CERES Software Bulletin 95-14

Error/Status Message file Creation and the ceresutil Utility

Alice Fan(t.f.fan@larc.nasa.gov)
Joe Stassi (j.c.stassi@larc.nasa.gov)

1.0 Purpose

This bulletin explains the steps required to create error/status message files for PGE processing. The message files contain PGE-specific error and information messages to be made available during PGE code compilation and execution.

2.0 Error/Status Message Files

There are three types of error/status messages files: the Message Text File, the Include File, and the Runtime File.

(1) **`Message Text File`** (".t" extension): This is a user-created file. It contains PGE error messages along with the mnemonic labels used in the code to access the messages. The Include File and the Runtime File are generated from the Message Text File, and therefore it is important that this file be correctly formatted. Specifics for creating the Message Text File are given in the Section 3.0.

(2) **`Include File`** (".f" extension): This file is generated from the Message Text File using the ceresutil utility. The Message Text File is "included" within the PGE code during compilation. It contains the mapping between the error message mnemonic labels and the system-generated error message numbers.

(3) **`Runtime File`** (no extension): The Runtime File is also generated from the Message Text File by the ceresutil utility. The Toolkit library procedures access this file during processing in order to map the message numbers back to the error/status messages.

In summary, the Message Text File provides information necessary for the creation of the Include File, which is required for code compilation, and the Runtime File, which is used during program execution.

3.0 Creating a Message Text File

The Message Text File can be created with any editor. It must contain 3 specific variable definitions and can define up to 510 error/status messages. When creating this file, you should give it the .t extension in its name to identify it as a text file.

A sample Message Text File is shown in Figure 3-1. The file, CLOUD_25450.t, was created for the Cloud subsystem of the CERES project.

A more detailed explanation of the Message Text File follows in Sections 3.1 and 3.2.

```
#
# A message Text file for CERES CLOUD
#
%INSTR  = CERES
%LABEL  = CLOUD
%SEED   = 25450
CLOUD_E_UNABLE_ALLOCATEMEMORY  ERROR...Unable to allocate memory
CLOUD_E_GETRUNTIMEPARAM_ERROR  ERROR...in getting run-time parameter
CLOUD_W_ALLOCATECIDFILE_UNABLE WARNING...unable to allocate CIDFile:
CLOUD_W_GETFILENAME_UNKNOWN    WARNING...in getting file name:
CLOUD_W_INQUIREFILE_NOTEXIST   WARNING...file does not exist:
CLOUD_E_GETFILENAME_ERROR      ERROR...in getting file name:
CLOUD_E_INQUIREFILE_ERROR      ERROR...file does not exist:
#
CLOUD_E_OPENFILE_UNABLE        ERROR...in opening file:
```

Figure 3-1. Sample Message Text File: CLOUD_25450.t

3.1 Variable Definitions

The first required entries in the Message Text File are three variable definitions which must appear in this exact order: %INSTR, %LABEL, and %SEED.

(1) %INSTR = CERES
This variable identifies the instrument name. Its value should be set to CERES for all PGEs from the CERES project.

(2) %LABEL = CLOUD
This variable identifies the subsystem, the module, or whatever best describes what the error/status messages are associated with. The label's value must consist of 3 to 10 uppercase letters.

(3) %SEED = 25450
This is the seed number for the error messages in this file. Each seed can have up to 510 error messages associated with it. The CERES project has been assigned 4000 seeds. A breakdown of seed assignment by subsystem number is given in Appendix A.

3.2 Error/Status Message Definitions

After the variable definitions come the message definitions. Each definition consists of two parts: the mnemonic label and the message string.

(1) Mnemonic Label
The mnemonic label itself consists of different components separated by underscores. Here's a breakdown of the first label listed in the sample "CLOUD_E_UNABLE_ALLOCATEMEMORY."

a) "CLOUD" : The first component of the label must correspond exactly with the value of the %LABEL variable listed at the top of the file.

b) "_E_" : This letter identifies the severity level of the message. See Appendix B for a listing of the various severity levels.

c) "UNABLE_ALLOCATEMEMORY" : This is the mnemonic portion of the label which contains information about the error. This portion consists of up to 30 uppercase letters and underscores.

The mnemonic label gets stored in the Include File. It is used within the code to identify which error message is to be written to the report file(s).

(2) Message String
The message string in our example is "ERROR...Unable to allocate memory."

a) This string should be as informative as possible since it is what will be written to the report file(s).

b) The string may span many lines and can include up to 240 ASCII characters.

c) Any white space within the string will be reduced to a single space.

4.0 Using ceresutil to Create Include and Runtime Files

The process of creating Include and Runtime Files from the Message Text File has been automated in a utility called "ceresutil." Ceresutil is a script file which uses the smfcompile procedure from the Toolkit. An example of running ceresutil is found in Appendix C. The ceresutil file can be found on the $CERESLIB directory.

4.1 Examples

For the Message Text File in Figure 3-1, the ceresutil utility creates an Include File called "PGS_CLOUD_25450.f" and a Runtime File called "PGS_CLOUD_25450," displayed in Figures 4-1 and 4-2, respectively.

```
INTEGER CLOUD_E_UNABLE_ALLOCATEMEMORY
PARAMETER (CLOUD_E_UNABLE_ALLOCATEMEMORY = 208489984)

INTEGER CLOUD_E_GETRUNTIMEPARAM_ERROR
PARAMETER (CLOUD_E_GETRUNTIMEPARAM_ERROR = 208489985)

INTEGER CLOUD_W_ALLOCATECIDFILE_UNABLE
PARAMETER (CLOUD_W_ALLOCATECIDFILE_UNABLE = 208489475)

INTEGER CLOUD_W_GETFILENAME_UNKNOWN
PARAMETER (CLOUD_W_GETFILENAME_UNKNOWN = 208489476)

INTEGER CLOUD_W_INQUIREFILE_NOTEXIST
PARAMETER (CLOUD_W_INQUIREFILE_NOTEXIST = 208489477)

INTEGER CLOUD_E_GETFILENAME_ERROR
PARAMETER (CLOUD_E_GETFILENAME_ERROR = 208489990)

INTEGER CLOUD_E_INQUIREFILE_ERROR
PARAMETER (CLOUD_E_INQUIREFILE_ERROR = 208489991)

INTEGER CLOUD_E_OPENFILE_UNABLE
PARAMETER (CLOUD_E_OPENFILE_UNABLE = 208489992)
```

Figure 4-1. Sample Fortran Include File: PGS_CLOUD_25450.f

Our example is a Fortran example, and therefore the Include File is a Fortran Include file. The Runtime File is not language dependent.

```
CERES,CLOUD,25450
208489984,CLOUD_E_UNABLE_ALLOCATEMEMORY,NULL,ERROR...Unable to allocate memory
208489985,CLOUD_E_GETRUNTIMEPARAM_ERROR,NULL,ERROR...in getting run-time parameter:
208489475,CLOUD_W_ALLOCATECIDFILE_UNABLE,NULL,WARNING...unable to allocate CIDFile:
208489476,CLOUD_W_GETFILENAME_UNKNOWN,NULL,WARNING...in getting file name:
208489477,CLOUD_W_INQUIREFILE_NOTEXIST,NULL,WARNING...file not exist:
208489990,CLOUD_E_GETFILENAME_ERROR,NULL,ERROR...in getting file name:
208489991,CLOUD_E_INQUIREFILE_ERROR,NULL,ERROR...file not exist:
208489992,CLOUD_E_OPENFILE_UNABLE,NULL,ERROR...in opening file:
```

Figure 4-2. Sample Runtime File: PGS_25450

4.2 Where to put Include and Runtime Files

Include Files are referenced within the PGE source code. They are “included” in those routines which need access to the error message labels. The Include Files must be visible to the compiler during compilation. This can be done by placing the Include Files within the source code directory during compilation, or else by putting them in an Include directory which is made visible to the compiler by compilation flags.

The Toolkit library uses the Runtime Files during execution and expects to find them in a directory defined by the PGSMSG environment variable. In the development stages of the CERES subsystem code, analysts must create their own message directory to contain PGE-specific message Runtime files. Any general system Runtime files required by the PGE must also be copied to this directory from the $PGSDIR/message directory. Prior to executing your PGE, define the PGSMSG environment variable to point to your message directory. Appendix D gives a sample script file which runs a cloud subsystem PGE. It contains several environment variable definitions, including one for the PGSMSG variable which points to the $HOME/Tk_msgdir directory.

Appendix A : Seed Number

CERES has seed numbers 25000 - 29000. Each subsystem gets 100 seeds.
Each seed can contain up to a maximum of 510 error messages.
Each subsystem has 510 x 100 = 51,000 possible error messages.

		(examples)
system	25000_25099	LABEL = CERESxxx
subsystem		
1	25100 - 25199	LABEL = INSTRxxx
2	25200 - 25299	
3	25300 - 25399	
4.1 - 4.3	25400 - 25499	LABEL = CLOUDxxx
4.4	25500 - 25599	
4.5 - 4.6	25600 - 25699	
5	25700 - 25799	
6	25800 - 25899	
7.1	25900 - 25999	
7.2	26000 - 26099	
8	26100 - 26199	
9	26200 - 26299	
10	26300 - 26399	
11	26400 - 26499	
12	26500 - 26599	

Appendix B : Error Message Severity Levels

Label	Level	What happens after message output?
S:	Success	returns to calling procedure
M:	Message	“
U:	User Information	“
N:	Notice	“
W:	Warning	“
E:	Error	terminates processing
F:	Fatal	“

Appendix C : Using the ceresutil to access smfcompile

```
fan@saisun15 % ceresutil

 CERES UTIL
-------------


 1. Compile message status file
 2. Check PCF file correctness

 3. Create a message File - not ready yet
 Q. Quit

 Please enter option: 1

 Enter MESSAGE file name ( xxxxx.t) ---> CLOUD_25450.t

 Enter language( ada, f, c, all) ---> f
Successful operation on SMF file <CLOUD_25450.t>
Files created:
      Fortran file: PGS_CLOUD_25450.f
      ASCII message file: PGS_25450

 Exiting CERES UTILITY, Goodbye !
```

Appendix D : Run script example which shows PGSMSG variable definition

```
#!/bin/csh
##########################################################
# need to source pgs-dev-env.csh before any following setenv
# else Toolkit will overwrite all your set up
##########################################################
setenv PGSDIR /opt/net/PGSTKV5
source $PGSDIR/bin/pgs-dev-env.csh
set path= ($path $pgs_path)

setenv PGS_PC_INFO_FILE $HOME/f90_cld/PCfile
setenv PGSMSG $HOME/Tk_msgdir

setenv PGS_PRODUCT_INPUT  $HOME/product_in
setenv PGS_PRODUCT_OUTPUT $HOME/product_out

setenv PGS_SUPPORT_INPUT  $HOME/support_in
setenv PGS_SUPPORT_OUTPUT $HOME/support_out

setenv PGS_INTERMEDIATE_INPUT $HOME/interm_in
setenv PGS_INTERMEDIATE_OUTPUT $HOME/interm_out

setenv PGS_TEMPORARY_IO $HOME/temp_io
##########################################################
cloud.exe
```