

# CERES Software Bulletin 96-05

November 4, 1996

## Environment Variables, Makefiles, Scripts, and PCF Files

Jill Travers (d.j.travers@larc.nasa.gov)  
Maria Mitchum (m.v.mitchum@larc.nasa.gov)  
Sandy Nolan (s.k.nolan@larc.nasa.gov)

### Purpose:

This software bulletin is designed to document known requirements for environment variables, makefiles, scripts, and PCF files which will be delivered to the DAAC. If requirements change, this bulletin will be updated and redistributed. The PGE Design Information Web Page (URL-[http://ecsinfo.htic.com/iteams/pge\\_wp.html](http://ecsinfo.htic.com/iteams/pge_wp.html)) maintained by ECS provides additional information on the ECS environment and PGE requirements. This bulletin combines DAAC SSI&T requirements with a summary of the ECS system requirements for the topics below.

### Introduction:

During the CERES Release 1 Science Software Integration and Test Campaign at the DAAC, it was noted that a consistent set of requirements for the use of environment variables, makefiles, scripts, and PCF files was needed. The CERES System Engineering Committee has put together a list of these requirements which will assist in a smooth transition from the SCF to the DAAC environment.

### Environment Variables:

The environment variable issue is divided into two parts; the first part is an issue for using environment variables during SSI&T and then using environment variables in the production environment.

The ceres-env.csh file will set environment variables used during SSI&T. These will include CERESHOME, CERESBIN, CERESLIB, CERESDATA, CERESSRC, and CERESTEST. These variables will be built using a variable called WORKING\_GROUP which will be previously set to the CERES working group currently being handled. It should be noted that while data and binaries are currently located under \$CERESHOME, due to requirements at the DAAC this might not always be the case. The CERESBIN environment variable should be used in the Makefile to move the executable to the bin directory. The ceres-env.csh file will also contain environment variables for compiler locations and options which should be used in the Makefiles delivered to the DAAC. Included below is the current ceres-env.csh file used at the DAAC.

```
#-----#  
# Source file to set environment variables #  
# for CERES build and execute procedures #
```

```

#                                     #
# Name - ceres-env.csh                #
# Date - March 14, 1996               #
# Update - October 20, 1996          #
#-----#

#-----#
# CERES home and CERESLIB directories #
# Must be inside a ClearCase view to #
# see /CERES directory contents     #
#-----#
setenv CERESHOME /CERES
setenv CERESLIB $CERESHOME/lib
setenv CERESBIN $CERESHOME/$WORKING_GROUP/bin
setenv CERESDATA $CERESHOME/$WORKING_GROUP/data
setenv CERESSRC $CERESHOME/$WORKING_GROUP/src
setenv CERESTEST $CERESHOME/$WORKING_GROUP/test_suites

#-----#
# Compiler location and default flag #
#-----#
setenv CC      /usr/bin/cc
setenv CFLAGS  '-32 -O'
setenv NAGF90  /usr/local/bin/f90
setenv SGIF90  /usr/bin/f90
setenv F90     $NAGF90
setenv F90COMP '-ccarg "-32" -O -c'
setenv FCOMP   '-ccarg "-32" -O w -c'
setenv F90LOAD '-ldarg "-32" -s'
setenv ADA     /usr/vads/bin/ada

#-----#
# Compiler debug flags                #
# (Uncomment these flags to compile in debug mode) #
#-----#
#setenv CFLAGS  '-32 -g'
#setenv F90COMP '-ccarg "-32" -g -C=all -c'
#setenv FCOMP   '-ccarg "-32" -g -C=all -w -c'
#setenv F90LOAD 'F90LOAD -ldarg "-32" -g'

```

Environment variables defined by the SDP toolkit include PGSHOME, PGS LIB, PGSMSG, PGSINC, HD FLIB, HD FINC. The toolkit definitions will be used at the DAAC and are therefore not included in the ceres-env.csh file. The PGS and HDF environment variables defined by the toolkit should be used in each Makefile delivered to the DAAC to locate the toolkit and HDF libraries, the toolkit message files and include files.

Environment variables should not be set by the PGE in the production environment unless it is absolutely necessary. Before using environment variables, it would be wise to check with the DAAC/ECS to make sure there are no problems. Possible risks of setting environment variables in the PGE include the risk of name collision with the data processing system and running out of free environment variable name space since the number of environment variables that can be defined is limited. If possible a safer mechanism is to use the runtime parameters capability of the toolkit. If environment variables must be used then:

- limit the number to the absolute minimum
- Choose a naming convention to ensure that names will not collide with those used by the data processing system.
- Do not reset any environment variable used by the SDP Toolkit or typical user environment variables (i.e PATH, HOME, USER, DISPLAY, etc.).

### **Makefiles:**

Makefiles are requested to be delivered to the DAAC as a mechanism to correctly build the science software. Makefiles are preferred over build scripts in order to enable the DAAC to use the Clearmake utility of the ClearCase Configuration Management Software. Clearmake keeps a build record of which version of individual software components were used to create a particular executable. The following Makefile requirements should be followed:

- All Subsystems coded in Fortran 90 should use the makemake utility to build Makefiles. Joe Stassi has modified makemake to use correct environment variables and links for the DAAC environment.
- Other Subsystems should use the approved environment variables from above in specifying compiler and library path names.

### **Scripts:**

Shell scripts may be delivered which wrap together one or more executable programs for processing. Shell scripts in general allow great freedom in the tasks they perform. In general, all of the standard UNIX utilities are available to shell scripts; however, some UNIX commands can be problematic to the DAAC production environment and should not be used. The following is a list of known script requirements:

- Command line arguments may not be used to invoke your outermost script. Arguments may be used to invoke subsequent scripts; however, these arguments would in fact be static.
- Do not create PCF files on the fly from within your scripts. A separate script for generating PCF files at the SCF would be advisable. See PCF section below for additional information.
- Do not build your executable in your script. Builds will take place during SSI&T and the executables will be promoted to production.
- PGEs cannot be interactive; do not attempt to access or use standard input, standard output, or standard error.
- PGEs should not redirect output to a file unless you have specified that file within the PCF file. See PGE Design Information Web page for details on how to accomplish this.
- ECS provides the management of files and file systems. Therefore, PGEs should not attempt to change directories, create or remove directories, create or remove files (except via SDP

Toolkit calls), rename files or directories, copy or move files and directories, or alter ownership and permissions on files and directories.

- ECS manages and controls processes. PGEs should not attempt to kill processes, set up cron jobs, or change process priorities.
- Network connectivity is handled by ECS. PGE shell scripts should not circumvent this by initiating network connections to transfer files to or from a remote site (ftp), send or receive electronic mail, make socket connections, or use remote procedure calls (RPC).
- PGE shell scripts should not assume a particular directory structure. The SSI&T directory structure may not be compatible with the production environment. Therefore, PGE scripts should not check for the existence of files in particular directories and should not set or source environment variables that assume a particular directory structure.

### **PCF Files:**

Process Control Files (PCF) are used to supply input and output file information and runtime parameter information to the PGE. The following are known PCF requirements:

- The PCF template, PCF.relA.template, should be used for the CERES Release 2 deliveries.
- All toolkit supplied information in the template should not be deleted. These are logical id numbers 10000 or greater.
- There is only one PCF file allowed per PGE. Remember a PGE is defined as the smallest schedulable science software unit. Therefore a single PGE may contain several FORTRAN, C, or ADA executable programs wrapped within a shell script.
- PCF files are created and pointed to prior to execution in the production environment; therefore, PCF files must not be generated or environment variables set to point to a particular PCF in your shell scripts.
- Add input/output files in the appropriate area (i.e. Product Input/Output, Support Input/Output, Intermediate Input/Output, and Temporary).
- Be careful to add runtime parameters only as necessary. At the time of this writing the use of runtime parameters is under discussion. **Runtime parameters are not dynamic.** Default runtime parameter values are set during SSI&T by registering a separate PGE for each set of runtime parameter values, and can only be changed with manual operator intervention.