

CERES Software Bulletin 97-11

October 7, 1997

PCF Generator Approach

Maria Vallas Mitchum (m.v.mitchum@larc.nasa.gov)
Sandy Nolan (s.k.nolan@larc.nasa.gov)

1.0 Purpose

This bulletin describes the general rules for developing a Process Control File Generator, which is a required step in the Langley TRMM Information System (LaTIS) but not in the ECS system. Since each PCF is unique, this bulletin will serve as a style guideline for the individual generators. The authors thank Alice Fan, Denise Cooper, and Ricky Brown, SAIC, for their contributions and assistance in documenting this computer bulletin.

2.0 PCF Generator Function

The methodology used by a Product Generation Executive (PGE), or processor, to obtain processing information is by reading an external file called a Process Control File (PCF). The PCF contains all of the file information and runtime parameters required by a PGE. The PCF is unique for each processor and furthermore is unique for each instance of execution of each processor.

The PCF Generator is a script code tailored for each PGE to create a PCF for each instance of PGE execution. It is the responsibility of each CERES Software Developer to write a PCF Generator for each processor and supply it to the LaTIS Processing System.

3.0 LaTIS Interface

To provide one PCF for each PGE, and not to have to supply a separate PCF for each instance that must be executed, each PCF will be stripped of all dynamic parameters, dynamic file names, and directory paths and made into a template. All of the PCF dynamic run-time variable values will be determined at Production Request time, stored in the LaTIS Process Request Information database and retrieved at Production Initiation time. This information will be used by a LaTIS Preprocessor to determine that all Production Rules for a PGE have been met and that all of the input files for the PGE exist for processing to be initiated.

The information that is required for each PCF template will be written to an ASCII file by the LaTIS PGE-unique Preprocessor. The PCF Generator will read the ASCII file, as input, and populate the PCF template, contained within the generator, with all of the current information needed to run the PGE for each Production Request instance. Once the PCF template has been populated, the process can be staged for execution.

Appendix A shows a script which retrieves the information from the ASCII file and places the items of information into the appropriate PCF template locations. Appendix B shows the created

PCF file.

4.0 The PCF Generator Input File

Using the assumption that the PCF Generator input ASCII file will be created by the LaTIS system, the developer of the PCF Generator should expect the input file to contain the following information where the format of the information is:

Parameter <space> = <space> Value (example: file.1 = Infile)

Each Parameter will be a predetermined variable name and will be expected on the ASCII file, if it exists. If an Input file does not exist, the LaTIS Preprocessor may or may not, depending on the PGE rules, insert the value: NULL or blank or completely leave out the file name in the ASCII file. The Generator may need to contain logic to insure that if a NULL value, or blank, is found, that appropriate action will be taken to insure that the file is not placed on the PCF.

- 1) CERES required metadata - from production request
- 2) PGE specific runtime parameters - from production request, vary with each PGE.
- 3) PGE required directories
- 4) Input file names
- 5) Output file names

4.1 CERES Runtime Parameters

At Production Request time, a user will access the LaTIS Processing Request procedure. The following information will be entered, where applicable, and stored in the LaTIS database:

PGENAME	SamplingStrategyInput	ProductionStrategyInput
	SamplingStrategyOutput	ProductionStrategyOutput
CERDataDateYear	CERDataDateMonth	CERDataDateDay CERHrOfMonth

The LaTIS system will provide the ConfigurationCode, Software SCCR number (SWsccr), and Ancillary Data SCCR number (DATAsccr) based on the PGENAME from the Configuration Control database. The above information and PGE specific runtime parameters are used to create the input and output file names and to fulfill the CERES metadata requirements.

The following parameters are the CERES baseline metadata. The format and spelling of the parameters must appear on the PCF file, as shown in field 2, using “|” as the field delimiter.

141 PGENAME ”4.5-6.1P1”	/* from Production request */
142 SamplingStrategy ”TRMM-PFM-VIRS”	/* from Production Request */
143 ProductionStrategy ”PreFlight”	/* from Production Request */
144 CERDataDateYear ”1986”	/* from Production Request */
145 CERDataDateMonth ”10”	/* from Production Request */
146 CERDataDateDay ”01”	/* from Production Request */
147 CERHrOfMonth ”05”	/* from Production Request */
148 ConfigurationCode ”00000”	/* from LaTIS database */
149 SWsccr ”00013”	/* from LaTIS database */
150 DATAsccr ”00015”	/* from LaTIS database */
151 <ProductSpecificAttributeName> <”User’s value”>	/* from Production Request*/

4.2 PGE Directory Paths

A table association will be built in the LaTIS database for each PGE that relates that PGE's Directory Names with the respective Directory Paths. This set of tables is required to provide the LaTIS Processing System the greatest flexibility to change directory paths without affecting the internal operations of each PGE. The ASCII file should contain all the PGE required DirectoryName = DirectoryPath. The PGE Generator will then use each DirectoryName where it is applicable within the PCF template.

4.3 PGE Input Files

The Input files for a PGE are categorized as: Constant Ancillary Input Data, External Input Data, and PGE-Dependent Input Data.

1. Constant Ancillary Input Data files - are those data files which remain constant from month to month and rarely change. These are typically delivered to the LaTIS site from the Subsystem Development Team. The Ancillary Input Data files are tracked by the LaTIS CM system. There have been no restrictions made on the naming convention of these files, other than the names of the files must be a string of 80 characters or less. It is suggested that the date of the file creation be entered as part of the file name, for identification. Even though the files are described as static files, they could, in the future, be replaced by updated files. The number of files may also change. A database table will be structured in the LaTIS system to be used by all of the CERES processors to store the Constant Ancillary Input File Names, allowing the sharing of files by more than one processor. The database table will store the file names according to the Data Set Shortnames, the Product-ID (an abbreviated Data Set Name), the number of files, and the actual file names. The Data Set Shortname will be provided to the LaTIS Developers by the Subsystem Development Team for each PGE Preprocessor, to retrieve the file information.

For example, DataSetName = CBANALAA, used in Subsystem 1, the Product-ID is 'BANAL' and the number of files = 2, the ASCII file would contain:

```
BANAL.1 = Instrument_Engineering_Limits.19980203  
BANAL.2 = Elevation_Scan_Offsets.19971212
```

2. External Input Data - is data received at the LaTIS DAAC from other DAACs or other processing Data Centers. If any special database tables of information need to be built and populated by the Processing System at the time of ingestion, they will be designed and created for each specific External Data Set. The filenaming convention will depend on the data provider and the LaTIS system.

For example, Subsystem 2 will require the snow data from NSIDC which comes in the following format: f10_iwva_YYDDD_dayAD.hdf.gz where YYDDD is the year and day of the year. The ASCII file would contain:

```
Inputf10.1 = f10_iwva_97350_dayAD.hdf.gz
```

3. PGE-Dependent Input Data Files - are those data files that have been generated by an earlier

PGE processor. The specific PGE-Dependent input Data Set Names (Shortnames) will be identified for each PGE Preprocessor. It will be the task of the Processing System to provide the PGE the knowledge of the existence of an input file and the AutomaticQualityAssessment value of the desired input file. Given the Data Set Name, SamplingStrategyInput, ProductionStrategyInput, and the appropriate instance qualifiers (such as year, month, day, hour), the LaTIS system will determine the existence and status of the input file. As stated above in section 4.0 and restated here for emphasis ‘If an Input file does not exist, the LaTIS Preprocessor may or may not, depending on the PGE rules, insert the value: NULL or blank or completely leave out the file name in the ASCII file. The Generator may need to contain logic to insure that if a NULL value, or blank, is found, that appropriate action will be taken to insure that the file is not placed on the PCF.’

For example, for Subsystem 4.5-6.1P1, a typical input data file would be:

```
SS4.5_Inputfile.1 = CER_SSFI_TRMM-PFM-VIRS_Edition1_00012.1998100105
```

4.4 PGE Output Files

The output filenames expected to be produced by a processor will be structured in the Preprocessor and written to the ASCII file. The naming convention will use the Runtime Parameters as the input values to be used in the construction of each output filename.

For example for Subsystem 4.5-6.1P1, a typical output data file would be:

```
SS4.5_Outputfile.1 = CER_SSFB_TRMM-PFM-VIRS_Edition1_00015.1998020105
```

4.5 Sample ASCII File

A sample ASCII file is listed below, using the PGE 4.5-6.1P1 requirements.

```
#####
# CERES baseline Metadata
#####
PGName = 4.5-6.1P1
SamplingStrategy = TRMM-PFM-VIRS
ProductionStrategy = PreFlight           #used during Prelaunch testing period
CERDataDateYear = 1986
CERDataDateMonth = 10
CERDataDateDay = 01                     #some PGEs do not have value for data day
CERHrOfMonth = 05                       #some PGEs do not have value for hour
ConfigurationCode = 00000
SWsccr = 00013
DATAAsccr = 00015
TK_Ver = "SCF B.0 TK5.2"

#####
# PGE specific runtime parameters
#####
```

```
Satellite_Instrument = NOAA9/CERES      # required by CERESlib routine: qcheader
Ancillary_Data_Set = ERBE_ADMs          # required by CERESlib routine: qcheader
SP_MODEL_NUM = 1
RUN_SURF_ALG = 1
PGE_VERSION = 2.0
CERHrOfDay = 05
```

```
#####
```

```
# PCF required directories
```

```
#####
```

```
SS4.5_InputDir.1 = /CERES/clouds/data/out_comp/data/SSF_Int
SS4.5_InputDir.2 = /CERES/clouds/data/out_comp/QA_Reports
SS4.5_InputDir.3 = /CERES/sarb/data/out_comp/data/regridmoa/release2
SS4.5_InputDir.4 = /CERES/shared_data
SS4.5_InputDir.5 = /CERES/inversion/data/ancillary/static
SS4.5_OutputDir.1 = /CERES/inversion/data/out_comp/data
SS4.5_OutputDir.2 = /CERES/inversion/data/out_comp/QC
SS4.5_LogsDir = /CERES/inversion/data/runlogs
SS4.5_MCFDir = /CERES/inversion/rcf
SS4.5_TempDir = /CERES/inversion/data/scr
```

```
#####
```

```
# Input file names
```

```
#####
```

```
SS4.5_Inputfile.1 = CER_SSMI_TRMM-PFM-VIRS_PreFlight_00000.1986100105
SS4.5_Inputfile.2 = CER_FQCI_TRMM-PFM-VIRS_PreFlight_00000.1986100105
SS4.5_Inputfile.3 = CER_MOA_TRMM-PFM_PreFlight_00000.1986100105
SS4.5_Inputfile.4 = IISCTR.093097
SS4.5_Inputfile.5_1 = NIISW03.093097
SS4.5_Inputfile.5_2 = NIILWAT.093097
SS4.5_Inputfile.5_3 = NIILWWN.093097
SS4.5_Inputfile.5_4 = NIILWSP.093097
SS4.5_Inputfile.5_5 = NIILWSM.093097
```

```
#####
```

```
# Output file names
```

```
#####
```

```
SS4.5_Outputfile.1 = CER_SSMI_TRMM-PFM-VIRS_PreFlight_00000.1986100105
SS4.5_Outputfile.2 = CER_GQCA_TRMM-PFM-VIRS_PreFlight_00000.1986100105
SS4.5_Outputfile.3 = CER_GQCI_TRMM-PFM-VIRS_PreFlight_00000.1986100105
```

```
#####
```

```
# Log file names (Optional - if user wishes to store each log file)
```

```
#####
```

```
SS4.5_Logsfile.1 = LogStatus_19861001_05
SS4.5_Logsfile.2 = LogReport_19861001_05
```

SS4.5_Logsfile.3 = LogUser_19861001_05

Note: Under #PGE specific runtime parameters, I have included some of the Runtime Parameters required by the CERES qheader CERESlib routine (see below):

! Information located on the PC file
!
! If any of the following Logical Identifiers are specified in the
! USER DEFINED RUNTIME PARAMETERS section in the PC file,
! the corresponding value (3rd field) will be retrieved by the
! WriteHeader CERESlib subroutine and written to the QC report.
!
! 9991|Subsystem |4.4/Cookie Cutter
! 9992|CERES ID |Cookie Dough
! 9993|Software Version |2.1
! 9994|EOSDIS ID |CER11
! 9995|Satellite Instrument |TRMM-PFM
! 9996|Ancillary Data Set |GOES-E, Meteosat

Note: At this time the LaTIS system has not matured to the point that they can provide the PGE Generator code the necessary ASCII input file. The CERES developers have written a simple script to produce the ASCII file dynamically, see Appendix C for reference.

5.0 PCF Generator Example Using a Simulated Input File

An example PCF Generator script, `inv_pcfgen.csh`, for Subsystem 4.5-6.1P1 is located in Appendix A of this bulletin and a copy of the generated PCF file, `PCF_4.5-6.1P1_TRMM-PFM-VIRS_PreFlight_00000_1986100105`, is located in Appendix B.

Appendix C contains a script, `inv_ascii_gen.csh`, which is used to simulate the LaTIS PGE-unique Preprocessor and creates an ASCII file, `inv_ascii_TRMM-PFM-VIRS_PreFlight_00000_1986100105`, which is input to the PCF Generator. The generation of the ASCII file will be a LaTIS function, and is provided here for testing the PCF Generator. The contents of `inv_ascii_TRMM-PFM-VIRS_PreFlight_00000_1986100105` are found in Section 4.5 of this bulletin.

Electronic copies of files, `inv_pcfgen.csh`, `PCF_4.5-6.1P1_TRMM-PFM-VIRS_PreFlight_00000_1986100105`, `inv_ascii_gen.csh`, and `inv_ascii_TRMM-PFM-VIRS_PreFlight_00000_1986100105`, can be obtained from directory, `/ftp/pub/nolan/PCFgen`, on the ftp anonymous account on thunder.

To test the example PCF Generator:

- 1) Ftp scripts, `inv_pcfgen.csh` and `inv_ascii_gen.csh`, to your working directory
- 2) Verify scripts have execute permission
- 3) Execute LaTIS PGE-unique Preprocessor Simulator script:

inv_ascii_gen.csh

Input file, `inv_ascii_TRMM-PFM-VIRS_PreFlight_00000_1986100105`, will be created.

- 4) Execute, example PCF Generator script with the ASCII input file as a command line argument:
inv_pcfgen.csh inv_ascii_TRMM-PFM-VIRS_PreFlight_00000_1986100105
`PCF_4.5-6.1P1_TRMM-PFM-VIRS_PreFlight_00000_1986100105` will be created.

5.1 Customizing the PCF Generator Example

To customize the PCF Generator example for a different PGE, perform the following steps:

- 1) Examine the PCF template for your PGE and identify all variable directory paths, file names, and runtime parameters.
- 2) Create an ASCII test file which lists all variables identified in Step 1 using a “parameter = value” format (see Section 4.5).
 - A) Modify the LaTIS PGE-unique Preprocessor Simulator script, `inv_ascii_gen.csh`, to include the runtime parameters, directory paths, and file names required for your PGE.
 - B) Set and use environment variables for all variable PCF data.
 - C) Set environment variable `pcf_input` to the name of the new ASCII input file for the PGE Generator.
 - D) Run the modified LaTIS PGE-unique Preprocessor Simulator script to create a new ASCII PGE Generator input file.
- 3) Modify the first section of the PCF Generator script, `inv_pcfgen.csh`, to obtain all of the input data located in your input ASCII file and set each piece of data to an environment variable.
- 4) Modify the second section of the PCF Generator script to create the PCF, using these environment variables set in Step 3.
- 5) Set environment variable `$PGS_PC_INFO_FILE` to the path and name of your PCF template. The PCF Generator script will set `$PCFile` to the path and name of the generated PCF.
- 6) Run the PCF Generator script with the input ASCII filename as a command line argument. A PCF file will be generated.

Appendix A: PCF Generator Example

```
#!/bin/csh
#####
#
# C-shell script - inv_pcfgen.csh
#
# This script generates a PCF file for Inversion PGE 4.5-6.1P1.
# It requires one command line argument which is the ASCII
# input file name
#
# The following environment variable needs to be set before
# using this script:
# $PGS_PC_INFO_FILE - defines the path and name of the PCF Template
#####
set filename = $argv[1]
#####
# This section sets the required environment variables using values
# that are retrieved from the "parameter = value" entries of the ASCII
# file. Using the example ASCII file, parameter $Sampling_Strategy
# gets the value "TRMM-PFM" and parameter $Prod_Strategy gets the value
# "PreFlight"
#####
set PGENAME = `grep "PGENAME" $filename | awk '{print $3}'`
set Sampling_Strategy = `grep "SamplingStrategy" $filename | awk '{print $3}'`
set Prod_Strategy = `grep "ProductionStrategy" $filename | awk '{print $3}'`
set DataYear = `grep "CERDataDateYear" $filename | awk '{print $3}'`
set DataMonth = `grep "CERDataDateMonth" $filename | awk '{print $3}'`
set DataDay = `grep "CERDataDateDay" $filename | awk '{print $3}'`
set HrofMonth = `grep "CERHrOfMonth" $filename | awk '{print $3}'`
set DataHr = `grep "CERHrOfDay" $filename | awk '{print $3}'`
set CC_Code = `grep "ConfigurationCode" $filename | awk '{print $3}'`
set Software_SCCR = `grep "SWsccr" $filename | awk '{print $3}'`
set Ancillary_SCCR = `grep "DATAscrr" $filename | awk '{print $3}'`
set SatInt = `grep "Satellite_Instrument" $filename | awk '{print $3}'`
set AncDataSet = `grep "Ancillary_Data_Set" $filename | awk '{print $3}'`
set SP_Model = `grep "SP_MODEL_NUM" $filename | awk '{print $3}'`
set Surf_Alg = `grep "RUN_SURF_ALG" $filename | awk '{print $3}'`
set PGE_Ver = `grep "PGE_VERSION" $filename | awk '{print $3}'`
set TKVer = `grep "TK_Ver" $filename | awk '{print $3,$4,$5}'`

set InDir1 = `grep "SS4.5_InputDir.1" $filename | awk '{print $3}'`
set InDir2 = `grep "SS4.5_InputDir.2" $filename | awk '{print $3}'`
set InDir3 = `grep "SS4.5_InputDir.3" $filename | awk '{print $3}'`
set InDir4 = `grep "SS4.5_InputDir.4" $filename | awk '{print $3}'`
set InDir5 = `grep "SS4.5_InputDir.5" $filename | awk '{print $3}'`
set OutDir1 = `grep "SS4.5_OutputDir.1" $filename | awk '{print $3}'`
set OutDir2 = `grep "SS4.5_OutputDir.2" $filename | awk '{print $3}'`
set LogsDir = `grep "SS4.5_LogsDir" $filename | awk '{print $3}'`
set Temp_File_Dir = `grep "SS4.5_TempDir" $filename | awk '{print $3}'`
set MCFDir = `grep "SS4.5_MCFDir" $filename | awk '{print $3}'`

set inputfile1 = `grep "SS4.5_Inputfile.1" $filename | awk '{print $3}'`
```



```

echo "#" >> $PCFile
echo "701|SSFB.MCF|$MCFDir||||1" >> $PCFile
echo "702|GQCA.MCF|$MCFDir||||1" >> $PCFile
echo "703|GQCI.MCF|$MCFDir||||1" >> $PCFile
echo "# (DO NOT REMOVE THESE TWO ENTRIES)" >> $PCFile
echo "10252|GetAttr.temp|$Temp_File_Dir||||1" >> $PCFile
echo "10254|MCFWrite.temp|$Temp_File_Dir||||1" >> $PCFile
echo "#" >> $PCFile
echo "# ===== " >> $PCFile
echo "? PRODUCT OUTPUT FILES " >> $PCFile
echo "# ===== " >> $PCFile
echo "! ~/runtime" >> $PCFile
echo "#" >> $PCFile
echo "# -----" >> $PCFile
echo "# Inversion output files" >> $PCFile
echo "# -----" >> $PCFile
echo "#" >> $PCFile
echo "201|$outputfile1|$OutDir1||||1" >> $PCFile
echo "202|$outputfile2|$OutDir2||||1" >> $PCFile
echo "203|$outputfile3|$OutDir2||||1" >> $PCFile
echo "#" >> $PCFile
echo "# ===== " >> $PCFile
echo "? SUPPORT INPUT FILES " >> $PCFile
echo "# ===== " >> $PCFile
echo "! ~/runtime" >> $PCFile
echo "#" >> $PCFile
echo "104|$inputfile4|$InDir4||||1" >> $PCFile
echo "107|$inputfile5_1|$InDir4||||1" >> $PCFile
echo "108|$inputfile5_2|$InDir4||||1" >> $PCFile
echo "109|$inputfile5_3|$InDir4||||1" >> $PCFile
echo "110|$inputfile5_4|$InDir4||||1" >> $PCFile
echo "111|$inputfile5_5|$InDir4||||1" >> $PCFile
echo "#" >> $PCFile
echo "#" >> $PCFile
echo "# ----- " >> $PCFile
echo "# leap seconds (TAI-UTC) file (DO NOT REMOVE THIS ENTRY) " >> $PCFile
echo "# ----- " >> $PCFile
echo "10301|leapsec.dat|~/database/sgi/TD||||1" >> $PCFile
echo "# ----- " >> $PCFile
echo "# polar motion and UTC-UT1 file (DO NOT REMOVE THIS ENTRY) " >> $PCFile
echo "# ----- " >> $PCFile
echo "10401|utcpole.dat|~/database/sgi/CSC||||1" >> $PCFile
echo "# ----- " >> $PCFile
echo "# earth model tags file (DO NOT REMOVE THIS ENTRY)" >> $PCFile
echo "# ----- " >> $PCFile
echo "10402|earthfigure.dat|~/database/sgi/CSC||||1" >> $PCFile
echo "# -----" >> $PCFile
echo "# JPL planetary ephemeris file (binary form) (DO NOT REMOVE THIS ENTRY) " >> $PCFile
echo "# -----" >> $PCFile
echo "10601|de200.eos|~/database/sgi/CBP||||1" >> $PCFile
echo "# -----" >> $PCFile
echo "# spacecraft tag definition file (DO NOT REMOVE THIS ENTRY)" >> $PCFile
echo "# -----" >> $PCFile
echo "10801|sc_tags.dat|~/database/common/EPH||||1" >> $PCFile

```

```

echo "# -----" >> $PCFile
echo "# units conversion definition file (DO NOT REMOVE THIS ENTRY) " >> $PCFile
echo "# -----" >> $PCFile
echo "11001|udunits.dat|~/database/common/CUC|||1" >> $PCFile
echo "# " >> $PCFile
echo "# ===== " >> $PCFile
echo "? SUPPORT OUTPUT FILES " >> $PCFile
echo "# ===== " >> $PCFile
echo "# " >> $PCFile
echo "# -----" >> $PCFile
echo "# These files support the SMF log functionality. Each run will cause " >> $PCFile
echo "# status information to be written to 1 or more of the Log files. To " >> $PCFile
echo "# simulate DAAC operations, remove the 3 Logfiles between test runs. " >> $PCFile
echo "# Remember: all executables within a PGE will contribute status data to " >> $PCFile
echo "# the same batch of log files. (DO NOT REMOVE THESE ENTRIES) " >> $PCFile
echo "# -----" >> $PCFile
echo "10100|$logfile1|$LogsDir|||1" >> $PCFile
echo "10101|$logfile2|$LogsDir|||1" >> $PCFile
echo "10102|$logfile3|$LogsDir|||1" >> $PCFile
echo "10103|TmpStatus|||1" >> $PCFile
echo "10104|TmpReport|||1" >> $PCFile
echo "10105|TmpUser|||1" >> $PCFile
echo "10110|MailFile|||1" >> $PCFile
echo "10111|ShmMem|||1" >> $PCFile
echo "10113|eventLogger.log|||1" >> $PCFile
echo "# " >> $PCFile
echo "# =====" >> $PCFile
echo "? USER DEFINED RUNTIME PARAMETERS" >> $PCFile
echo "# =====" >> $PCFile
echo "# " >> $PCFile
echo "601|SP_MODEL_NUM|$SP_Model" >> $PCFile
echo "602|RUN_SURF_ALG|$Surf_Alg" >> $PCFile
echo "603|PGE_VERSION|$PGE_Ver" >> $PCFile
echo "9991|Subsystem|4.5-6/Inversion" >> $PCFile
echo "9992|CERES ID|SSF" >> $PCFile
echo "9993|Software Version|$PGE_Ver" >> $PCFile
echo "9994|EOSDIS ID|CER12" >> $PCFile
echo "9995|Satellite Instrument|$SatInt" >> $PCFile
echo "9996|Ancillary Data Set|$AncDataSet" >> $PCFile
echo "141|PGENAME|$PGENAME" >> $PCFile
echo "142|SamplingStrategy|$Sampling_Strategy" >> $PCFile
echo "143|ProductionStrategyOUTPUT|$Prod_Strategy" >> $PCFile
echo "144|CERDataDateYear|$DataYear" >> $PCFile
echo "145|CERDataDateMonth|$DataMonth" >> $PCFile
echo "146|CERDataDateDay|$DataDay" >> $PCFile
echo "147|CERHrofMonth|$HrofMonth" >> $PCFile
echo "148|ConfigurationCode|$CC_Code" >> $PCFile
echo "149|SWsccr|$Software_SCCR" >> $PCFile
echo "150|DATAscrr|$Ancillary_SCCR" >> $PCFile
echo "# " >> $PCFile
echo "801|$outputfile1.met|201:1" >> $PCFile
echo "802|$outputfile2.met|202:1" >> $PCFile
echo "803|$outputfile3.met|203:1" >> $PCFile
echo "# " >> $PCFile

```



```
echo "#"  
echo "# ===== "  
echo "? INTERMEDIATE OUTPUT "  
echo "# ===== "  
echo "# "  
echo "# ===== "  
echo "? TEMPORARY I/O "  
echo "# ===== "  
echo "#"  
echo "? END"
```

```
>> $PCFile  
>> $PCFile  
>> $PCFile  
>> $PCFile  
>> $PCFile  
>> $PCFile  
>> $PCFile  
>> $PCFile  
>> $PCFile  
>> $PCFile
```

Appendix B: PCF Example

```
#
# Process Control File: PCF_4.5-6.1P1_TRMM-PFM-VIRS_PreFlight_00000_1986100105
#
# =====
? SYSTEM RUNTIME PARAMETERS
# =====
# -----
# Production Run ID - unique production instance identifier
# -----
1
# -----
# Software ID - unique software configuration identifier
# -----
1
# =====
? PRODUCT INPUT FILES
# =====
# Next non-comment line is the default location for PRODUCT INPUT FILES
# WARNING! DO NOT MODIFY THIS LINE unless you have relocated these
# data set files to the location specified by the new setting.
! ~/runtime
#
# -----
# Inversion input files
# -----
#
#
101|CER_SSFI_TRMM-PFM-VIRS_PreFlight_00000.1986100105|/CERES/clouds/data/out_comp/data/SSF_Int|||1
112|CER_FQCI_TRMM-PFM-VIRS_PreFlight_00000.1986100105|/CERES/clouds/data/out_comp/QA_Reports|||1
102|CER_MOA_TRMM-PFM-VIRS_PreFlight_00000.1986100105|/CERES/sarb/data/out_comp/data/regridmoa/release2|||1
#
701|SSFB.MCF|/CERES/inversion/rcf|||1
702|GQCA.MCF|/CERES/inversion/rcf|||1
703|GQCI.MCF|/CERES/inversion/rcf|||1
# (DO NOT REMOVE THESE TWO ENTRIES)
10252|GetAttr.temp|/CERES/inversion/data/scr|||1
10254|MCFWrite.temp|/CERES/inversion/data/scr|||1
#
# =====
? PRODUCT OUTPUT FILES
# =====
! ~/runtime
#
# -----
# Inversion output files
# -----
#
201|CER_SSFB_TRMM-PFM-VIRS_PreFlight_00000.1986100105|/CERES/inversion/data/out_comp/data|||1
202|CER_GQCA_TRMM-PFM-VIRS_PreFlight_00000.1986100105|/CERES/inversion/data/out_comp/QC|||1
203|CER_GQCI_TRMM-PFM-VIRS_PreFlight_00000.1986100105|/CERES/inversion/data/out_comp/QC|||1
#
```

```

# =====
? SUPPORT INPUT FILES
# =====
! ~/runtime
#
104|ISCTR.093097|/CERES/shared_data|||1
107|NIISW03.093097|/CERES/shared_data|||1
108|NIILWAT.093097|/CERES/shared_data|||1
109|NIILWWN.093097|/CERES/shared_data|||1
110|NIILWSP.093097|/CERES/shared_data|||1
111|NIILWSM.093097|/CERES/shared_data|||1
#
#
# -----
# leap seconds (TAI-UTC) file (DO NOT REMOVE THIS ENTRY)
# -----
10301|leapsec.dat|~/database/sgi/TD|||1
# -----
# polar motion and UTC-UT1 file (DO NOT REMOVE THIS ENTRY)
# -----
10401|utcpole.dat|~/database/sgi/CSC|||1
# -----
# earth model tags file (DO NOT REMOVE THIS ENTRY)
# -----
10402|earthfigure.dat|~/database/sgi/CSC|||1
# -----
# JPL planetary ephemeris file (binary form) (DO NOT REMOVE THIS ENTRY)
# -----
10601|de200.eos|~/database/sgi/CBP|||1
# -----
# spacecraft tag definition file (DO NOT REMOVE THIS ENTRY)
# -----
10801|sc_tags.dat|~/database/common/EPH|||1
# -----
# units conversion definition file (DO NOT REMOVE THIS ENTRY)
# -----
11001|udunits.dat|~/database/common/CUC|||1
#
# =====
? SUPPORT OUTPUT FILES
# =====
#
# -----
# These files support the SMF log functionality. Each run will cause
# status information to be written to 1 or more of the Log files. To
# simulate DAAC operations, remove the 3 Logfiles between test runs.
# Remember: all executables within a PGE will contribute status data to
# the same batch of log files. (DO NOT REMOVE THESE ENTRIES)
# -----
10100|LogStatus_19861001_05|/CERES/inversion/data/runlogs|||1
10101|LogReport_19861001_05|/CERES/inversion/data/runlogs|||1
10102|LogUser_19861001_05|/CERES/inversion/data/runlogs|||1
10103|TmpStatus|||1
10104|TmpReport|||1

```

```

10105|TmpUser||||1
10110|MailFile||||1
10111|ShmMem||||1
10113|eventLogger.log||||1
#
# =====
? USER DEFINED RUNTIME PARAMETERS
# =====
#
601|SP_MODEL_NUM|1
602|RUN_SURF_ALG|1
603|PGE_VERSION|2.0
9991|Subsystem|4.5-6/Inversion
9992|CERES ID|SSF
9993|Software Version|2.0
9994|EOSDIS ID|CER12
9995|Satellite Instrument|NOAA9/CERES
9996|Ancillary Data Set|ERBE_ADMs
141|PGENAME|4.5-6.1P1
142|SamplingStrategy|TRMM-PFM-VIRS
143|ProductionStrategyOUTPUT|PreFlight
144|CERDataDateYear|1986
145|CERDataDateMonth|10
146|CERDataDateDay|01
147|CERHRofMonth|05
148|ConfigurationCode|00000
149|SWsccr|00013
150|DATAsccr|00015
#
801|CER_SSF_TRMM-PFM-VIRS_PreFlight_00000.1986100105.met|201:1
802|CER_GOCA_TRMM-PFM-VIRS_PreFlight_00000.1986100105.met|202:1
803|CER_GOCI_TRMM-PFM-VIRS_PreFlight_00000.1986100105.met|203:1
#
# -----
# These parameters are required to support the PGS_SMF_Send...() tools.
# If the first parameter (TransmitFlag) is disabled, then none of the
# other parameters need to be set. By default, this functionality has been
# disabled. To enable, set TransmitFlag to 1 and supply the other 3
# parameters with local information. (DO NOT REMOVE THESE ENTRIES)
# -----
10109|TransmitFlag; 1=transmit,0=disable|0
10106|RemoteHost|sandcrab
10107|RemotePath|usr/kwan/test/PC/data
10108|EmailAddresses|t.f.fan@larc.nasa.gov
# -----
# The following runtime parameters define various logging options.
# Parameters described as lists should be space (i.e. ) separated.
# The logical IDs 10117, 10118, 10119 listed below are for OPTIONAL
# control of SMF logging. Any of these logical IDs which is unused by a
# PGE may be safely commented out (e.g. if logging is not disabled for
# any status level, then the line beginning 10117 may be commented out).
# -----
10114|Logging Control; 0=disable logging, 1=enable logging|1
10115|Trace Control; 0=no trace, 1=error trace, 2=full trace|0

```



```

10116|Process ID logging; 0=don't log PID, 1=log PID|0
#10117|Disabled status level list (e.g. W S F)|
#10118|Disabled seed list|
#10119|Disabled status code mnemonic list|
#
# -----
# Toolkit version for which this PCF was intended.
# DO NOT EDIT OR REMOVE THIS ENTRY!
# -----
10220|Toolkit version string|SCF B.0 TK5.2
# -----
# ECS DPS trigger for PGE debug runs
# NOTICE TO PGE DEVELOPERS: PGEs which have a debug mode
# need to examine this parameter to evaluate activation rule
# (DO NOT REMOVE THIS ENTRY)
# -----
10911|ECS DEBUG; 0=normal, 1=debug|0
#
# -----
# This entry defines the IP address of the processing host and is used
# by the Toolkit when generating unique Intermediate and Temporary file
# names. The Toolkit no longer relies on the PGS_HOST_PATH environment
# variable to obtain this information. (DO NOT REMOVE THIS ENTRY)
# -----
10099|Local IP Address of 'ether'|155.157.31.87
#
#
10112|Event logging is turned on|1
#
#
# =====
? INTERMEDIATE INPUT
# =====
#
# =====
? INTERMEDIATE OUTPUT
# =====
#
# =====
? TEMPORARY I/O
# =====
#
? END

```

Appendix C: LaTIS PGE-unique Preprocessor Simulator Script

```
#!/bin/csh -x
#####
# inv_ascii_gen.csh
#
# Inversion script which simulates LaTIS PGE-unique Preprocessor
# and creates ASCII input file to test the PGE Generator for PGE
# 4.5-6.1P1
# This will be a LaTIS function and is provided here to
# provide input for PGE Generator testing.
#####
# Set the following 18 environment variables
# Note: the user must enter values for the
#       following environment variables
#       or additional code can be added
#       to enter them interactively
#####
set PGENam = 4.5-6.1P1
set Sampling = TRMM-PFM-VIRS
set Production = PreFlight
set CERYear = 1986
set CERMon = 10
set CERDay = 01
set CERHrMon = 05
set CCode = 00000
set SWsccr = 00013
set DATAsccr = 00015
set SatInst = NOAA9/CERES
set AncData = ERBE_ADMS
set SP_MODEL = 1
set SURF_ALG = 1
set PGE_Ver = 2.0
set TKVer = "SCF B.0 TK5.2"
set CERHr = 05

#####
# Create additional environment variables
#####

set RUN = $CERYear$CERMon$CERDay\_ $CERHr
set INSTANCE = $Sampling\_ $Production\_ $CCode\. $CERYear$CERMon$CERDay$CERHr
set pcf_input = inv_ascii\_ $INSTANCE

#####
# Create the ASCII input file for PCF Generator
#####

echo "#####" >> $pcf_input
echo "# CERES baseline Metadata" >> $pcf_input
echo "#####" >> $pcf_input
echo "PGEName = $PGENam" >> $pcf_input
echo "SamplingStrategy = $Sampling" >> $pcf_input
```

```

echo "ProductionStrategy = $Production" >> $pcf_input
echo "CERDataDateYear = $CERYear" >> $pcf_input
echo "CERDataDateMonth = $CERMon" >> $pcf_input
echo "CERDataDateDay = $CERDay" >> $pcf_input #some PGEs do not have value for CERDataDateDay
echo "CERHrOfMonth = $CERHrMon" >> $pcf_input #some PGEs do not have value for CERHrOfMonth
echo "ConfigurationCode = $CCode" >> $pcf_input
echo "SWsccr = $SWsccr" >> $pcf_input
echo "DATAsccr = $DATAsccr" >> $pcf_input
echo "" >> $pcf_input
echo "#####" >> $pcf_input
echo "# PGE specific runtime parameters" >> $pcf_input
echo "#####" >> $pcf_input
echo "Satellite_Instrument = $SatInst" >> $pcf_input
echo "Ancillary_Data_Set = $AncData" >> $pcf_input
echo "SP_MODEL_NUM = $SP_MODEL" >> $pcf_input
echo "RUN_SURF_ALG = $SURF_ALG" >> $pcf_input
echo "PGE_VERSION = $PGE_Ver" >> $pcf_input
echo "TK_Ver = $TKVer" >> $pcf_input
echo "CERHrOfDay = $CERHr" >> $pcf_input
echo "" >> $pcf_input
echo "#####" >> $pcf_input
echo "# PCF required directories" >> $pcf_input
echo "#####" >> $pcf_input
echo "SS4.5_InputDir.1 = $CERESHOME/clouds/data/out_comp/data/SSF_Int" >> $pcf_input
echo "SS4.5_InputDir.2 = $CERESHOME/clouds/data/out_comp/QA_Reports" >> $pcf_input
echo "SS4.5_InputDir.3 = $CERESHOME/sarb/data/out_comp/data/regridmoa/release2" >> $pcf_input
echo "SS4.5_InputDir.4 = $CERESHOME/shared_data" >> $pcf_input
echo "SS4.5_InputDir.5 = $CERESHOME/inversion/data/ancillary/static" >> $pcf_input
echo "SS4.5_OutputDir.1 = $CERESHOME/inversion/data/out_comp/data" >> $pcf_input
echo "SS4.5_OutputDir.2 = $CERESHOME/inversion/data/out_comp/QC" >> $pcf_input
echo "SS4.5_LogsDir = $CERESHOME/inversion/data/runlogs" >> $pcf_input
echo "SS4.5_MCFDir = $CERESHOME/inversion/rcf" >> $pcf_input
echo "SS4.5_TempDir = $CERESHOME/inversion/data/scr" >> $pcf_input
echo "" >> $pcf_input
echo "#####" >> $pcf_input
echo "# Input file names" >> $pcf_input
echo "#####" >> $pcf_input
echo "SS4.5_Inputfile.1 = CER_SSF1_$INSTANCE" >> $pcf_input
echo "SS4.5_Inputfile.2 = CER_FQCI_$INSTANCE" >> $pcf_input
echo "SS4.5_Inputfile.3 = CER_MOA_$INSTANCE" >> $pcf_input
echo "SS4.5_Inputfile.4 = IISCTR.093097" >> $pcf_input
echo "SS4.5_Inputfile.5_1 = NIISW03.093097" >> $pcf_input
echo "SS4.5_Inputfile.5_2 = NIILWAT.093097" >> $pcf_input
echo "SS4.5_Inputfile.5_3 = NIILWWN.093097" >> $pcf_input
echo "SS4.5_Inputfile.5_4 = NIILWSP.093097" >> $pcf_input
echo "SS4.5_Inputfile.5_5 = NIILWSM.093097" >> $pcf_input
echo "" >> $pcf_input
echo "#####" >> $pcf_input
echo "# Output file names" >> $pcf_input
echo "#####" >> $pcf_input
echo "SS4.5_Outputfile.1 = CER_SSF1_$INSTANCE" >> $pcf_input
echo "SS4.5_Outputfile.2 = CER_GQCA_$INSTANCE" >> $pcf_input
echo "SS4.5_Outputfile.3 = CER_GQCI_$INSTANCE" >> $pcf_input
echo "" >> $pcf_input

```

```
echo "#####"  
echo "# Log file names"  
echo "#####"  
echo ""  
echo "SS4.5_Logsfile.1 = LogStatus_$RUN"  
echo "SS4.5_Logsfile.2 = LogReport_$RUN"  
echo "SS4.5_Logsfile.3 = LogUser_$RUN"
```

```
>> $pcf_input  
>> $pcf_input  
>> $pcf_input  
>> $pcf_input  
>> $pcf_input  
>> $pcf_input  
>> $pcf_input
```