercode but either empty double quotes or a filename that does not exist is supplied, then input to Fortran unit 5 will come from the keyboard. There are no defaults for this argument.

`pegs_data_file` PEGS4 data file name: Either `~/egs4/pegs4/data/pegs_data_file.pegs4dat` if it exists, or else `$HEN_HOUSE/pegs4/data/pegs_data_file.pegs4dat`. The PEGS4 datafile is opened as Fortran unit 12 by the EGS4 system. There are no defaults for this argument.

`debug` This is an optional argument to invoke the symbolic debugger, usually `dbx`. In this case, the usercode would have had to have been compiled using the `debug` compilation switch described previously. The default is to run without debugging.

`egs4_run` makes some assumptions about Fortran unit numbers. Fortran unit 5 is employed for input, either from the keyboard or a file, as described above. Fortran units 0 and 6 are employed as the usual unix stderr and stdout. For batch processing, these outputs are redirected to a file with name `input_file.egs4log`. Fortran unit 8 is assigned to `/dev/null`, the "bit dump" for unix. Fortran unit 12 is assigned for PEGS4 datafile input.

If the user wants to employ other Fortran unit numbers, or wishes to add operations before and/or after execution of the usercode, then one can define a csh script file called `~/egs4/usercode/usercode.environment`. This is a file that is "sourced" by `egs4_run` both before and after execution. Before execution it is called with a unix variable called `user_operation` assigned the value `setenv`. After execution `user_operation` is assigned the value `unsetenv`. This gives the user the ability to detect whether or not the execution is about to begin or has just completed and perform operations accordingly. There are several other unix variables that are set by `egs4_run`. The most important are:

```
$usercode  = 1'st argument to egs4_run (executable, example tutor1)
$inputfile = 2'nd argument to egs4_run (input file name, without the .egs4inp)
$pegsfile  = 3'rd argument to egs4_run (PEGS4 file name, without the .pegs4dat)
```

For example, if a user wanted to define an output listing to Fortran unit 1 and wanted this file called `input_file.egs4lst`, this is what the environment file ought to look like:

```
if ($user_operation == "setenv") then
    # Remove the listing file. Otherwise there can be confusion as
    # part of it may be from the previous run.
    if (-e $inputfile.egs4lst) \rm $inputfile.egs4lst
    ln -s $inputfile.egs4lst    fort.1  # Listing file
endif
```

Note that the attachment is supplied via unix links because there is no universal or standard architecture-independent method for telling Fortran about files.

Here is a simple example of the execution of the `tutor1 code`:

```
> ex tutor1 "" tutor_data
/home/nrc/egsboss/egs4_run
+-----------------------------------------------------------------+
* egs4_run - Sun/SGI/unix/csh Version 1992/11/04/19:30            *
*                                                                 *
*           Sun/SGI/unix/csh script file to run a user code       *
*           in the EGS4 code system.                              *
*                                                                 *
*           First time user, inquiries, BUGS?                     *
*           Contact alex@irs.phy.nrc.ca                           *
*           and/or                                                *
*           Alex F Bielajew                                       *
*           Institute for National Measurement Standards          *
*           National Research Council of Canada                   *
*           Ottawa, CANADA                                        *
*           K1A OR6                                               *
*           Tel: 613-993-2197                                     *
*           FAX: 613-952-9865                                     *
+-----------------------------------------------------------------+

User code................................ tutor1.mortran
Input file............................... .egs4inp
PEGS4 DATA file.......................... tutor_data.pegs4dat
DEBUG option............................. OFF
Using host............................... nrc

Type <Control-C><Return> (soft exit), or <Return> (continue) :

Using executable /home/nrc/egsboss/bin/sparc/tutor1.exe ...
# The executable was not found in the user's EGS4 binary
# depository, but it was found in the system's repository

No environment file. Continuing with execution...
# The tutor codes do not require environment files

Directory /home/nrc/egsuser/egs4 does not exist. Creating it.
# This directory did not exist, so it was created

Directory /home/nrc/egsuser/egs4/tutor1 does not exist. Creating it.
# This directory did not exist, so it was created.
# If the directory is left empty it is deleted afterwards.

Changing directory to /home/nrc/egsuser/egs4/tutor1
/home/nrc/egsuser/egs4/tutor1

Using pegsfile /home/nrc/egsboss/pegs4/data/tutor_data ...
```

```
Running tutor1::tutor_data on nrc ...

No input file specified.
Any input should come from the keyboard.

1  START TUTOR1

 CALL HATCH TO GET CROSS-SECTION DATA

 RAYLEIGH DATA AVAILABLE FOR MEDIUM  1 BUT OPTION NOT REQUESTED.

 EGS SUCCESSFULLY 'HATCHED' FOR ONE MEDIUM.
OKNOCK-ON ELECTRONS CAN BE CREATED AND ANY ELECTRON FOLLOWED DOWN TO
                                          0.189 MeV KINETIC ENERGY
   BREM PHOTONS CAN BE CREATED AND ANY PHOTON FOLLOWED DOWN TO
                                          0.010 MeV
0                 KINETIC ENERGY(MeV)  CHARGE  ANGLE W.R.T. Z AXIS-DEGREES
 START HISTORY   1
                         1.537           0            20.5
                        15.846          -1            19.0
 START HISTORY   2
                         0.797           0            13.7
                        17.443          -1            22.5
 START HISTORY   3
                         5.995          -1             7.8
                        12.199           0            20.8
 START HISTORY   4
                         0.421           0             5.2
                         1.795           0            13.5
                        15.602          -1            14.9
 START HISTORY   5
                         7.636          -1            71.5
                        10.388           0            33.5
 START HISTORY   6
                         3.018           0             2.2
                        14.360          -1            21.4
 START HISTORY   7
                         0.347           0            38.3
                         1.925           0             5.5
                        15.857          -1             9.9
 START HISTORY   8
                         4.981           0             8.0
                        12.952          -1            15.9
 START HISTORY   9
                         0.352           0            18.4
                        17.802          -1            19.4
 START HISTORY  10
                         0.538           0            40.5
```

```
                        0.118            0            51.8
                       16.610           -1            57.7
 Warning: nonstandard floating-point arithmetic mode
 was enabled in this program and was never cleared.
# Ignore, this is always produced.

Execution completed successfully.
```

Execution takes place on a subdirectory to `~/egs4/usercode` that has the form `n_input_file_hostname` where `n` is a unique process number, `input_file` is the second argument to `egs4_run` and `hostname` is the hostname of the machine on which the execution takes place. Some of this structure foreshadows a future parallel version of EGS4. The working directory contains all the necessary files for execution, keeping files separate until completion when they are copied back. This allow users to employ the original files on the parent directory. However, changes to these files will be lost after the execution in the working directory completes.

3.3 Making PEGS4 datafiles

PEGS4 data files are created with the command:

```
$HEN_HOUSE/pegs4/pegs4_run [pegs_input_file]
```

where `pegs_input_file` is the name of a PEGS4 input file with the name `pegs_input_file.pegs4inp` on the current working directory. (Do not include the `.pegs4inp` on the command line.) The syntax of PEGS4 input files is described elsewhere.

Some useful mnemonics(aliases) made by `Cshrc_additions_for_egs4` are:

```
pegs4 == $HEN_HOUSE/pegs4/pegs4_run
pegs4b == $HEN_HOUSE/pegs4/pegs4_batch_run # Do it in batch
```

3.4 Standalone Mortran3/Fortran compilation

Given a standalone (separate from the EGS4 system) Mortran3 source code, called `usercode.mortran`, the Mortran3 and Fortran compilers are called with the statement:

```
$HEN_HOUSE/mortran3/mortran3 [n|e] [usercode] {[nodebug]|debug|DEBUG} {[n]|quickm}
```

The interpretation of the arguments is as follows:

`n|e` If the choice is `n` the Mortran3 source is compiled standalone. If the choice is `e`, the EGS4 macros set `$HEN_HOUSE/egs4mac.mortran` is placed before the source. Although the source code is considered to be independent of the EGS4 system, there are some nice macros defined in `$HEN_HOUSE/egs4mac.mortran` that are of general use.

`usercode` A source file on the current working directory called `usercode.mortran`. The `.mortran` should not be included on the command line.

`debug status` With the default, `nodebug`, Mortran3 compiles and then invokes level 3 optimisation on the Fortran step. `debug` invokes debugging switches on the Fortran compilation step. `DEBUG` invokes run-time debugging on the Mortran3 compiler (for debugging compiler errors).

`n|quickm` The default, `n`, performs Mortran3 followed by Fortran, as described above. `quickm` is employed by `egs4_compile` which signals the compilation to begin immediately and to bypass the Fortran compilation. (`egs4_compile` invokes the Fortran compiler on its own).

Some useful mnemonics(aliases) made by `Cshrc_additions_for_egs4` are:

```
mor   == $HEN_HOUSE/mortran3/mortran3 n   # standalone Mortran3 compile
emor  == $HEN_HOUSE/mortran3/mortran3 e   # Mortran3 compile with EGS4 macros
morb  == $HEN_HOUSE/mortran3/mort3bat n   # standalone Mortran3 compile in batch
emorb == $HEN_HOUSE/mortran3/mort3bat e   # Mortran3 compile with EGS4 macros in batch
```

The Mortran3 compilation generates the outputs `usercode.mortlst`, the Mortran3 compile listing, `usercode.f`, the Fortran output and `usercode.exe`, the executable.

3.5 Batch processing

Batch compilation of EGS4 usercodes is accomplished with the script `$HEN_HOUSE/egs4_batch_compile` and execution with `$HEN_HOUSE/egs4_batch_run`. Mnemonics for employing these scripts have been described above. (Just append a "`b`" to any of the mnemonics for interactive processes.) Both the immediate batch queue and deferred batch queues may be defined for these processes. To access the deferred batch processor, the fourth and higher arguments to the mnemonics (not the scripts!) are passed through as time arguments for unix `cron`. (Do a man cron for explanation of the syntax.) Note that if not all the compiler switches are used, these positions musted be filled with empty double quotes `""`'s. To access the immediate batch processor, simply do not supply any arguments to the mnemonics. Here's are two examples of delayed batch processing:

```
> mfb dosrz "" "" Midnight Sep 16
job 0001 at Thu Sep 16 00:00:00 1993

> exb dosrz inputfile pegsfile Noon Sep 17
job 0002 at Fri Sep 17 12:00:00 1993
```

Immediate batch processing would be accomplished by leaving the deferred time arguments off the above statements. The batch processes notify the user of their completion with mail messages.

The PEGS4 and Mortran3 standalone processing have immediate batch capability as well. Deferred batch processing is not supplied in these cases.