NAME

modGetL1B − extract a band from a MOD02 file and write to an APS−HDF file.

SYNOPSIS

modGetL1B [options] infile outfile prod1 prod2 ...

DESCRIPTION

The program **modGetL1B** is used to extract data from a MODIS MOD02 Level−1B file and put it in an APS−type HDF file. Additionally, the data may be converted from scaled integers to floating point geophysical (radiance or reflectance) values. It might also be Rayleigh corrected. A navigational control points structure is written to the output file.

The products should be given as MODIS product name, an underscore, and a channel name. The product name may be on of:

modis_ch	for MODIS channel values (integers)
modis_L	for MODIS radiance values
modis_rho	for MODIS reflectance values
modis_rho_s	for MODIS reflectance values correct for Rayleigh reflectance

For example, ‘‘modis_L_1’’ represents the channel 1 reflectance data and ‘‘modis_rho_s_8’’ represents the channel 8 Rayleigh corrected reflectance data. The channel names range from 1 to 38. Channels 13 and 14 are special because of two gain settings on the instrument. These channel numbers are ‘‘13hi’’ ‘‘13lo’’ ‘‘14hi’’ or ‘‘14lo’’.

If the input data file is a MOD02QKM or MOD02HKM file, their is no solar angle information needed for the Rayleigh correction. In this case the user must also have the MOD03 geolocation data file (use the -G option). Additionally, the embedded navigation in the MOD021KM file is a subsample of that provided by the MOD03 file.

OPTIONS

-B isp=isp,isl=isl,iep=iep,iel=iel

This option allows the user to subsection out only a part of the file of interest.

-G MOD03

This option selects the geolocation data file

-l Add the arrays ’latitudes’ and ’longitudes’ (locations at each and every point)

EXAMPLE

This example shows what happens when you try to extract a band from a file that does not contain the proper SDS attribute.

```
$ modGetL1B MOD021KM.A2001337.0340.003.2001339033031.hdf test.hdf 1
MOD021KM.A2001337.0340.003.2001339033031.hdf - can’t open SDS (EV_250_RefSB).
```

An example of extract bands 1 and 2 from a MOD02QKM file.

```
$ modGetL1B MOD02QKM.A2001274.1635.003.2001281150149.hdf oct1_250.hdf modis_ch_1 mod
```

An example of extracting Rayleigh-corrected refectance data from bands 1 (red), 4 (green), 3 (blue) from a MOD02HKM file.

```
$ modGetL1B -G MOD03.A2003133.0745.NOAA MOD02HKM.A2003133.0745.NOAA  ˜/rgb.hdf modis
```

NAME

modArea − determine file extents of geographical area

SYNOPSIS

modArea [-M mapFile] mapName inFile

DESCRIPTION

Determine the file extents (start/stop pixel/line) of a MODIS Level−1B file (still in sensor projection, etc.) that covers a map using the geolocation data in the file. It can handle the MOD03 or any of the MOD02 files.

ModArea begins by reading in the map from the mapFile. If the file can not be opened or the named map is not in the file, a diagnostic is printed and the program will exit.

Next, the input file is opened and the navigation information initialized. If unable to open the MODIS file or retrieve the navigation information from it, the program will print a diagnostic and exit.

Once the navigation has been set, **modArea** reads every point to detemine if that point falls within the desired map. From this, the smallest box that will cover the box will be determined. These file extents will be printed to the screen. If none of the latitude/longitude pairs fall in the map, then the message ‘‘No coverage’’ will be printed. If the file extents are the original input file, then the message will be ‘‘Complete coverage.’’

In addition ‘‘No Water Coverage’’ is output if file does not cover any water pixels in the map.

OPTIONS

-a Reduce data file extents by given sensor zenith angle.

-d Debug output.

-h Treat output (sample/lines*2) as HKM for MOD03 file.

-l Don’t output start/stop line locations.

-m percent
Set a minimum coverage that the input data must cover the region of interest. Default value is 0.

-M mapFile
Use the given map file to find mapName. Defaults to $AUTO_DATA/maps.hdf.

-p Don’t output start/stop pixel locations.

-q Treat output (sample/lines*4) as QKM for MOD03 file.

-v Make output verbose.

--help Print out help and exit.

--version
Print out version and exit.

ENVIRONMENTAL VARIABLES

AUTO_DATA
The location of the APS data directory. Used to determine location of the default map file.

EXAMPLES

The first line shows what happens when then environment variable is not set. If not set, user must use -M to define location, unless the default file is in the current directory. The last shows the normal behavior.

```
$ modArea GulfOfMexico MOD021KM.P2003134.1140.NOAA
Map (GulfOfMexico) does not exist in file (maps.hdf).
$ modArea -M ~/aps_v2.6/data/maps.hdf GulfOfMexico MOD021KM.P2003134.1140.NOAA
unable open landmask file, not checking water coverage
No coverage
$ export AUTO_DATA=~/aps_v2.6/data
$ modArea GulfOfMexico MOD021KM.P2003134.1140.NOAA
No coverage
```

The next examples shows examples of regions that cover and do not cover the given regions of interest. It also shows examples of running the code on different inputs files.

```
$ modArea MissBight MOD03.A2002031.1535.003.2002034024442.hdf
No coverage
$ modArea PersianGulf MOD021KM.A2003133.0745.NOAA
747 1354 2 1157
$ modArea PersianGulf MOD03.A2003133.0745.NOAA
742 1354 1 1149
$ modArea PersianGulf MOD02HKM.A2003133.0745.NOAA
1484 2708 1 2298
$ modArea -h PersianGulf MOD03.A2003133.0745.NOAA
1484 2708 1 2298
$ modArea PersianGulf MOD02QKM.A2003133.0745.NOAA
2968 5416 1 4596
$ modArea -q PersianGulf MOD03.A2003133.0745.NOAA
2968 5416 1 4596
```

These examples show a file that is completely over land, and how changing the angle reduces the coverage of the data.

```
$ modArea PersianGulf MOD02HKM.A2003129.0805.NOAA
2350 2708 3634 4042 No Water Coverage
$ modArea ArabianSea MOD03.A2003133.0745.NOAA
678 1354 508 2030
$ modArea -a 60 ArabianSea MOD03.A2003133.0745.NOAA
678 1310 557 2030
$ modArea -a 55 ArabianSea MOD03.A2003133.0745.NOAA
678 1264 594 2030
```

As the angle is reduced, the percent coverage of the region of interest is also reduced. So, as you can see if we set a minimum amount of coverage, we eventually get "No Coverage".

```
$ modArea -m 15 -a 55 MOD03.A2003133.0745.NOAA
```

```
678 1264 594 2030
$ modArea -m 15 -a 50 MOD03.A2003133.0745.NOAA
No coverage
```

If a problem is suspected, then use the -v (verbose) option to output more information.

```
$ modArea -v GulfOfMexico MOD03.A2002031.1535.003.2002034024442.hdf
Using Default MapFile ($AUTO_DATA/maps.hdf)
Initializing Map GulfOfMexico (From File /people/martinol/aps_v2.6/data/maps.hdf)
Reading Navigation Data MOD03.A2002031.1535.003.2002034024442.hdf ... done
Projecting Navigation Data to GulfOfMexico ... done
Scaning Navigation for file limits ... done
Percent Coverage/Miniumum = 0.04747047/0
limits of input file
sinpix = 1
einpix = 25
sinlin = 1
einlin = 11
limits of image map
soupix = 1872
eoupix = 2010
soulin = 1796
eoulin = 1810
size   = 2430 x 1810
1 26 1 12
Normal Completion!
```

NAME

modInfo − query information about a MODIS Level−1B file

SYNOPSIS

modInfo [option] modFile

DESCRIPTION

This program is used to dump information about a MODIS data file. With no options the program will print out a series of parameters. A single parameter can obtained using a selected option. The options are succint as they were designed with shell scripting in mind.

OPTIONS

-day Day of month of input file.

-doy Day of year of input file.

-hour 2-digit time (HH) of input file.

-min 2-digit time (MM) of input file.

-month 3−character month of input file. Months are "jan", "feb", "mar", "apr", "may", "jun", "jul", "aug", "sep", "oct", "nov", "dec"

-platform
Platform of MODIS file (1=Terra, 2=Aqua)

-time 6−digit time (HHMMSS) of input file.

-year Year of input file.

-version
Major component of PGE version of file.

--version
Print out version and exit.

EXAMPLES

```
$ modInfo MOD021KM.A2001337.0340.003.2001339033031.hdf
Filename:         MOD021KM.A2001337.0340.003.2001339033031.hdf
Starting Time:    12/03/2001 03:40, 337
Ending Time:      12/03/2001 03:45, 337
$ modInfo -year MOD021KM.A2001337.0340.003.2001339033031.hdf
2002
$
```

Here is how a Bourne shell script function might use **modInfo** to set the name of the output files from the input file:

```
set_name()
{
    yr=`modInfo -year $1`
    jday=`modInfo -doy $1`
    time=`modInfo -time $1`
    file=MODAM$yr$jday$time.L1A
}
```

NAME

modcol − level-2 product generation for ocean color products

SYNOPSIS

modcol par=parameter.pcf

DESCRIPTION

The program `modcol` is the main processing program for MODIS ocean color products. The main functionality is as follows:

trap scan lines with missing channels
revised (v2) Gordon aerosol algorithm
Gordon O_2 correction for band 7
Gordon tau(865) algorithm
Carder chlorophyll algorithm
Alternate chlor. method when Carder chlor. fails
CZCS-like chlorophyll ratio algorithm
Added parameter file inputs
Added pixel-to-pixel epsilon optimization
Low nLw555, High La865, Cloud/Ice tests
Gordon/Balch detached Coccolith and calcite concentration and coccolith corrected chlorophyll (MOD25)
Hoge inherent optical properties (MOD 31)
Carder ipar, arp (MOD22)
Carder clear water epsilon (MOD39)
Abbott fluorescense line height (MOD20)
Clark CZCS pigm, MODIS pigm, Chlor-a, Diffuse attn, Beam attn total suspended solids (MOD23,21,19)

This program is part of the EOSDIS system and was obtained independently of NASA by contacting U. of Miami directly. However, because the software was written to fit into the MODAPS (MODIS Adaptive Processing System) some features are requirements of the EOSDIS system. These include the Process Control File (PCF), Metadata Configuration Files (MCF), SDP Toolkit, and Product Generation Executives (PGEs).

PROCESS CONTROL FILES

The PCF is essentially an indexed list of all of the input and output data filenames and associated paths that are to be used by the code. It is a routine that defines the files to be staged and/or used by the program. The main code accesses the required files using the appropriate index numbers, which are mapped to variable names within the main code. Further PCF details regarding the format and contents are in **pcf**(5).

The PCFs are automatically generated by EOS during data processing: the files listed within them depend on the location, day and time of data acquisition, and the available ancillary data. For code processing outside of the EOS framework, extra code must be written to generate these PCFs. In the APS, the scripts in **modScripts**(1) automatically generate these files. An example, PCF file for this program is included below.

The PCF is a requirement for the Process Control (PC) tools in the SDP toolkit (described below), enabling input/output functions to occur. The correct PCF is set using the command:

export PGS_PC_INFO_FILE=path/PCFname,

where path is the path where the PCF resides, and PCFname is the name of the appropriate PCF (e.g. PGE09.pcf). All PCFs have the filename extension ".pcf".

In the sections that follow, a description of each entry in the PCF file is given below. The **modcol** program additionally gives names to each logical identifier. When present, these are given with the logical identifier in paranethsis.

MODIS Data Scene Files

These identifiers are part of the `PRODUCT INPUT` section of the PCF file.

l1b (700000)
: Directory path and filename of the input MOD021KM Level−1B data product. Required.

geoloc (600000)
: Directory path and filename of the input MOD03 Geolocation data product. Required.

cloud (422500)
: Directory path and filename of the input MOD35 Cloud mask data product. This parameter is optional.

Output Files

These identifiers are part of the `PRODUCT OUTPUT` section of the PCF file.

outgor (302401)
: Directory path and filename of the output MODOCL2 data product file containing Gordons nLw products. **Required**. The output contains the following products.

nLw_412	Normalized water-leaving radiance at 412 nm
nLw_443	Normalized water-leaving radiance at 443 nm
nLw_490	Normalized water-leaving radiance at 490 nm
nLw_531	Normalized water-leaving radiance at 531 nm
nLw_555	Normalized water-leaving radiance at 555 nm
nLw_667	Normalized water-leaving radiance at 667 nm
nLw_678	Normalized water-leaving radiance at 678 nm
Tau_865	Aerosol optical thickness at 865 nm
Eps_78	Epsilon of aerosol correction at 765 and 865 nm
aer_model1	Aerosol model identification number 1
aer_model2	Aerosol model identification number 2
eps_clr_water	Epsilon of clear water aerosol correction at 531 and 667 nm
cldmsk_flags	Cloud flags
common_flags	Common flags
L2_flags	Level-2 Flags
quality	Quality Values

outdr1 (302402)
: Directory path and filename of the output MODOCL2A data product containing DR1 data products. **Required**. The output contains the following products.

CZCS_pigment	CZCS-like pigment concentration
chlor_MODIS	MODIS chlorophyll concentration
pigment_c1_total	Total case 1 pigment concentration
chlor_fluor_ht	chlorophyll fluorescence line height
chlor_fluor_base	chlorophyll fluorescence baseline
chlor_fluor_effic	chlorophyll fluorescence line efficiency
susp_solids_conc	suspended-solids concentration in ocean
cocco_pigmnt_conc	Pigment concentration in coccolithophore blooms
cocco_conc_detach	Concentration of detached coccolithophores

calcite_conc	Calcite concentration
K_490	Ocean water diffuse attenuation coefficient at 490 nm
phycoeryth_conc	Phycoerythrobilin concentration
phycou_conc	Phycourobilin concentration
cldmsk_flags	Cloud flags
common_flags	Common flags
L2_flags	Level-2 Flags
quality	Quality Values

outdr2 (302403)

Directory path and filename of the output MODOCL2B data product containing DR2 data products. **Required**. The output contains the following products.

chlor_a_2	Chlorophyll a concentration (OC3M)
chlor_a_3	Chlorophyll a concentration (semianalytic)
ipar	Instaneaous photosynthetically available radiation
arp	Instaneaous absorbed radiation by phytoplankton for flourescense
absorp_coef_gelb	Gelbstoff absorption coefficient 400nm
chlor_absorb	Phytoplankton absorption coefficient at 675nm
tot_absorb_412	Total absorption coefficient at 412nm
tot_absorb_443	Total absorption coefficient at 443nm
tot_absorb_488	Total absorption coefficient at 488nm
tot_absorb_531	Total absorption coefficient at 531nm
tot_absorb_551	Total absorption coefficient at 551nm
cldmsk_flags	Cloud flags
common_flags	Common flags
L2_flags	Level-2 Flags
quality	Quality Values

outqc (302405)

Directory path and filename of the output MODOCQC data product containing quality control products. **Required**. The output contains the following products.

U_Wind	U component of winds
V_Wind	U component of winds
Pressure	Pressure
Humidity	Humidity
Ozone	Ozone
Latitude	Latitude
Longitude	Longitude
SolarZenith	Solar Zenith Angle
SolarAzimuth	Solar Azimuth Angle
SatelliteZenith	Satellite Zenith Angle
SatelliteAzimuth	Satellite Azimuth Angle
nLw670	Normalized Water Leaving Radiance at 667 nm
La765	Aerosol Radiance at 765 nm
Ray443	Rayleigh Radiance at 443 nm
Lg865	Glint Radiance at 865 nm
Lf865	Whitecap Radiance at 865 nm
aer_model1	Aerosol model identification number 1
aer_model2	Aerosol model identification number 2

nfile (302412)

Directory path and filename of the output NRL APS data product containing all products listed in the parameters `prods1`, `prods2`, `prod3`, and `prods4`

Products

These keywords control the output of products for the NRL APS data file (nfile). The parameters are `prods1`, `prods2`, `prod3`, and `prods4`, which are all contatenated to produce the full list of products (for internal software reasons these parameters can not exceed 256 characters).

The products which are available are listed in the following tables.

K_532	diffuse attenuation at 532 nm
chl_modis	chlorophyll-a using Clark algorithm
chl_carder	chlorophyll-a using Carder algorithm
adg400_carder	modeled adg400 from Carder algorithm
aph675_carder	modeled aph675 from Carder algorithm
ipar_carder	integrated total sub-surface irradiance (Carder)
arp_carder	integrated total photons absorbed by phytoplankton (Carder)
chl_oc3m	chlorophyll-a using SeaBAM algorithm
peb_555_hoge	PEB absorption
pub_490_hoge	PUB absorption
aph_415_hoge	phytoplankton absorption
adg_415_hoge	CDOM absorption
bb_415_hoge	backscattering
cloud_albedo	cloud albedo
horiz_vis	horizontal diver visibility (Carder,4.8)
vert_vis	veritical diver visibility (Carder,4.0)
horiz_vis_arnone	horizontal diver visibility (Arnone,4.8)
vert_vis_arnone	veritical diver visibility (Arnone,4.0)
l2_flags	Level-2 processing flags

lats	latitude
lons	longitude
solz	solar zenith
sola	solar azimuth
senz	sensor zenith
sena	sensor azimuth
zwind	zonal wind
mwind	meridial wind
precipitation	percipitation
pressure	pressure
ozone	ozone concentration
water_vapor	water vapor
windspeed	wind speed
epsilon	epsilon
tau_865	aerosol optical thickness
glint_865	glint radiance
foam_865	foam radiance

rrs_412	remote sensing reflectance at 412
rrs_443	remote sensing reflectance at 443
rrs_488	remote sensing reflectance at 488
rrs_531	remote sensing reflectance at 531

rrs_551	remote sensing reflectance at 551
rrs_667	remote sensing reflectance at 667
rrs_678	remote sensing reflectance at 678
rrs_750	remote sensing reflectance at 750
rrs_865	remote sensing reflectance at 865
nLw_412	normalized water-leaving radiance at 412
nLw_443	normalized water-leaving radiance at 443
nLw_488	normalized water-leaving radiance at 488
nLw_531	normalized water-leaving radiance at 531
nLw_551	normalized water-leaving radiance at 551
nLw_667	normalized water-leaving radiance at 667
nLw_678	normalized water-leaving radiance at 678
nLw_750	normalized water-leaving radiance at 750
nLw_865	normalized water-leaving radiance at 865

a_412_qaa	absorption at 412 using QAA
a_443_qaa	absorption at 443 using QAA
a_488_qaa	absorption at 488 using QAA
a_531_qaa	absorption at 531 using QAA
a_551_qaa	absorption at 551 using QAA
aph_412_qaa	phytoplankton absorption at 412 using QAA
aph_443_qaa	phytoplankton absorption at 443 using QAA
aph_488_qaa	phytoplankton absorption at 488 using QAA
aph_531_qaa	phytoplankton absorption at 531 using QAA
aph_551_qaa	phytoplankton absorption at 551 using QAA
adg_412_qaa	phytoplankton absorption at 412 using QAA
adg_443_qaa	phytoplankton absorption at 443 using QAA
adg_488_qaa	phytoplankton absorption at 488 using QAA
adg_531_qaa	phytoplankton absorption at 531 using QAA
adg_551_qaa	phytoplankton absorption at 551 using QAA
bb_412_qaa	backscattering at 412 using QAA
bb_443_qaa	backscattering at 443 using QAA
bb_488_qaa	backscattering at 488 using QAA
bb_531_qaa	backscattering at 531 using QAA
bb_551_qaa	backscattering at 551 using QAA
b_412_qaa	scattering at 412 using QAA
b_443_qaa	scattering at 443 using QAA
b_488_qaa	scattering at 488 using QAA
b_531_qaa	scattering at 531 using QAA
b_551_qaa	scattering at 551 using QAA
c_412_qaa	beam attenuation at 412 using QAA
c_443_qaa	beam attenuation at 443 using QAA
c_488_qaa	beam attenuation at 488 using QAA
c_531_qaa	beam attenuation at 531 using QAA
c_551_qaa	beam attenuation at 551 using QAA

a_412_carder	absorption at 412 using Carder
a_443_carder	absorption at 443 using Carder
a_488_carder	absorption at 488 using Carder
a_531_carder	absorption at 531 using Carder
a_551_carder	absorption at 551 using Carder
aph_412_carder	phytoplankton absorption at 412 using Carder
aph_443_carder	phytoplankton absorption at 443 using Carder

aph_488_carder	phytoplankton absorption at 488 using Carder
aph_531_carder	phytoplankton absorption at 531 using Carder
aph_551_carder	phytoplankton absorption at 551 using Carder
adg_412_carder	phytoplankton absorption at 412 using Carder
adg_443_carder	phytoplankton absorption at 443 using Carder
adg_488_carder	phytoplankton absorption at 488 using Carder
adg_531_carder	phytoplankton absorption at 531 using Carder
adg_551_carder	phytoplankton absorption at 551 using Carder
bb_412_carder	backscattering at 412 using Carder
bb_443_carder	backscattering at 443 using Carder
bb_488_carder	backscattering at 488 using Carder
bb_531_carder	backscattering at 531 using Carder
bb_551_carder	backscattering at 551 using Carder
b_412_carder	scattering at 412 using Carder
b_443_carder	scattering at 443 using Carder
b_488_carder	scattering at 488 using Carder
b_531_carder	scattering at 531 using Carder
b_551_carder	scattering at 551 using Carder
c_412_carder	beam attenuation at 412 using Carder
c_443_carder	beam attenuation at 443 using Carder
c_488_carder	beam attenuation at 488 using Carder
c_531_carder	beam attenuation at 531 using Carder
c_551_carder	beam attenuation at 551 using Carder

a_412_arnone	absorption at 412 using Arnone
a_443_arnone	absorption at 443 using Arnone
a_488_arnone	absorption at 488 using Arnone
a_531_arnone	absorption at 531 using Arnone
a_551_arnone	absorption at 551 using Arnone
bb_412_arnone	backscattering at 412 using Arnone
bb_443_arnone	backscattering at 443 using Arnone
bb_488_arnone	backscattering at 488 using Arnone
bb_531_arnone	backscattering at 531 using Arnone
bb_551_arnone	backscattering at 551 using Arnone
b_412_arnone	scattering at 412 using Arnone
b_443_arnone	scattering at 443 using Arnone
b_488_arnone	scattering at 488 using Arnone
b_531_arnone	scattering at 531 using Arnone
b_551_arnone	scattering at 551 using Arnone
c_412_arnone	beam attenuation at 412 using Arnone
c_443_arnone	beam attenuation at 443 using Arnone
c_488_arnone	beam attenuation at 488 using Arnone
c_531_arnone	beam attenuation at 531 using Arnone
c_551_arnone	beam attenuation at 551 using Arnone

Lt_412	Total radiance at 412
Lt_443	Total radiance at 443
Lt_488	Total radiance at 488
Lt_531	Total radiance at 531
Lt_551	Total radiance at 551
Lt_667	Total radiance at 667
Lt_678	Total radiance at 678
Lt_750	Total radiance at 750

Lt_865	Total radiance at 865
La_412	aerosol radiance at 412
La_443	aerosol radiance at 443
La_488	aerosol radiance at 488
La_531	aerosol radiance at 531
La_551	aerosol radiance at 551
La_667	aerosol radiance at 667
La_678	aerosol radiance at 678
La_750	aerosol radiance at 750
La_865	aerosol radiance at 865
Lr_412	Rayleigh radiance at 412
Lr_443	Rayleigh radiance at 443
Lr_488	Rayleigh radiance at 488
Lr_531	Rayleigh radiance at 531
Lr_551	Rayleigh radiance at 551
Lr_667	Rayleigh radiance at 667
Lr_678	Rayleigh radiance at 678
Lr_750	Rayleigh radiance at 750
Lr_865	Rayleigh radiance at 865

rho_t_412	TOA reflectance at 412
rho_t_443	TOA reflectance at 443
rho_t_488	TOA reflectance at 488
rho_t_531	TOA reflectance at 531
rho_t_551	TOA reflectance at 551
rho_t_667	TOA reflectance at 667
rho_t_678	TOA reflectance at 678
rho_t_750	TOA reflectance at 750
rho_t_865	TOA reflectance at 865
rho_a_412	aerosol reflectance at 412
rho_a_443	aerosol reflectance at 443
rho_a_488	aerosol reflectance at 488
rho_a_531	aerosol reflectance at 531
rho_a_551	aerosol reflectance at 551
rho_a_667	aerosol reflectance at 667
rho_a_678	aerosol reflectance at 678
rho_a_750	aerosol reflectance at 750
rho_a_865	aerosol reflectance at 865
rho_r_412	Rayleigh reflectance at 412
rho_r_443	Rayleigh reflectance at 443
rho_r_488	Rayleigh reflectance at 488
rho_r_531	Rayleigh reflectance at 531
rho_r_551	Rayleigh reflectance at 551
rho_r_667	Rayleigh reflectance at 667
rho_r_678	Rayleigh reflectance at 678
rho_r_750	Rayleigh reflectance at 750
rho_r_865	Rayleigh reflectance at 865

Ancillary Input Files

These keywords control the ancilliary input files. If the environment variables have been set (see "ENVIRONMENTAL VARIABLES"), then these keywords have reasonable defaults. These are placed in the `PRODUCT INPUT` section of the PCF file.

met1 (302015)
: Directory path and filename of the climatological product or the near−real−time (NRT) meteorological ancillary data product available for the nearest time preceding the time of ifile product's first scan line. If met1 is the climatological file, then met2 and met3 will not be used, otherwise, see met2 for logic. Default is $MSL1_DATA/CLIMATOLOGY.MET.

met2 (302016)
: Directory path and filename of the NRT meteorological ancillary data product available for the nearest time following the time of ifile product's first scan line's. If met2 is not specified (null) and met1 is a NRT product, then met2 will be set to met1. If met1 <> met2 and the scan line's date and time, fall between the times of met1 and met2, get_ancillary will use met1 and met2 to generate the interpolated meteorological values (if the scan line's date and time fall before those of met1, an error occurs). If met1 = met2 and the scan line's date and time fall before met2, get_ancillary will use only met2 to generate the meteorological values. If met2 <> met3 and the scan line's date and time fall between the times of met2 and met3, get_ancillary will use met2 and met3 to generate the interpolated meteorological values (if the scan line's date and time fall after those of met3, an error occurs). If met2 = met3 and the scan line's date and time fall after met2, get_ancillary will use only met2 to generate the meteorological values.

met3 (302017)
: Directory path and filename of the NRT meteorological ancillary data product for the nearest time following the time of ifile product's last scan line. If met3 is not specified (null) and met1 is a NRT product, then met3 will be set to met2 and the logic specified in met2 will be applied.

ozone1 (302018)
: Directory path and filename of the climatological product or the NRT ozone ancillary data product available for the nearest time preceding the time of ifile product's first scan line. If ozone1 is the climatological file, then ozone2 and ozone3 will not be used, otherwise see ozone2 for logic. (For TOVS data, the center point time is used to represent the time of that product.) Defaults to $MSL12_DATA/CLIMATOLOTY.OZONE.

ozone2 (302019)
: Directory path and filename of the NRT ozone ancillary data product available for the nearest time following the time of ifile product's first scan line's. If ozone2 is not specified (null) and ozone1 is a NRT product, then ozone2 will be set to ozone1. If ozone1 <> ozone2 and the scan line's date and time, fall between the times of ozone1 and ozone2, get_ancillary will use ozone1 and ozone2 to generate the interpolated ozone values (if the scan line's date and time fall before those of ozone1, an error occurs). If ozone1 = ozone2 and the scan line's date and time fall before ozone2, get_ancillary will use only ozone2 to generate the ozone values. If ozone2 <> ozone3 and the scan line's date and time fall between the times of ozone2 and ozone3, get_ancillary will use ozone2 and ozone3 to generate the interpolated ozone values (if the scan line's date and time fall after those of ozone3, an error occurs). If ozone2 = ozone3 and the scan line's date and time fall after ozone2, get_ancillary will use only ozone2 to generate the ozone values. (For TOVS data, the center point time is used to represent the time of that product.)

ozone3 (302020)
: Directory path and filename of the NRT ozone ancillary data product for the nearest time following the time of ifile product's last scan line. If ozone3 is not specified (null) and ozone1 is a NRT product, then ozone3 will be set to ozone2 and the logic specified in ozone2 will be applied. (For TOVS data, the center point time is used to represent the time of that product.)

Additional Ancillary Input Files

These are placed in the `SUPPORT INPUT` section of the PCF file.

rey_wk_lun (900030)
: Directory path and filename of the input Reynolds temperature reference image (version 1).

rey_wk_unf_lun (900031)
: Directory path and filename of the input Reynolds temperature reference image (version 2).

sstimg (302404)
: Directory path and filename of the input MODIS SST file to use as the temperature reference image.

sstqcimg (302406)
: Directory path and filename of the input MODIS SST QC file to use as the temperature reference image.

Input Parameter Tables

These contain a list of parameter files that are required input. They range from values for F0 to parameters for Carder's semi-analytical algorithm. These are placed in the `SUPPORT INPUT` section of the PCF file.

302201 An ascii file that contains a list of parameters. This parameter file is slight different than the PCF file and uses the names rather than the logical identifiers. Due to the number of options, this provides another method of setting parameters than the PCF file. In general, these parameters are considered more static. The default is `$AUTO_DATA/modis/coeff/PGE09/modcol_params22.dat.coeff`.

301930 Default is $AUTO_DATA/modis/coeff/PGE09/coccolith_tables.dat.coeff

302080 Default is $AUTO_DATA/modis/coeff/PGE09/modisdob.cal.coeff

302083 Default is $AUTO_DATA/modis/coeff/PGE09/clark_params4.dat.coeff

302084 Default is $AUTO_DATA/modis/coeff/PGE09/modisf0_1.cal.coeff

302085 Default is $AUTO_DATA/modis/coeff/PGE09/modiswhite.cal.coeff

302086 Default is $AUTO_DATA/modis/coeff/PGE09/modislcw.cal.coeff

302087 Default is $AUTO_DATA/modis/coeff/PGE09/hoge_params2.dat.coeff

302088 Default is $AUTO_DATA/modis/coeff/PGE09/carder_params8.dat.coeff

302089 Default is $AUTO_DATA/modis/coeff/PGE09/ipar_prms.dat.coeff

302090 Default is $AUTO_DATA/modis/coeff/PGE09/global_ndt4.hdf

302091 Default is $AUTO_DATA/modis/coeff/PGE09/modisoob2.cal.coeff

302060 Default is $AUTO_DATA/modis/coeff/PGE09/new_modis_pol_corr5a.hdf

302061 Default is $AUTO_DATA/modis/coeff/PGE09/new_modis_pol_corr5a.hdf

301010 Default is $AUTO_DATA/modis/coeff/PGE09/modis_radcor_v11_28.col.hdf

Input Rayleigh Tables

These contain the data required for Rayleigh calculations. These are placed in the `SUPPORT INPUT` section of the PCF file.

302270 Default is $AUTO_DATA/modis/coeff/PGE09/rayleigh_modis_412_iqu3.hdf

302271 Default is $AUTO_DATA/modis/coeff/PGE09/rayleigh_modis_443_iqu3.hdf

302272 Default is $AUTO_DATA/modis/coeff/PGE09/rayleigh_modis_488_iqu3.hdf

302273 Default is $AUTO_DATA/modis/coeff/PGE09/rayleigh_modis_531_iqu3.hdf

302274 Default is $AUTO_DATA/modis/coeff/PGE09/rayleigh_modis_551_iqu3.hdf

302275 Default is $AUTO_DATA/modis/coeff/PGE09/rayleigh_modis_667_iqu3.hdf

302276 Default is $AUTO_DATA/modis/coeff/PGE09/rayleigh_modis_678_iqu3.hdf

302277 Default is $AUTO_DATA/modis/coeff/PGE09/rayleigh_modis_748_iqu3.hdf

302278 Default is $AUTO_DATA/modis/coeff/PGE09/rayleigh_modis_869_iqu3.hdf

Input Aerosol Model Tables

These contain the data required for the Aerosol calculations. These are placed in the `SUPPORT INPUT` section of the PCF file.

301701 Default is $AUTO_DATA/modis/coeff/PGE09/ocean_h99_v00_s00_d35_3.hdf

301702 Default is $AUTO_DATA/modis/coeff/PGE09/ocean_h90_v00_s00_d35_3.hdf

301703 Default is $AUTO_DATA/modis/coeff/PGE09/ocean_h70_v00_s00_d35_3.hdf

301704 Default is $AUTO_DATA/modis/coeff/PGE09/ocean_h50_v00_s00_d35_3.hdf

301705 Default is $AUTO_DATA/modis/coeff/PGE09/marit_h99_v00_s00_d35_3.hdf

301706 Default is $AUTO_DATA/modis/coeff/PGE09/marit_h90_v00_s00_d35_3.hdf

301707 Default is $AUTO_DATA/modis/coeff/PGE09/coast_h99_v00_s00_d35_3.hdf

301708 Default is $AUTO_DATA/modis/coeff/PGE09/coast_h90_v00_s00_d35_3.hdf

301709 Default is $AUTO_DATA/modis/coeff/PGE09/marit_h70_v00_s00_d35_3.hdf

301710 Default is $AUTO_DATA/modis/coeff/PGE09/marit_h50_v00_s00_d35_3.hdf

301711 Default is $AUTO_DATA/modis/coeff/PGE09/coast_h70_v00_s00_d35_3.hdf

301712 Default is $AUTO_DATA/modis/coeff/PGE09/coast_h50_v00_s00_d35_3.hdf

301713 Default is $AUTO_DATA/modis/coeff/PGE09/tropo_h99_v00_s00_d35_3.hdf

301714 Default is $AUTO_DATA/modis/coeff/PGE09/tropo_h90_v00_s00_d35_3.hdf

301715 Default is $AUTO_DATA/modis/coeff/PGE09/tropo_h70_v00_s00_d35_3.hdf

301716 Default is $AUTO_DATA/modis/coeff/PGE09/tropo_h50_v00_s00_d35_3.hdf

301717 Default is $AUTO_DATA/modis/coeff/PGE09/modisSW_afrid_h06_v00_s00_d35_3.hdf

301718 Default is $AUTO_DATA/modis/coeff/PGE09/modisSW_asiad_h02_v00_s00_d35_3.hdf

Algorithm control options

These keywords modify and/or select the algorithms used to correct the input data or change certain thresholds used for various tests. In general, these are contained in the parameter file (302201) noted above. These are placed in the `USER DEFINED RUNTIME PARAMETERS` section of the PCF file.

solzen1 (302023)

Solar zenith angle in degrees; threshold for setting the l2_flags bit to indicate large solar zenith angles (McClain et al., 1995). Default is 70.0.

satzen1 (302024)

Spacecraft zenith angle in degrees; threshold for setting the l2_flags bit to indicate large satellite zenith angles (McClain et al., 1995). Default is 63.0.

hightau1 (302025)

Aerosol optical thiskness at 865 nm; threshold for setting the <l2_flags> bit to indicate a high <tau_856> value in the level-2 product (McClain et. al. 1995). Default is 9.999.

cldice1 (302026)

Albedo for band 8 in percent; threshold for setting the <l2_flags> bits to indicate clouds or ice (McClain et. al. 1995). Default is 2.5

lowlw1 (302026)

Water-leaving radiance (mW cm-2 um-1 sr-1) at 555 nm; threshold for setting the l2_flags bit to indicate a low nLw_555 value in the Level-2 product (McClain et al., 1995). Default is 2.1

coccolith1

Threshold values (F1,...F8, respectively) used in the following test of Level-2 geophysical parameters for setting the l2_flags bit to indicate coccolithophores (McClain et al., 1995). Note: the first two values are in radiance units (mW cm-2 um-1 sr-1) and the remaining values are ratios. The parameter coccof1 (302028) is the upper threshold nLw_443 and defaults to 11.0. The parameter coccof2 (302029) is the upper threshold nLw_551 and defaults to 8.1. The parameter coccof3 (302030) is the maximum ratio nLw_443/nLw_551 and defaults to 1.0. The parameter coccof4 (302031) is the minimum ratio nLw_443/nLw_551 and defaults to 2.0. The parameter coccof5 (302032) is the maximum ratio nLw_531/nLw_551 and defaults to 1.0. The parameter coccof6 (302033) is the minimum ratio nLw_531/nLw551 and defaults to 1.6. The parameter coccof7 (302034) is the maximum ratio nLw_443/nLw_531 and defaults to 0.95. The parameter coccof8 (302035) is the minimum ratio nLw_443/nLw_531 and defaults to 1.5.

epsilon1

Acceptable minimum and maximum value of <eps_68> in the Level-2 product; used for setting the <l2_flags> bits to indicate failure in the atmospherric correction (McClain et. al. 1995). The parameter epsf1 (302036) is the minimum acceptable value and defaults to 0.67. The parameter epsf2 (302037) is the maximum acceptable value and defaults to 1.5.

turbidw1

Q factor value; used in calculations for setting the l2_flags bit to indicate Case-2 water (McClain et al., 1995). The parameters turbidq (302039) is the Q value for Case-2 water and defaults to 3.42.

mskflg (302022)

A list of algorithm names from those corresponding to each of the l2_flags bits to be interpreted as a mask. Default is LAND1

i_10 (302041)

Use to select type of fitting routine should be used for azimuth angle. If equal to 1 use fourier, otherwise use linear. Defaults to 2 (linear).

boxsiz (302042)

Cloud rejection,"Unit array" size (eg."3"=>a 3x3 box). Defaults to 3.

eps_th (302043)

Value below which the clear-water epsilon routine sets the ‘‘low epsilon’’ flag. Default value is 0.8.

aermodels (302045)

Aerosol models to use. Default value is "1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18"

aermodsea (302049)

Aerosol models for SeaWiFS input file. Default value is "1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18"

subsamplex (302046)

X subsampling factor. Defaults to 1.

subsampley (302047)
Y subsampling factor. Defaults to 1.

use670 (302048)
Select between 765/865 (0) and 670/865 (1) corrections. Defaults to 0.

seabam_thresh (302050)
SeaBAM algorithm threshold. Defaults to 64.0.

nowang (302013)
Disable (1) or enable (0) Gordon/Wang atmospheric correction. Default is 0.

noaero (302014)
Disable (1) or enable (0) Gordon/Wang atmospheric aerosol correction. Default is 0.

sunglint1
fraction of F0(865); used in calculations for setting the l2_flags bit to indicate sun glint (McClain et al., 1995). The parameter `cutoff` (302011) is the sun glint threshold and defaults to 0.005. The parameter `glenab` (302012) is the flag to enable (1) or disable (0) the sun glint test. The default for `glenab` is 1. **The code for this test has been commented out**.

k490_1 (302053)
One of the constants used in the K490 (diffuse attenuation) calculation: K(490) = k490_1 + k490_2 * (nLw(443) / nLw(555)) ** k490_3. Defaults to 0.016

k490_2 (302054)
One of the constants used in the K490 calculation (see k490_1). Defaults to 0.15645

k490_3 (302055)
One of the constants used in the K490 calculation (see k490_1). Defaults to -1.5401

aphiopt This parameter is used only if the Arnone NIR algorthim is used (LW670_THRESH>= 0). A value of 0 uses a constant water absorptance while value of 1 uses an absorption based on the chlorophyll amount. **This parameter is hard-coded to 1**.

wsfactor (322104)
Wind speed multiplier for Cox-Munk glint calculation. Default is 1.0.

reftyp (322097)
Set to (0) to use Reynolds v1, (1) to use MODIS SST (MOD28), or (2) to use Reynolds v2 as the input SST reference image. Defaults to 2.

restrictb (322095)
Number of pixels at beginning of scan line to set qualities to 3. Defaults to 10.

restricte (322096)
Number of pixels at end of scan line to set qualities to 3. Defaults to 10.

usef0var
: Enable (1) or disable (0) correction for Earth-Sun distance. Disabled by default because the MOD02 files have the correction already applied.

sglintth
: Sun glint threshold (Lg865). Defaults to 0.017.

poldet6
: Polarization detection option: 0=standard (default); 1=use detector 6 viewing angles for all detectors (Rayleigh polarization only); 2=same as 1 but rotate azimuth by +90 degrees.

polazim
: Use + (0), - (1) sign for polarization adjustment. Default is 0.

usedust (322084)
: Use a dust aerosol: (1) for african dust, (2) for asian, (3) to try to use both, or and (0) for no dust correction. Defaults to 0.

dustthreshold (322085)
: Threshold to select absorbing aerosol (dust) models. This threshold is against the R(531)/R(667) ratio.

aerothreshold (322086)
: Threshold to disable tropo aerosol models. This threshold is against the R(865).

useavg (322080)
: Enable (1,routine) or disable (0,calibrating) use of averaged pixels for a 3x3 box (see boxsiz).

doavgch1 (302081)
: doavgch1=2 replace detector 1 values with detector 2 values (all channels) doavgch1=1 replace detector 1 just for channel 16 doavgch1=0 use detector 1 values

glintsc
: Glint scaling coefficients for each of the 11 input bands for the removal of sun glint.

lgcalc (322102)
: Glint calculation. Set to 0 to use the Cox-Munk Lg865 calculation. Set to 1 to use the SST brights to calculate Lg865. If set to one the user must provid the four coefficients in `lgsstcs`.

lgsstcs (322103)
: The four coefficients used in the SST brights calculation of Lg865.

usepol (302057)
: Enable (1) or disable (0) polarization correction.

totalrads (302056)
: **This parameter appears to have been removed from modcol**. totalrads=0 (default) outputs normal QC products totalrads=1 outputs La765 instead of pressure and La865 instead of humidity totalrads=2 outputs lots of stuff instead of the normal QC products

docorr (302058)
Enable (1) or disable (0) detector/mirror side corrections radiance corrections.

rho_t_max
Threshold to flag high Lt-Lr-Lg for 676 channel as clouds. Defaults to 5.

rho_wn_max (302082)
Threshold to flag high nLw_676 as clouds. Defaults to 0.01.

usereals (302059)
Output real values (1) or scaled integers (0).

highaoi
The modcol can fix radiance corrections for High and Low AOI sides of scan. This parameter specifies which pixel to start the high corrections.

lacorband (322083)
La band to use to perform atmospheric correction: 1=865, 2=765. Default is 1.

psatazi (322093)
Selects the satellite azimuth calculation. Default is 0.

0	add 180 to input satellite azimuth
1	use input value
2	add 180 and invert sense
3	invert sense of input value
4-7	above but with detector 6 angle for each line of frame
8-11	above but with angle=angle from 11-detector (detector 1 angels are from detector 10 angels)

ndtpackaged (302051)
Threshold to determine Carder NDT packaging state using reference SST field. **Not used in code**.

ndtunpackaged (302052)
Threshold to determine Carder NDT unpackaging state using reference SST field. **Not used in code**.

usenircorr (322098)
Enable (1) or disable (0) the Seigel, et al (2001) based NIR correction. By default this correction is disabled.

nirchlmax (322099)
Set the Seigel NIR chlorophyll threhold. Defaults to 0.3.

nirpctmax (322100)
The precent change threshold for Seigel NIR correction. Defaults to 20.0

niritermx (322102)
The maxium number of iterations for Seigel NIR correction. Defaults to 4.

Input Parameter

These contain a list of parameters that are required input. These are placed in the `SUPPORT INPUT` section of the PCF file.

800510 The platform (AM1M) for Terra, (PM1M) for Aqua.

ALGORITHMS

The following paragraphs will give a synopsis of the various algorithms used by `modcol` where approriate. For those algorithms that are very indepth, the user is to pointed to some references.

Normalized Water Leaving Radiances

The normalized water leaving radiances (nL_w) are computed based on the atmospheric correction algorithm of Howard Gordon. A description for this algorithm can be found at `http://modis.gsfc.nasa.gov/data/atbd/atbd_mod17.pdf`.

Remote Sensing Reflectance

The R_{rs} is defined as the water-leaving radiance divided by the downwelling irradiance just above the sea surface. In `modcol`, it is computed by dividing nL_w by F_0.

Integrated Total Sub-surface Irradiance and Integrated Total Photons Absorbed by Phytoplankton

These PAR products are described in `http://modis.gsfc.nasa.gov/data/atbd/atbd_mod20.pdf`.

Phycouroblin (PUB) absorption coefficient and Phycoerythins (PEB) absorption coefficient

These two products are described in `http://modis.gsfc.nasa.gov/data/atbd/atbd_mod27.pdf`.

OC3M Chlorophyll-a

The OC3M Chlorophyll-a algorithm is found in O'Reilly, J.E., and 24 Coauthors, 2000: SeaWiFS Postlaunch Calibration and Validation Analyses, Part 3. NASA Tech. Memo. 2000-206892, Vol. 11, S.B. Hooker and E.R. Firestone, Eds., NASA Goddard Space Flight Center, 49 pp.

In `modcol`, the *chl* value is initialized to a value of 0.0. This value will be written to the output file for a given pixel in the event that the pixel does not process (due to any set masks). Once the R_{rs} is computed, `modcol` uses the algorithm below if the R_{rs443}, R_{rs488}, and R_{rs555} are all greater than $1 * 10^6$. If the reflectances are less than that threshold, the *chl* is set to -1 for the given pixel.

$$r = \log_{10}\left(\frac{\max(R_{rs443}, R_{rs488})}{R_{rs555}}\right)$$

$$chl = 10^{0.283 - 2.753*r + 1.457*r^2 + 0.659*r_p - 1.403*r^4}$$

The *chl* is bounded between $1 * 10^{-10}$ and $1 * 10^{10}$

Diffuse Attenuation at 490 nm

The K_{490} algorithm is found in O'Reilly, J.E., and 24 Coauthors, 2000: SeaWiFS Postlaunch Calibration and Validation Analyses, Part 3. NASA Tech. Memo. 2000-206892, Vol. 11, S.B. Hooker and E.R. Firestone, Eds., NASA Goddard Space Flight Center, 49 pp.

In `modcol`, the K_{490} value is initialized to a value of -1.0. This value will be written to the output file for a given pixel in the event that the pixel does not process (due to any set masks). Once the nL_w is computed, `modcol` uses the algorithm below if the nL_{w488}, and nL_{w555} are both greater than zero.

$$K_{490} = a_1 + a_2 * \left(\frac{nL_{w488}}{nL_{w555}}\right)^{a_3}$$

Diffuse Attenuation at 532 nm

The K_{532} algorithm uses the spectral nature of K as determined by Austin, Petzold in "Spectral Dependence of the diffuse attenuation coefficient of light in ocean waters", SPIE Vol 489 Ocean Optics V11 (1984) to spectrally shift the K_{490} to K_{532}.

PDL

```
set a bunch of variables
read parameter file to get run time parameters (getcolpar)
setup climatology datasets (set_climatology)
#    load global land mask (loadmask)
open the input file (anlyopi)
if reftyp is 1
    open MOD28/MOD28QC file
    if failed, reset reftyp to 0 (Reynolds)
endif
initialize the navigation data (nav_init)
if useF0 correction
    correct F0 for earth-sun distance
endif
initialize radiance correction
if reftyp is 0
   read in the Reynolds weekly global SST file (readreyw)
endif
decode aerosol model selection data (parsearray)
select models for use
dump run-time information to screen and append to data file, like
    SUNGLINT test, solzen flag angle, etc.
create output file (anlyopo)
load rayleigh data (read_rayleigh_iqu and read_outofband)
load carder NDT reference data (read_ndt_data)
load/init atmsopheric data (cocons)
setup carder chlorophyll algorithm (carder_interp_init)
setup hoge algorithm (hoge_init)
setup clark_init stuff (clark_init)
initialize all flag counts to zero for the various flags checks
initialize output buffers to no data present
write out filler data
read in Lt data from file (anlydri)
        this can be a 3x3 box for example (defined by boxsiz).
do for each output line
    read in Lt data from file (to keep box full, if using a box)
    if user requested an abort, quit
    get navigation data for line
    get atmospheric parameter data for line
    check if line is ascending/descending
    if use F0 correction, correct F0 for earth-sun distance for this line
    compute fresnel reflectance (frensel)
    compute a 3x3 average for bands 15 and 16 (750/865)
    do for each pixel on line
        interpolate various parameters for this pixel (like lat,lon,
                time,angles,ozone,Rayleigh,winds,etc.)
```

```
get sst reference value from Reynolds file
if lgcalc is 1
    set glint coefficient from SST data
else
    set glint coefficient from Cox-Munk
get carder NDT SST
get Lt data for pixel
    if useavg set, then get 3x3 average value
    if doavgch1 is 1,
        replace detector 1 band 16 with those of detector 2
    if doavgch1 is 2,
        replace all bands from detector 1 with those of detector 2
apply white cap correction
apply ozone correction
if atmospheric algorithm will user 6/8, then correct Lt667
    for clear water radiance 0.0090/zbsry1
check to see if count data is negative, then set the appropriate
    flags
check to see if count data is saturated, then set the appropriate
    flags
compute water vapor and wind speed for this pixel
check if navigation if valid for this pixel
quit if any of the flags set so far are reason enough
check for land/shallow water (getmask)
compute percent albedo at 865 (channel 9)
test cloud threshold
test solar zenith angle
test satellite zenith angle
initialize output products
set looping for Arnone NIR
do forever
    test for ancilliary data field problems
    test for bad rayleight
    test for sun glint
    break loop if any tests are used as masks
    estimate the clear water radiance
    if nowang is enabled algorithm
        if too far away from last good calculation
            force aerosol algorithm to do FULL epsilon lookup
        endif
        if channel 765 and 865 are good then
            if Arnone NIR selected
                estimate rho_nir based on Lw667
                    (on first iteration, set rho_nir to zeros, that
                     is no correction)
            compute aeorsol radiance (newatm)
        endif
        if atmospheric algorithm uses 6/8 then
            add back 667 clear water radiance
        endif
    endif
    compute Lw = tLw/t
    compute nLw by normalizing Lw
    test Lw for reasonableness
```

```
                test nLw for reasonableness
                test for clouds uing 678 channel and rho_wn_max threshold
                test Lg for sunglint (SunGlintTh)
                test for clouds using rho_t > rho_t_max
                test La for reasonableness
                test for low Lw555 (lowlw1)
                test for coccolithophore (SeaWiFS test)
                compute coccolithophore (coccolith, uses 443, 551)
                if use new coccolithophore test
                    compute coccolithophore (uses 667, 750, 865)
                compute IOP's (hoge_run)
                compute Pigments Case 1 CZCS
                compute Pigments Case 1 MODIS (all pigments)
                compute Chlor-a Case 1
                compute Total suspended solids
                compute Diffuse attenuation coefficient (443/490)
                compute CZCS-like chlorophyll: Ca+Cp
                test for turbid case 2 (Morel?)
                test for tau865 reasonableness
                compute Carder (packaging set by SST) products
                compute SeaBAM chlorophyll
                compute chl using OC2v4
                compute chl using OC4v4 (not recommended)
                compute chl using OC3M
                compute PAR (carder_ipar)
                compute clear water epsilons (Carder)
                determine if we can break from Arnone NIR
            end do
            store pixel data in output buffers
            determine quality for pixel
            call NRL code for this pixel
        enddo
        clean edges of scan line
        write scan line
    enddo
    clean up
    close all files
```

EXAMPLES

The following two commands show how to run the modcol and its input PCF file.

```
$ modcol par=MODOCL2.A2001244.1800.003.2001353114549.hdf.pcf
$ more MODOCL2.A2001244.1800.003.2001353114549.hdf.pcf
# PGE09 PCF file created for MOD021KM.A2001244.1800.003.2001353114549.hdf
?   SYSTEM RUNTIME PARAMETERS
1
1
?   PRODUCT INPUT FILES
!  /home/aps_v2.6/data/modis
10780|usatile12|/home/aps_v2.6/data/sdptk/AA|||10751|12
10780|usatile11|/home/aps_v2.6/data/sdptk/AA|||10750|11
10780|usatile10|/home/aps_v2.6/data/sdptk/AA|||10749|10
10780|usatile9|/home/aps_v2.6/data/sdptk/AA|||10748|9
10780|usatile8|/home/aps_v2.6/data/sdptk/AA|||10747|8
10780|usatile7|/home/aps_v2.6/data/sdptk/AA|||10746|7
```

```
10780|usatile6|/home/aps_v2.6/data/sdptk/AA|||10745|6
10780|usatile5|/home/aps_v2.6/data/sdptk/AA|||10744|5
10780|usatile4|/home/aps_v2.6/data/sdptk/AA|||10743|4
10780|usatile3|/home/aps_v2.6/data/sdptk/AA|||10742|3
10780|usatile2|/home/aps_v2.6/data/sdptk/AA|||10741|2
10780|usatile1|/home/aps_v2.6/data/sdptk/AA|||10740|1
10951|mowe13a.img|/home/aps_v2.6/data/sdptk/AA||||1
10952|owe13a.img|/home/aps_v2.6/data/sdptk/AA||||1
10953|owe14d.img|/home/aps_v2.6/data/sdptk/AA||||1
10954|owe14dr.img|/home/aps_v2.6/data/sdptk/AA||||1
10955|etop05.dat|/home/aps_v2.6/data/sdptk/AA||||1
10956|fnocazm.img|/home/aps_v2.6/data/sdptk/AA||||1
10957|fnococm.img|/home/aps_v2.6/data/sdptk/AA||||1
10958|fnocpt.img|/home/aps_v2.6/data/sdptk/AA||||1
10959|fnocrdg.img|/home/aps_v2.6/data/sdptk/AA||||1
10960|fnocst.img|/home/aps_v2.6/data/sdptk/AA||||1
10961|fnocurb.img|/home/aps_v2.6/data/sdptk/AA||||1
10962|fnocwat.img|/home/aps_v2.6/data/sdptk/AA||||1
10963|fnocmax.imgs|/home/aps_v2.6/data/sdptk/AA||||1
10964|fnocmin.imgs|/home/aps_v2.6/data/sdptk/AA||||1
10965|fnocmod.imgs|/home/aps_v2.6/data/sdptk/AA||||1
10966|srzarea.img|/home/aps_v2.6/data/sdptk/AA||||1
10967|srzcode.img|/home/aps_v2.6/data/sdptk/AA||||1
10968|srzphas.img|/home/aps_v2.6/data/sdptk/AA||||1
10969|srzslop.img|/home/aps_v2.6/data/sdptk/AA||||1
10970|srzsoil.img|/home/aps_v2.6/data/sdptk/AA||||1
10971|srztext.img|/home/aps_v2.6/data/sdptk/AA||||1
10972|nmcRucPotPres.datrepack|/home/aps_v2.6/data/sdptk/AA||||1
10973|tbase.bin|/home/aps_v2.6/data/sdptk/AA|||10915|1
10974|tbase.br|/home/aps_v2.6/data/sdptk/AA|||10919|4
10974|tbase.bl|/home/aps_v2.6/data/sdptk/AA|||10918|3
10974|tbase.bl|/home/aps_v2.6/data/sdptk/AA|||10918|3
10974|tbase.tr|/home/aps_v2.6/data/sdptk/AA|||10917|2
10974|tbase.tl|/home/aps_v2.6/data/sdptk/AA|||10916|1
10975|geoid.dat|/home/aps_v2.6/data/sdptk/AA||||1
10200|nad27sp|/home/aps_v2.6/data/sdptk/GCT||||1
10201|nad83sp|/home/aps_v2.6/data/sdptk/GCT||||1
10990|eurnasia/|/home/aps_v2.6/data/sdptk/AA||||1
10991|noamer/|/home/aps_v2.6/data/sdptk/AA||||1
10992|soamafr/|/home/aps_v2.6/data/sdptk/AA||||1
10993|sasaus/|/home/aps_v2.6/data/sdptk/AA||||1
10999|PGS_CUC_maths_parameters|/home/aps_v2.6/data/sdptk/CUC||||1
10250|MCF|||||1
10252|GetAttr.4534|/home/aps_v2.6/work||||1
10254|MCFWrite.4534|/home/aps_v2.6/work||||1
311001|MODOCL2.mcf|/home/aps_v2.6/data/modis/PGE09||MODOCL2.mcf||1
311002|MODOCL2A.mcf|/home/aps_v2.6/data/modis/PGE09||MODOCL2A.mcf||1
311003|MODOCL2B.mcf|/home/aps_v2.6/data/modis/PGE09||MODOCL2B.mcf||1
311004|MODOCQC.mcf|/home/aps_v2.6/data/modis/PGE09||MODOCQC.mcf||1
700000|MOD021KM.A2001244.1800.003.2001353114549.hdf|/home/aps_v2.6/work||MOD021KM.A2001244.1800.003.200135311454
600000|MOD03.A2001244.1800.003.2001303133153.hdf|/rs/lvl1/modis/2001/sep||MOD03.A2001244.1800.003.2001303133153.
302015|S200124412_NCEP.MET|/rs/ancil/MET/2001/sep||S200124412_NCEP.MET||1
302016|S200124418_NCEP.MET|/rs/ancil/MET/2001/sep||S200124418_NCEP.MET||1
302017|S200124500_NCEP.MET|/rs/ancil/MET/2001/sep||S200124500_NCEP.MET||1
```

```
302018|S200124312_EPTOMS.OZONE|/rs/ancil/EPTOMS/2001/aug||S200124312_EPTOMS.OZONE||1
302019|S200124412_EPTOMS.OZONE|/rs/ancil/EPTOMS/2001/sep||S200124412_EPTOMS.OZONE||1
302020|S200124512_EPTOMS.OZONE|/rs/ancil/EPTOMS/2001/sep||S200124512_EPTOMS.OZONE||1
?   PRODUCT OUTPUT FILES
!  /home/aps_v2.6/work
302401|MODOCL2.A2001244.1800.003.2001353114549.hdf|/home/aps_v2.6/work||||1
302402|MODOCL2A.A2001244.1800.003.2001353114549.hdf|/home/aps_v2.6/work||||1
302403|MODOCL2B.A2001244.1800.003.2001353114549.hdf|/home/aps_v2.6/work||||1
302405|MODOCQC.A2001244.1800.003.2001353114549.hdf|/home/aps_v2.6/work||||1
302412|NRLCOL.A2001244.1800.003.2001353114549.hdf|/home/aps_v2.6/work||||1
?   SUPPORT INPUT FILES
!  /home/aps_v2.6/data/modis
900030|oi.mean.bias.20010829|/home/aps_v2.6/data/modis/reynolds||oi.mean.bias.20010829||1
302404|MOD28L2.A2001244.1800.003.2001353114549.hdf|/home/aps_v2.6/work||MOD28L2.A2001244.1800.003.2001353114549.h
302406|MOD28QC.A2001244.1800.003.2001353114549.hdf|/home/aps_v2.6/work||MOD28QC.A2001244.1800.003.2001353114549.h
302201|modcol_params22.dat.coeff|/home/aps_v2.6/data/modis/coeff/PGE09||modcol_params22.dat.coeff||1
301930|coccolith_tables.dat.coeff|/home/aps_v2.6/data/modis/coeff/PGE09||coccolith_tables.dat.coeff||1
302080|modisdob.cal.coeff|/home/aps_v2.6/data/modis/coeff/PGE09||modisdob.cal.coeff||1
302083|clark_params4.dat.coeff|/home/aps_v2.6/data/modis/coeff/PGE09||clark_params4.dat||1
302084|modisf0_1.cal.coeff|/home/aps_v2.6/data/modis/coeff/PGE09||modisf0_1.cal.coeff||1
302085|modiswhite.cal.coeff|/home/aps_v2.6/data/modis/coeff/PGE09||modiswhite.cal.coeff||1
302086|modislcw.cal.coeff|/home/aps_v2.6/data/modis/coeff/PGE09||modislcw.cal.coeff||1
302087|hoge_params2.dat.coeff|/home/aps_v2.6/data/modis/coeff/PGE09||hoge_params2.dat||1
302088|carder_params8.dat.coeff|/home/aps_v2.6/data/modis/coeff/PGE09||carder_params8.dat||1
302089|ipar_prms.dat.coeff|/home/aps_v2.6/data/modis/coeff/PGE09||ipar_prms.dat||1
302090|global_ndt4.hdf|/home/aps_v2.6/data/modis/coeff/PGE09||global_ndt4.hdf||1
302091|modisoob2.cal.coeff|/home/aps_v2.6/data/modis/coeff/PGE09||modisoob2.cal.coeff||1
302060|new_modis_pol_corr5a.hdf|/home/aps_v2.6/data/modis/coeff/PGE09||new_modis_pol_corr5a.hdf||1
302061|new_modis_pol_corr5a.hdf|/home/aps_v2.6/data/modis/coeff/PGE09||new_modis_pol_corr5a.hdf||1
301010|modis_radcor_v11_28.col.hdf|/home/aps_v2.6/data/modis/coeff/PGE09||modis_radcor_v11_28.col.hdf||1
302270|rayleigh_modis_412_iqu3.hdf|/home/aps_v2.6/data/modis/coeff/PGE09||rayleigh_modis_412_iqu3.hdf||1
302271|rayleigh_modis_443_iqu3.hdf|/home/aps_v2.6/data/modis/coeff/PGE09||rayleigh_modis_443_iqu3.hdf||1
302272|rayleigh_modis_488_iqu3.hdf|/home/aps_v2.6/data/modis/coeff/PGE09||rayleigh_modis_488_iqu3.hdf||1
302273|rayleigh_modis_531_iqu3.hdf|/home/aps_v2.6/data/modis/coeff/PGE09||rayleigh_modis_531_iqu3.hdf||1
302274|rayleigh_modis_551_iqu3.hdf|/home/aps_v2.6/data/modis/coeff/PGE09||rayleigh_modis_551_iqu3.hdf||1
302275|rayleigh_modis_667_iqu3.hdf|/home/aps_v2.6/data/modis/coeff/PGE09||rayleigh_modis_667_iqu3.hdf||1
302276|rayleigh_modis_678_iqu3.hdf|/home/aps_v2.6/data/modis/coeff/PGE09||rayleigh_modis_678_iqu3.hdf||1
302277|rayleigh_modis_748_iqu3.hdf|/home/aps_v2.6/data/modis/coeff/PGE09||rayleigh_modis_748_iqu3.hdf||1
302278|rayleigh_modis_869_iqu3.hdf|/home/aps_v2.6/data/modis/coeff/PGE09||rayleigh_modis_869_iqu3.hdf||1
301701|ocean_h99_v00_s00_d35_3.hdf|/home/aps_v2.6/data/modis/coeff/PGE09||ocean_h99_v00_s00_d35_3.hdf||1
301702|ocean_h90_v00_s00_d35_3.hdf|/home/aps_v2.6/data/modis/coeff/PGE09||ocean_h90_v00_s00_d35_3.hdf||1
301703|ocean_h70_v00_s00_d35_3.hdf|/home/aps_v2.6/data/modis/coeff/PGE09||ocean_h70_v00_s00_d35_3.hdf||1
301704|ocean_h50_v00_s00_d35_3.hdf|/home/aps_v2.6/data/modis/coeff/PGE09||ocean_h50_v00_s00_d35_3.hdf||1
301705|marit_h99_v00_s00_d35_3.hdf|/home/aps_v2.6/data/modis/coeff/PGE09||marit_h99_v00_s00_d35_3.hdf||1
301706|marit_h90_v00_s00_d35_3.hdf|/home/aps_v2.6/data/modis/coeff/PGE09||marit_h90_v00_s00_d35_3.hdf||1
301707|coast_h99_v00_s00_d35_3.hdf|/home/aps_v2.6/data/modis/coeff/PGE09||coast_h99_v00_s00_d35_3.hdf||1
301708|coast_h90_v00_s00_d35_3.hdf|/home/aps_v2.6/data/modis/coeff/PGE09||coast_h90_v00_s00_d35_3.hdf||1
301709|marit_h70_v00_s00_d35_3.hdf|/home/aps_v2.6/data/modis/coeff/PGE09||marit_h70_v00_s00_d35_3.hdf||1
301710|marit_h50_v00_s00_d35_3.hdf|/home/aps_v2.6/data/modis/coeff/PGE09||marit_h50_v00_s00_d35_3.hdf||1
301711|coast_h70_v00_s00_d35_3.hdf|/home/aps_v2.6/data/modis/coeff/PGE09||coast_h70_v00_s00_d35_3.hdf||1
301712|coast_h50_v00_s00_d35_3.hdf|/home/aps_v2.6/data/modis/coeff/PGE09||coast_h50_v00_s00_d35_3.hdf||1
301713|tropo_h99_v00_s00_d35_3.hdf|/home/aps_v2.6/data/modis/coeff/PGE09||tropo_h99_v00_s00_d35_3.hdf||1
301714|tropo_h90_v00_s00_d35_3.hdf|/home/aps_v2.6/data/modis/coeff/PGE09||tropo_h90_v00_s00_d35_3.hdf||1
301715|tropo_h70_v00_s00_d35_3.hdf|/home/aps_v2.6/data/modis/coeff/PGE09||tropo_h70_v00_s00_d35_3.hdf||1
```

```
301716|tropo_h50_v00_s00_d35_3.hdf|/home/aps_v2.6/data/modis/coeff/PGE09||tropo_h50_v00_s00_d35_3.hdf||1
301717|modisSW_afrid_h06_v00_s00_d35_3.hdf|/home/aps_v2.6/data/modis/coeff/PGE09||modisSW_afrid_h06_v00_s00_d35_3
301718|modisSW_asiad_h02_v00_s00_d35_3.hdf|/home/aps_v2.6/data/modis/coeff/PGE09||modisSW_asiad_h02_v00_s00_d35_3
10900|indexFile|/home/aps_v2.6/data/sdptk/AA||||1
10901|mowe13aSupport|/home/aps_v2.6/data/sdptk/AA||||1
10902|owe13aSupport|/home/aps_v2.6/data/sdptk/AA||||1
10903|owe14Support|/home/aps_v2.6/data/sdptk/AA||||1
10904|etop05Support|/home/aps_v2.6/data/sdptk/AA||||1
10905|fnoc1Support|/home/aps_v2.6/data/sdptk/AA||||1
10906|fnoc2Support|/home/aps_v2.6/data/sdptk/AA||||1
10907|zobler1Support|/home/aps_v2.6/data/sdptk/AA||||1
10908|zobler2Support|/home/aps_v2.6/data/sdptk/AA||||1
10909|nmcRucSupport|/home/aps_v2.6/data/sdptk/AA||||1
10915|tbaseSupport|/home/aps_v2.6/data/sdptk/AA||||1
10916|tbase1Support|/home/aps_v2.6/data/sdptk/AA||||1
10917|tbase2Support|/home/aps_v2.6/data/sdptk/AA||||1
10918|tbase3Support|/home/aps_v2.6/data/sdptk/AA||||1
10919|tbase4Support|/home/aps_v2.6/data/sdptk/AA||||1
10740|usatile1Support|/home/aps_v2.6/data/sdptk/AA||||1
10741|usatile2Support|/home/aps_v2.6/data/sdptk/AA||||1
10742|usatile3Support|/home/aps_v2.6/data/sdptk/AA||||1
10743|usatile4Support|/home/aps_v2.6/data/sdptk/AA||||1
10744|usatile5Support|/home/aps_v2.6/data/sdptk/AA||||1
10745|usatile6Support|/home/aps_v2.6/data/sdptk/AA||||1
10746|usatile7Support|/home/aps_v2.6/data/sdptk/AA||||1
10747|usatile8Support|/home/aps_v2.6/data/sdptk/AA||||1
10748|usatile9Support|/home/aps_v2.6/data/sdptk/AA||||1
10749|usatile10Support|/home/aps_v2.6/data/sdptk/AA||||1
10750|usatile11Support|/home/aps_v2.6/data/sdptk/AA||||1
10751|usatile12Support|/home/aps_v2.6/data/sdptk/AA||||1
10948|geoidSupport|/home/aps_v2.6/data/sdptk/AA||||1
10920|mowe13a.bfm|/home/aps_v2.6/data/sdptk/AA||||1
10921|owe13a.bfm|/home/aps_v2.6/data/sdptk/AA||||1
10922|owe14d.bfm|/home/aps_v2.6/data/sdptk/AA||||1
10923|owe14dr.bfm|/home/aps_v2.6/data/sdptk/AA||||1
10924|etop05.bfm|/home/aps_v2.6/data/sdptk/AA||||1
10925|fnocAzm.bfm|/home/aps_v2.6/data/sdptk/AA||||1
10926|fnocOcm.bfm|/home/aps_v2.6/data/sdptk/AA||||1
10927|fnocPt.bfm|/home/aps_v2.6/data/sdptk/AA||||1
10928|fnocRdg.bfm|/home/aps_v2.6/data/sdptk/AA||||1
10929|fnocSt.bfm|/home/aps_v2.6/data/sdptk/AA||||1
10930|fnocUrb.bfm|/home/aps_v2.6/data/sdptk/AA||||1
10931|fnocWat.bfm|/home/aps_v2.6/data/sdptk/AA||||1
10932|fnocMax.bfm|/home/aps_v2.6/data/sdptk/AA||||1
10933|fnocMin.bfm|/home/aps_v2.6/data/sdptk/AA||||1
10934|fnocMod.bfm|/home/aps_v2.6/data/sdptk/AA||||1
10935|srzArea.bfm|/home/aps_v2.6/data/sdptk/AA||||1
10936|srzCode.bfm|/home/aps_v2.6/data/sdptk/AA||||1
10937|srzPhas.bfm|/home/aps_v2.6/data/sdptk/AA||||1
10938|srzSlop.bfm|/home/aps_v2.6/data/sdptk/AA||||1
10939|srzSoil.bfm|/home/aps_v2.6/data/sdptk/AA||||1
10940|srzText.bfm|/home/aps_v2.6/data/sdptk/AA||||1
10941|nmcRucSigPotPres.bfm|/home/aps_v2.6/data/sdptk/AA||||1
10942|tbase.bfm|/home/aps_v2.6/data/sdptk/AA||||1
```

```
10943|tbase1.bfm|/home/aps_v2.6/data/sdptk/AA||||1
10944|tbase2.bfm|/home/aps_v2.6/data/sdptk/AA||||1
10945|tbase3.bfm|/home/aps_v2.6/data/sdptk/AA||||1
10946|tbase4.bfm|/home/aps_v2.6/data/sdptk/AA||||1
10700|usatile1.bfm|/home/aps_v2.6/data/sdptk/AA||||1
10701|usatile2.bfm|/home/aps_v2.6/data/sdptk/AA||||1
10702|usatile3.bfm|/home/aps_v2.6/data/sdptk/AA||||1
10703|usatile4.bfm|/home/aps_v2.6/data/sdptk/AA||||1
10704|usatile5.bfm|/home/aps_v2.6/data/sdptk/AA||||1
10705|usatile6.bfm|/home/aps_v2.6/data/sdptk/AA||||1
10706|usatile7.bfm|/home/aps_v2.6/data/sdptk/AA||||1
10707|usatile8.bfm|/home/aps_v2.6/data/sdptk/AA||||1
10708|usatile9.bfm|/home/aps_v2.6/data/sdptk/AA||||1
10709|usatile10.bfm|/home/aps_v2.6/data/sdptk/AA||||1
10710|usatile11.bfm|/home/aps_v2.6/data/sdptk/AA||||1
10711|usatile12.bfm|/home/aps_v2.6/data/sdptk/AA||||1
10947|geoid.bfm|/home/aps_v2.6/data/sdptk/AA||||1
10301|leapsec.dat|/home/aps_v2.6/data/sdptk/TD||||1
10401|utcpole.dat|/home/aps_v2.6/data/sdptk/CSC||||1
10402|earthfigure.dat|/home/aps_v2.6/data/sdptk/CSC||||1
10601|de200.eos|/home/aps_v2.6/data/sdptk/sgi/CBP||||1
10801|sc_tags.dat|/home/aps_v2.6/data/sdptk/EPH||||1
10302|udunits.dat|/home/aps_v2.6/data/sdptk/CUC||||1
?   SUPPORT OUTPUT FILES
!  /home/aps_v2.6/work
10100|LogStatus.4534|||||1
10101|LogReport.4534|||||1
10102|LogUser.4534|||||1
10103|TmpStatus|||||1
10104|TmpReport|||||1
10105|TmpUser|||||1
10110|MailFile|||||1
10111|ShmMem.4534|||||1
?   USER DEFINED RUNTIME PARAMETERS
302000|SMFLOG_SCREEN Switch|2
800500|PGEVersion|3.0.0
800550|ProcessingEnvironement|Linux poisonivy 2.4.9-21 #1 Thu Jan 17 14:16:30 EST 2002 i686 unknown
312060|prods1|rrs_412 rrs_443 rrs_488 rrs_531 rrs_551 rrs_667 rrs_678 K_532 chl_modis chl_oc3m chl_carder a_412_a
312061|prods2|a_488_carder a_531_carder a_551_carder aph_443_carder adg_412_carder bb_412_arnone bb_443_arnone bb
312062|prods3|bb_488_carder bb_531_carder bb_551_carder bb_667_carder horiz_vis vert_vis l2_flags
312063|prods4|
10109|TransmitFlag; 1=transmit,0=disable|0
10106|RemoteHost|sandcrab
10107|RemotePath|/usr/kwan/test/PC/data
10108|EmailAddresses|kwan@eos.hitc.com
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|
10220|Toolkit version string|SCF B.0 TK5.2
10120|ADEOS-II s/c reference time|
10121|ADEOS-II ground reference time|
```

```
10122|ADEOS-II s/c clock period|
10123|TRMM UTCF value|
10124|NASA PB5C time Epoch date (ASCII UTC)|
10507|ephemeris data quality flag mask|65536
10508|attitude data quality flag mask|65536
10911|ECS DEBUG; 0=normal, 1=debug|0
10099|Local IP Address of 'ether'|155.157.31.87
?   INTERMEDIATE INPUT
!  ~/runtime
?   INTERMEDIATE OUTPUT
!  ~/runtime
?   TEMPORARY I/O
!  ~/runtime
?   END
```

ACKNOWLEDGMENTS

NAME

modsst – level-2 product generation for sea surface temperature products

SYNOPSIS

modsst par=parameter.pcf

DESCRIPTION

The program `modsst` is the main processing program for MODIS sea surface temperature products. This program is part of the EOSDIS system and was obtained independently of NASA by contacting U. of Miami directly. However, because the software was written to fit into the MODAPS (MODIS Adaptive Processing System) some features are requirements of the EOSDIS system. These include the Process Control File (PCF), Metadata Configuration Files (MCF), SDP Toolkit, and Product Generation Executives (PGEs).

PROCESS CONTROL FILES

The PCF is essentially an indexed list of all of the input and output data filenames and associated paths that are to be used by the code. It is a routine that defines the files to be staged and/or used by the program. The main code accesses the required files using the appropriate index numbers, which are mapped to variable names within the main code. Further PCF details regarding the format and contents are in **pcf**(5).

The PCFs are automatically generated by EOS during data processing: the files listed within them depend on the location, day and time of data acquisition, and the available ancillary data. For code processing outside of the EOS framework, extra code must be written to generate these PCFs. In the APS, the scripts in **modScripts**(1) automatically generate these files. An example, PCF file for this program is included below.

The PCF is a requirement for the Process Control (PC) tools in the SDP toolkit (described below), enabling input/output functions to occur. The correct PCF is set using the command:

export PGS_PC_INFO_FILE=path/PCFname,

where path is the path where the PCF resides, and PCFname is the name of the appropriate PCF (e.g. PGE09.pcf). All PCFs have the filename extension ".pcf".

In the sections that follow, a description of each entry in the PCF file is given below. The **modcol** program additionally gives names to each logical identifier. When present, these are given with the logical identifier in paranethsis.

MODIS Data Scene Files

These identifiers are part of the `PRODUCT INPUT` section of the PCF file.

l1b (700000)
: Directory path and filename of the input MOD021KM Level−1B data product. Required.

geoloc (600000)
: Directory path and filename of the input MOD03 Geolocation data product. Required.

cloud (422500)
: Directory path and filename of the input MOD35 Cloud mask data product. This parameter is optional.

Output Files

out (302401)
Directory path and filename of the output MODOCL2 data product containing Gordons nLw products. **Required**.

outqc (302405)
Directory path and filename of the output MODOCQC data product containing quality control products. **Required**.

nfile (302412)
Directory path and filename of the output NRL APS data product containing all products listed in the parameters `prods1`, `prods2`, `prod3`, and `prods4`

Ancillary Input Files

These keywords control the ancilliary input files. If the environment variables have been set (see ‘‘ENVIRONMENTAL VARIABLES’’), then these keywords have reasonable defaults. These are placed in the `PRODUCT INPUT` section of the PCF file.

met1 (302015)
Directory path and filename of the climatological product or the near−real−time (NRT) meteorological ancillary data product available for the nearest time preceding the time of ifile product’s first scan line. If met1 is the climatological file, then met2 and met3 will not be used, otherwise, see met2 for logic. Default is $MSL1_DATA/CLIMATOLOGY.MET.

met2 (302016)
Directory path and filename of the NRT meteorological ancillary data product available for the nearest time following the time of ifile product’s first scan line’s. If met2 is not specified (null) and met1 is a NRT product, then met2 will be set to met1. If met1 <> met2 and the scan line’s date and time, fall between the times of met1 and met2, get_ancillary will use met1 and met2 to generate the interpolated meteorological values (if the scan line’s date and time fall before those of met1, an error occurs). If met1 = met2 and the scan line’s date and time fall before met2, get_ancillary will use only met2 to generate the meteorological values. If met2 <> met3 and the scan line’s date and time fall between the times of met2 and met3, get_ancillary will use met2 and met3 to generate the interpolated meteorological values (if the scan line’s date and time fall after those of met3, an error occurs). If met2 = met3 and the scan line’s date and time fall after met2, get_ancillary will use only met2 to generate the meteorological values.

met3 (302017)
Directory path and filename of the NRT meteorological ancillary data product for the nearest time following the time of ifile product’s last scan line. If met3 is not specified (null) and met1 is a NRT product, then met3 will be set to met2 and the logic specified in met2 will be applied.

ozone1 (302018)
Directory path and filename of the climatological product or the NRT ozone ancillary data product available for the nearest time preceding the time of ifile product’s first scan line. If ozone1 is the climatological file, then ozone2 and ozone3 will not be used, otherwise see ozone2 for logic. (For TOVS data, the center point time is used to represent the time of that product.) Defaults to $MSL12_DATA/CLIMATOLOTY.OZONE.

ozone2 (302019)
Directory path and filename of the NRT ozone ancillary data product available for the nearest time following the time of ifile product’s first scan line’s. If ozone2 is not specified (null) and ozone1 is a NRT product, then ozone2 will be set to ozone1. If ozone1 <> ozone2 and the scan line’s date and time, fall between the times of ozone1 and ozone2, get_ancillary will use ozone1 and ozone2

to generate the interpolated ozone values (if the scan line's date and time fall before those of ozone1, an error occurs). If ozone1 = ozone2 and the scan line's date and time fall before ozone2, get_ancillary will use only ozone2 to generate the ozone values. If ozone2 <> ozone3 and the scan line's date and time fall between the times of ozone2 and ozone3, get_ancillary will use ozone2 and ozone3 to generate the interpolated ozone values (if the scan line's date and time fall after those of ozone3, an error occurs). If ozone2 = ozone3 and the scan line's date and time fall after ozone2, get_ancillary will use only ozone2 to generate the ozone values. (For TOVS data, the center point time is used to represent the time of that product.)

ozone3 (302020)
Directory path and filename of the NRT ozone ancillary data product for the nearest time following the time of ifile product's last scan line. If ozone3 is not specified (null) and ozone1 is a NRT product, then ozone3 will be set to ozone2 and the logic specified in ozone2 will be applied. (For TOVS data, the center point time is used to represent the time of that product.)

Input Parameter Tables

These contain a list of parameter files that are required input. These are placed in the `SUPPORT INPUT` section of the PCF file.

302201 An ascii file that contains a list of parameters. This parameter file is slight different than the PCF file and uses the names rather than the logical identifiers. Due to the number of options, this provides another method of setting parameters than the PCF file. In general, these parameters are considered more static. The default is `$AUTO_DATA/modis/coeff/PGE10/modsst_params16.dat.coeff`.

301010 Default is $AUTO_DATA/modis/coeff/PGE10/modis_radcor_v4_0.sst.hdf

301097 Default is $AUTO_DATA/modis/coeff/PGE10/modsst_sst_6.coeff

301096 Default is $AUTO_DATA/modis/coeff/PGE10/modsst_sst4_12.coeff

301098 Default is $AUTO_DATA/modis/coeff/PGE10/emissivity.dat.coeff

301099 Default is $AUTO_DATA/modis/coeff/PGE10/modis_rsr2.hdf

Additional Input Parameter Tables

These contain a list of parameter files that are required input due to modsst's reliance upon some routines from modcol for cloud detection. These are placed in the `SUPPORT INPUT` section of the PCF file.

302080 Default is $AUTO_DATA/modis/coeff/PGE09/modisdob.cal.coeff

302083 Default is $AUTO_DATA/modis/coeff/PGE09/clark_params4.dat.coeff

302084 Default is $AUTO_DATA/modis/coeff/PGE09/modisf0_1.cal.coeff

302085 Default is $AUTO_DATA/modis/coeff/PGE09/modiswhite.cal.coeff

302086 Default is $AUTO_DATA/modis/coeff/PGE09/modislcw.cal.coeff

302087 Default is $AUTO_DATA/modis/coeff/PGE09/hoge_params2.dat.coeff

302088 Default is $AUTO_DATA/modis/coeff/PGE09/carder_params8.dat.coeff

302089 Default is $AUTO_DATA/modis/coeff/PGE09/ipar_prms.dat.coeff

302090 Default is $AUTO_DATA/modis/coeff/PGE09/global_ndt4.hdf

302091 Default is $AUTO_DATA/modis/coeff/PGE09/modisoob2.cal.coeff

302060 Default is $AUTO_DATA/modis/coeff/PGE09/new_modis_pol_corr5a.hdf

Options

docorr (301102)
Enable (1) or disable (0) detector/mirror side corrections radiance corrections.

outtlwd (301103)
Enable (1) or disable (0) detector/mirror side corrections radiance corrections.

usereals (301104)
Output real values (1) or scaled integers (0).

degc (321010)
Output temperatures in Celsius.

boxsiz (301011)
Cloud rejection,"Unit array" size (eg."3"=>a 3x3 box). Defaults to 3.

rho_t_max
Threshold to flag high Lt-Lr-Lg for 676 channel as clouds. Defaults to 5.

rho_wn_max (302082)
Threshold to flag high nLw_676 as clouds. Defaults to 0.01.

wsfactor (322104)
Wind speed multiplier for Cox-Munk glint calculation. Default is 1.0.

glintsc Glint scaling coefficients for each of the 11 input bands for the removal of sun glint.

solzen1 (302023)
Solar zenith angle in degrees; threshold for setting the l2_flags bit to indicate large solar zenith angles (McClain et al., 1995). Default is 80.0.

satzen1 (302024)
Spacecraft zenith angle in degrees; threshold for setting the l2_flags bit to indicate large satellite zenith angles (McClain et al., 1995). Default is 55.0.

satzen2 (302024)
A second spacecraft zenith angle in degrees; threshold for setting the l2_flags bit to indicate large satellite zenith angles (McClain et al., 1995). Default is 75.0.

satzen14 (302024)
Spacecraft zenith angle in degrees for sst4 product; threshold for setting the l2_flags bit to indicate large satellite zenith angles (McClain et al., 1995). Default is 55.0.

satzen24 (302024)
A second spacecraft zenith angle in degrees for sst4 product; threshold for setting the l2_flags bit to indicate large satellite zenith angles (McClain et al., 1995). Default is 75.0.

Input Rayleigh Tables

These contain the data required for Rayleigh calculations. These are placed in the `SUPPORT INPUT` section of the PCF file.

302270 Default is $AUTO_DATA/modis/coeff/PGE09/rayleigh_modis_412_iqu3.hdf

302271 Default is $AUTO_DATA/modis/coeff/PGE09/rayleigh_modis_443_iqu3.hdf

302272 Default is $AUTO_DATA/modis/coeff/PGE09/rayleigh_modis_488_iqu3.hdf

302273 Default is $AUTO_DATA/modis/coeff/PGE09/rayleigh_modis_531_iqu3.hdf

302274 Default is $AUTO_DATA/modis/coeff/PGE09/rayleigh_modis_551_iqu3.hdf

302275 Default is $AUTO_DATA/modis/coeff/PGE09/rayleigh_modis_667_iqu3.hdf

302276 Default is $AUTO_DATA/modis/coeff/PGE09/rayleigh_modis_678_iqu3.hdf

302277 Default is $AUTO_DATA/modis/coeff/PGE09/rayleigh_modis_748_iqu3.hdf

302278 Default is $AUTO_DATA/modis/coeff/PGE09/rayleigh_modis_869_iqu3.hdf

ALGORITHMS

The following paragraphs will give a synopsis of the various algorithms used by `modsst` where approriate. For those algorithms that are very indepth, the user to pointed to some references.

Sea Surface Temperature

The derivation of the sea surface temperature (%sst%) product is described in `http://modis.gsfc.nasa.gov/data/atbd/atbd_mod25.pdf`.

PDL

```
CALL GETSSTPAR to get optional parameters
IF refimg not defined and nlsst desired THEN
    PRINT must specify a refimg image with nlsst
    EXIT
ENDIF
CALL SET_CLIMATOLOGY
```

```
CALL MOCEAN_CREATEOUTPUTIMAGE
IF refimg defined THEN
    IF reftyp is SST THEN
        PRINT SST reftyp not supported
        EXIT
    ENDIF
    IF reftyp is REYNOLDS THEN
        CALL READREYW
    ENDIF
ENDIF
CALL MODSSTCOEFFS to get coeff's for tlw calculation
BUILD radiance to temperature lookup tables
CALL GETEMISSIVITY for non-linearity error
CALL GETCORRECTIONS to initialize radiance correction tables
CALL READ_RAYLEIGH_IQU to load Rayleigh data
CALL COCONS to load/initialize atmospheric correction algorithm data structures
LOOP OVER LINES
    CALL MODGETL1BRECORDF to get next record
    CALL AVLOOP to get atmospheric coefficients
    CALL COLOOP to get modcol atmospheric coefficients
    LOOP OVER PIXELS
        GET atmospheric parameters for pixel
        FLAG for high satellite zenith angle
        FLAG for low satellite zenith angle
        IF day THEN
            FLAG for high solar zenith angle
            IF not high solar zenith THEN
                FLAG for glint
            ENDIF
       ENDIF
    ENDLOOP
ENDLOOP
```

EXAMPLES

```
$ modsst par=MOD021KM.A2001337.0340.003.2001339033031.hdf.pcf
```

ACKNOWLEDGMENTS

NAME

modRGB − create a “true color” image from a MOD02 Level-1B file

SYNOPSIS

modRGB [options] infile outfile band1 band2 ...

DESCRIPTION

The program **modRGB** is used to create a true color (rgb) image from channels 1, 4, 3 from a MODIS Level−1B file. If the input file is a 1KM file, the red channel (1) is read from ‘EV_250_Aggr1km_RefSB‘ and the green (4) and blue (3) channels are read from ‘EV_500_Aggr1km_RefSB‘. If the input file is a 500 meter file the red channel is read from ‘EV_250_Aggr500_RefSB‘ and the green and blue channels are read from ‘EV_500_RefSB‘. If the input file is a 250 meter file the red channel is read from ‘EV_250_RefSB‘ and the green and blue channels are read from ‘EV_500_RefSB‘ from the required 500 meter file. If Rayleigh correction is requested, the sensor and solar azimuth and zenith angles will be read from the input file unless -G is used. The output file will contain a 3 banded SDS named ‘true_color‘

OPTIONS

-3 Replace band 3 (BLUE) with band 2.

-4 Replace band 4 (GREEN) with band 2.

-a Compute scaling based on mean and standard devation of the input band data

-B [isp=isp,iep=iep,isl=isl,iel=iel,irp=irp,irl=irl]
Define a subsection of the original image for output.

isp is the starting sample number iep is the ending sample number isl is the starting line number iel is the ending line number irp is the replication factor along the samples dimension(not implemented) irl is the replication factor along the lines dimension(not implemented)

The irp/irl indicates the number of samples/lines to skip or repeat. If set to a negative number each sample is repeated the number of times equal to the absolute value of that number. Thus a positive irp is used to reduce or shrink the image and a negative irp is used to enlarge or magnify the image.

-G file The latitude, longitude, sensor and solar azimuth and zenith angles will be read from "file". These will be read from first input file if this option is not used.

-H pct Compute scaling based on histogram. pct% of pixels are discarded off either edge of the pixels falling within the input range r1,r2(set using -r). The image scaling is then maximized for the remaining data.

-i Write only the ’true_color’ SDS to the output file.

-l Output lat,lon for each pixel.

-r r1,r2 Set range of input data, defaults to .01,1 for all bands. Ranges for the three bands may be set individually by separating each range by a colon.

ex.

-r .01,1;.02,.9;.08,.6

-R r1,r2
Set range of output image. Defaults to 0,250.

-s type
Set output scaling type. A number of scaling options are available including linear scaling and log linear scaling. A complete list can be obtained by using help as the option to -s.

-S
Do not handle saturated pixels.

-x
Do not apply Rayleigh correction.

-v
Verbose

--help
Print out help message and exit.

--version
Print out version and exit.

EXAMPLES

NAME

pcf – Process Control File

DESCRIPTION

This man page describes the format of the Process Control File (PCF) required by **modcol**(1) and **modsst**(1) programs. This documentation is basically lifted from `http://newsroom.gsfc.nasa.gov/sdptoolkit/primer/pc_overview.html`

Introduction

The next highest level of the Toolkit above the SMF tools includes the Process Control (PC) tools. Their purpose is to provide a direct interface between the science software and the rest of the SDPS, including accessing file attributes (data about files), physical filenames (for use by HDF functions), and other functions. These tools are used internally by many Toolkit functions, such as Generic I/O, Ancillary Access, and other tools.

There are two sets of PC tools: the Command tools, which are callable from Unix shell scripts, and the API tools, callable from C and Fortran. Much of the functionality is duplicated between these two groups; many of the Command tools are simple wrappers on the C code of the API tools, with some exceptions. For more information about the Command tools see below. Note: Most of the information in this overview applies to both Command tools and API tools; in particular, both read from the same Process Control File.

The Process Control File (PCF) is central to the PC tools. At the SCF, you construct a PCF using a text editor, one for each PGE. These PCFs are part of the delivery of your software to the DAAC. Your software will access files by logical identifiers (essentially integers, defined by mnemonics). The PCF maps these logical identifiers to physical references (currently physical file names and directories). Each logical identifier corresponds to one or more physical references, or versions. At the SCF, you can use any physical reference you like. In the production environment, the physical reference is supplied by the DAAC. Details are given below.

In this overview section, we walk you through the procedure of constructing your own Process Control File step-by-step, then explain the workings of the pccheck utility, which checks the format of this file. The PCF is read by most of the PC tools (directly or indirectly), and is the current mechanism by which the Toolkit interfaces with the rest of the SDPS. The mechanism may change in the future, but the interface to your code will not.

Constructing your Process Control file

This section explains how to customize a Process Control File for use in your code.

A default Process Control File (PCF) is included in the TK5.1.1 delivery. It contains entries which are either required or optional for use of many Toolkit functions. This file is named `$PGSRUN/PCF.relA.template`. The particular example we use here is from the Pathfinder AVHRR/Land Toolkit Prototype study. The complete example file appears in Appendix B of this document.

It is recommended that you start with the same (customized) copy of the PCF each time you run at the SCF, especially if you are using temporary files in your processing. You don't want previous temporary file references in the PCF, since these files are deleted by the system (unless you are not using PGS_PC_Shell.sh or PGS_PC_TermCom).

The Unix environment variable `$PGS_PC_INFO_FILE` must point to your Process Control file in order for the Toolkit to work at all.

We go through the example file section-by-section. The sections of a Process Control file include:

- SYSTEM RUNTIME PARAMETERS

- PRODUCT INPUT
- PRODUCT OUTPUT
- SUPPORT INPUT
- SUPPORT OUTPUT
- USER-DEFINED RUNTIME PARAMETERS
- INTERMEDIATE INPUT
- INTERMEDIATE OUTPUT
- TEMPORARY IO

All sections of the PCF, except the SYSTEM RUNTIME PARAMETERS and USER-DEFINED RUNTIME PARAMETERS sections, consist of names, locations and other data about physical files. Each of these sections has a default file location, which is at the beginning of the section. The default file location is delimited by a '!' in column one of the PCF. This location points to the default directory in which these files are stored. This may be overridden for individual files, by inserting the fully-qualified physical directory path, as explained below. The PRODUCT INPUT section provides a detailed example of considerations that apply to all sections that involve files. Explanations of other sections provide only differences unique to those sections.

General considerations:

```
#   Process Control File: Pathfinder AVHRR/Land Toolkit
#      Prototype
#
#   Env variable PGS_PC_INFO_FILE must point to this file
```

Comments in a PCF are any lines that begin in the first column with "#".

```
?   SYSTEM RUNTIME PARAMETERS
```

The "?" symbol in the first column defines this line as the subject of the section. These nine subject names must not be changed nor deleted from the PCF.

Blank lines are not allowed.

Pipe character "|" must be used to delimit fields.

The exclamation point "!" must be used to designate the default file location. This must appear before any file entries in each section of the PCF.

The entire length of any line in the PCF may not exceed 1000 characters.

Different sections of the PCF have different numbers of required and optional fields for each entry. In the examples below, each entry is identified as required or optional.

SYSTEM RUNTIME PARAMETERS

```
?   SYSTEM RUNTIME PARAMETERS
# ------------------------------------------------------------
# Production Run ID - unique production run identifier
# ------------------------------------------------------------
1
```

This string identifies the particular run of your algorithm at the SDPS. **This field is required**, and may be up to 200 characters. It cannot be the string "0".

```
# -----------------------------------------------------------
# Software ID - unique software configuration
identifier
# -----------------------------------------------------------
1
```

This string identifies the particular software of which your PGE consists. **This field is required**, and may be up to 200 characters. It cannot be the string "0".

In the production system, both of these fields are written into the PCF by the SDP Planning and Scheduling sub-system. At the SCF, you may use any string you like. Note that the 'Production Run Id' value is used in the naming of Temporary and Intermediate files.

Currently these are the only two fields allowed in this section. DAAC and hardware identification are being consider

PRODUCT INPUT

This section is for primary data files used as input to create standard products. This includes such files as ancillary data, Level 0 data, and standard products output from other PGEs; in general, all of your input files.

```
?   PRODUCT INPUT FILES
# [ next line is for default location ]
! ~/runtime
```

Environment variable PGSHOME/runtime is the default location of the files in this section, unless it is overridden for individual files, as explained below. Note that the tilde character "~" is equated to the environment variable PGSHOME. This is true throughout the entire PCF. This particular default file location $PGSHOME/runtime must not be changed, because of the way the Toolkit Ancillary Data Access input files are handled. Default file locations of all other sections of the PCF may be changed to whatever you like.

```
# -----------------------------------------------------------
# Pathfinder AVHRR/Land input files
# -----------------------------------------------------------
201|87002002709.no9_gac|||||1
401|goldtopolandsea8.bin|||||1
402|gridtoms_1987_sngl_ntwk|||||1
403|ephem8788.dat|||||1
404|timecorr8788.dat|||||1
405|SDSannotations.dat|||||1
406|HDFmetadata.dat|||||1
410|jan021987.proclog|||||1
```

```
201|87002002709.no9_gac|||||1
```

The first entry in this section is used as an example; it is the primary input file for Pathfinder AVHRR/Land processing.

```
201|87002002709.no9_gac|||||1
```

Field 1 is the link between your software and this PCF entry, the **logical identifier**. This identifier should be associated with a mnemonic in your code, at the beginning of the module where you use PGS_IO_Gen_Open to open this file, as shown below. **This field is required**, and must be an integer, of type PGSt_integer (long) in C, INTEGER in Fortran. Science software may use any positive integer for logical identifiers, except integers in the range 10,000-10,999; these numbers are reserved for the Toolkit.

In C, the form of this is

```
#define GAC_FILE 201
```

In Fortran,

```
      PARAMETER (GAC_FILE=201)
```

You then use GAC_FILE as an input parameter to Toolkit function PGS_IO_Gen_Open.

Note that while you can use hard-coded numbers in calling sequences, instead of mnemonics (C) or parameters (Fortran), this will make things difficult for integration and test, and also for maintainance; this practice is strongly discouraged.

```
201|87002002709.no9_gac|||||1
```

Field 2 is the **file reference**, currently the actual physical filename, unqualified (i.e., without directory information). In the future production system, this mechanism may change (for example to a Universal Reference), but this will not affect the science software. **This field is required**, and is a string of up to 256 characters.

```
201|87002002709.no9_gac|||||1
```

Field 3 is the path name, for overriding the default directory. In this example, the Toolkit will look for this file in location `$PGSHOME/runtime/87002002709.no9_gac`. If instead this entry were

```
201|87002002709.no9_gac|/fire2/toma/data||||1
```

then the Toolkit would look for this file in `/fire2/toma/data/87002002709.no9_gac`. This field is optional, and is a string of up to 100 characters.

```
201|87002002709.no9_gac|||||1
```

Field 4, blank here, is reserved for future use.

```
201|87002002709.no9_gac|||||1
```

Field 5, blank here, is the **universal reference**. It may contain any string of up to 150 characters. This value may be returned by calling the function PGS_PC_GetUniversalRef.

```
201|87002002709.no9_gac|||||1
```

Field 6, blank here, is the **attribute location**. It is the name of a file that contains data about the file of Field 2. This file must be in the same directory as the file in Field 2. This field is optional, and is a string of up to 256 characters. For an example of an attribute file, see the descriptions of the PGS_PC_Get*Attr Tools.

```
201|87002002709.no9_gac|||||1
```

Field 7 is the **sequence number**. It is used if there is more than one physical file associated with the logical identifier of Field 1, which is normally only the case for PRODUCT INPUT and PRODUCT OUTPUT files. At the SCF, you must assign this sequence number to each instance of the file in the PCF; at the DAAC, this is done by the production system. The actual value of the sequence number is not relevant to your code; it is an internal number used by the production system. At the SCF, you must list these sequence numbers in the PCF starting with the largest first, then decrementing by one, down to the smallest (1), as shown in the example.

The version number, which is used as an argument to Toolkit functions that access different instances of a file, is not the same as sequence number. The version number is the order which the files are listed in the PCF, from smallest (1) to largest. As an example, if the PCF contains the entries

```
201|87002002710.no9_gac|||||2
201|87002002709.no9_gac|||||1
```

then file *87002002710.no9_gac* is version #1 (sequence #2), and file *87002002709.no9_gac* is version #2 (sequence #1). **No information about file content may be inferred from sequence number**. This number is for internal system use only. Use the version number, i.e., the order of listing of PCF entries, as the input to appropriate Toolkit functions.

(As you may have noticed, it so happens that the the version numbers specified in your code run opposite to the sequence numbers defined in the PCF.)

Field 7 is required for PRODUCT INPUT and PRODUCT OUTPUT files (but is optional for all other sections of the PCF). It must be an integer.

The rest of the entries in the PRODUCT INPUT section of the Pathfinder AVHRR/Land Toolkit Prototype file (Appendix B), in the section labeled "Toolkit product input files", are Toolkit files. These are normally not modified.

PRODUCT OUTPUT

This section is for standard product output files.

```
?   PRODUCT OUTPUT FILES
# [ next line is for default location ]
! ~/runtime
#
# -----------------------------------------------------------
# Pathfinder AVHRR/Land main output file
# -----------------------------------------------------------
301|test11.hdf|||||1
```

This file is defined in C code as

```
#define HDF_FILE      301
```

or in Fortran code as

```
      PARAMETER (HDF_FILE=301)
```

It resides in directory $PGSHOME/runtime. It does not have an attribute file.

This section has the same fields as PRODUCT INPUT.

SUPPORT INPUT

This section is primarily for files that are input to Toolkit functions. Ordinarily, you would not modify any entries in this section. An exception to this is the template files used for ancillary files; see the Ancillary Data Access Tools section.

```
?   SUPPORT INPUT FILES
# [ next line is for default location ]
! ~/runtime
#
# -----------------------------------------------------------
# Pathfinder AVHRR/Land support input files
# -----------------------------------------------------------
```

They reside in directory $PGSHOME/runtime. They have no attribute files.

This section has the same fields as PRODUCT INPUT and PRODUCT OUTPUT, except that Field 7 is not required>

There are no support input files in the Pathfinder AVHRR/Land Toolkit Prototype.

The entries in the SUPPORT INPUT section of the Pathfinder AVHRR/Land Toolkit Prototype file (Appendix B) are Toolkit files, mostly to support the Ancillary Data Access (AA) Tools. You may modify these, as explained in the AA Tools section of this document.

SUPPORT OUTPUT

This section is primarily for files that are output from Toolkit functions

```
?   SUPPORT OUTPUT FILES
# [ next line is for default location ]
! ~/runtime
#
```

This section has the same fields as PRODUCT INPUT and PRODUCT OUTPUT, except that Field 7 is not required.

There are no support output files in the Pathfinder AVHRR/Land Toolkit Prototype.

The Toolkit files in this section support the SMF Log files . You may change the names of the files and directories if you want, but not the logical identifier (Field 1). his section of the PCF is different from the other sections in that it does not contain information about files. Instead, it may be used to obtain other kinds of information from the production environment.

```
?   USER DEFINED RUNTIME PARAMETERS
#
# -----------------------------------------------------------
# Pathfinder AVHRR/Land runtime parameters
# -----------------------------------------------------------
601|requested_size_x|409
602|requested_size_y|128
603|wait_time|3

601|requested_size_x|409

Field 1 is as always the logical identifier. This field is required.
```

```
601|requested_size_x|409
```

Field 2 is the parameter name. It is an optional text string of up to 200 characters The Toolkit ignores this field; its intended use is for identification in this PCF, so you may enter whatever you like here. In the Pathfinder AVHRR/Land Toolkit Protot the name of the variable in the code is used for this purpose.

```
601|requested_size_x|409
```

Field 3 is the parameter value. This is read into your code as a string of up to 200 characters by the Toolkit (PGS_PC_GetConfigData). Your code is responsible for any necessary conversion. e.g. to integer. This field is required.

Toolkit files in this section support the sending of files and email to remote locations. For an explanation of these entries in the PCF, see the Notes section of the Tool Description for PGS_SMF_SendRuntimeData.

INTERMEDIATE INPUT

This section and the next are for intermediate files, or files that will exist for longer than a single PGE, but are not standard products. This section is for intermediate input files.

```
?   INTERMEDIATE INPUT
# [ next line is for default location]
! ~/runtime
#
```

This section has the same fields as PRODUCT INPUT and PRODUCT OUTPUT, except that Field 7 is not required. The unqualified file name (Field 2) is ordinarily the name of a file that was generated as an INTERMEDIATE OUTPUT file by a previous run of this PGE. At the SCF, if you are testing successive runs of a PGE which share intermediate files, you need to make sure that the logical identifier is the same in the PCF that you use for all the runs. If you are accessing the intermediate file from a different PGE than the one that created it, you also need to make sure that the mnemonic definitions in your code reference the same logical identifier. You should also copy over the file name if you use a different PCF.

How intermediate files are handled in the production environment, specifically how long they stay around, has not been determined at this writing.

Use function PGS_IO_Gen_Temp_Open (C) or PGS_IO_Gen_Temp_OpenF (Fortran) to open intermediate files.

INTERMEDIATE OUTPUT

This section is for intermediate output files.

```
?   INTERMEDIATE OUTPUT
# [next line is for default location]
! ~/runtime
#
```

Entries for this section of the PCF are created by the Toolkit; you do not need to e

This section has the same fields as PRODUCT INPUT and PRODUCT OUTPUT, except that Field 7 is not required. The unqualified file name (Field 2) is generated by th

```
How intermediate files are handled in the production environment, specifically
how long they stay around, has not been determined at this writing.

Use function PGS_IO_Gen_Temp_Open (C) or PGS_IO_Gen_Temp_OpenF (Fortran) to open int

There are no intermediate files in the Pathfinder AVHRR/Land Toolkit Prototype.
```

TEMPORARY IO

Temporary files are files that exist only for the duration of a single PGE; the production system deletes these files automatically on PGE termination. (You may use the function PGS_IO_Gen_Temp_Delete to do this at the SCF.) Since a single PGE may consist of several of your executables, this section is part of the PCF to enable these files to be passed among these executables.

Entries for this section of the PCF are created by the Toolkit; you do not need to enter values. The unqualified file name (Field 2) is generated by the Toolkit.

```
?   TEMPORARY IO
# [ next line is for default location ]
! ~/runtime
#
# -----------------------------------------------------------
# Pathfinder AVHRR/Land temporary file
# -----------------------------------------------------------
901|pc1157318822894183312||0|0|0|0

This file is defined in C code as

#define BINARY_OUTPUT           901

or in Fortran code as

      PARAMETER (BINARY_OUTPUT=901)

It resides in directory $PGSHOME/runtime. It has no attribute file.

If you are sharing this temporary file among executables in the same PGE, then you
need to have the same #define or PARAMETER statement in the code for each appropriat

Use function PGS_IO_Gen_Temp_Open (C) or PGS_IO_Gen_Temp_OpenF (Fortran) to open
temporary files. Use function PGS_IO_Gen_Temp_Delete to delete files you no
longer need within a PGE.

All PCFs must end with the line

? END

Any information after this line is ignored.
```

Checking your Process Control File

Now that you have created your PCF, you can use it from your software through use of the Toolkit. However, you might want to check it to see if you have entered everything correctly. You can do this by using the pccheck utility, a Unix executable included with the Toolkit. This program is compiled at the time of Toolkit installation, and is located in directory $PGSBIN. You execute shell script pccheck.sh, which calls executable pctcheck; its source code is $PGSSRC/PC/PGS_PC_Check.c . To run it on your file mypcfile, on the Unix command line type

```
$PGSBIN/pccheck.sh -i mypcfile
```

If there are any errors in your file, you will see messages of the form

```
Error - problem with version number in Standard input file
Line number:  23
Line:  401|goldtopolandsea8.bin|||||
```

In this example the version number was omitted from the STANDARD INPUT file entry.

At the end, you will see a summary of the form

```
Check of mypcfile completed
Errors found:  7
Warnings found:  0
```

For this utility, a pccheck error is defined as a PCF entry that will cause a Toolkit PC function to return an error message. A pccheck warning is defined as an incorrect entry that will not cause the Toolkit trouble, but may cause the PGE to operate incorrectly. For example, a blank character in the file name field does not bother a Toolkit PC function, since it simply returns the string as is; pccheck will not return an error. But a blank character will certainly cause a Unix error, when the file open is attempted by a Toolkit function; pccheck will return a warning to this effect. Output is returned to stdout (usually the screen).

This is a simple explanation of how the pccheck utility works. For details, including a list of error messages, and information about other command line options, see "Validating Process Control Files", sec. C.2 of Appendix C, in the Toolkit Users Guide.

Sample Process Control File (PCF)

This is an example Pathfinder AVHRR/Land process control file

```
# Process Control File: Pathfinder AVHRR/Land Toolkit Prototype
#
# Environment variable PGS_PC_INFO_FILE must point to this file
#
?   SYSTEM RUNTIME PARAMETERS
# ----------------------------------------------------------------
# Production Run ID - unique production instance identifier
# ----------------------------------------------------------------
1
# ----------------------------------------------------------------
# Software ID - unique software configuration identifier
# ----------------------------------------------------------------
1
#
?   PRODUCT INPUT FILES
# [ next line is for default location ]
! ~/runtime
#
# ----------------------------------------------------------------
# Pathfinder AVHRR/Land input files
# ----------------------------------------------------------------
201|87002002709.no9_gac|||||1
401|goldtopolandsea8.bin|||||1
```

```
402|gridtoms_1987_sngl_ntwk|||||1
403|ephem8788.dat|||||1
404|timecorr8788.dat|||||1
405|SDSannotations.dat|||||1
406|HDFmetadata.dat|||||1
410|jan021987.proclog|||||1
#
#
# ---------------------------------------------------------------
# Toolkit product input files
# ---------------------------------------------------------------
#
# ---------------------------------------------------------------
# These are actual ancillary data set files - supplied by ECS or
# the user.
# The following are supplied for purposes of tests and as a
# useful set of ancillary data.
# The files will be located in $PGSHOME/runtime.
#
# WARNING! DO NOT MODIFY DEFAULT FILE LOCATION FOR THIS SECTION
# unless you have relocated these data set files to the location
# specified by the location's new setting.
# ---------------------------------------------------------------

10780|usatile12||||10751|12
10780|usatile11||||10750|11
10780|usatile10||||10749|10
10780|usatile9||||10748|9
10780|usatile8||||10747|8
10780|usatile7||||10746|7
10780|usatile6||||10745|6
10780|usatile5||||10744|5
10780|usatile4||||10743|4
10780|usatile3||||10742|3
10780|usatile2||||10741|2
10780|usatile1||||10740|1
10951|mowe13a.img|||||1
10952|owe13a.img|||||1
10953|owe14d.img|||||1
10954|owe14dr.img|||||1
10955|etop05.dat|||||1
10956|fnocazm.img|||||1
10957|fnococm.img|||||1
10958|fnocpt.img|||||1
10959|fnocrdg.img|||||1
10960|fnocst.img|||||1
10961|fnocurb.img|||||1
10962|fnocwat.img|||||1
10963|fnocmax.imgs|||||1
10964|fnocmin.imgs|||||1
10965|fnocmod.imgs|||||1
10966|srzarea.img|||||1
10967|srzcode.img|||||1
10968|srzphas.img|||||1
```

```
10969|srzslop.img|||||1
10970|srzsoil.img|||||1
10971|srztext.img|||||1
10972|nmcRucPotPres.datrepack|||||1
10973|tbase.bin||||10915|1
10974|tbase.br||||10919|4
10974|tbase.bl||||10918|3
10974|tbase.tr||||10917|2
10974|tbase.tl||||10916|1
# -----------------------------------------------------------------------
# Constant & Unit Conversion file
# IMPORTANT NOTE: THIS FILE WILL BE SUPPLIED AFTER TK4 DELIVERY!
# -----------------------------------------------------------------------
10999|PGS_CUC_maths_parameters|||||1
#
#
# -----------------------------------------------------------------------
# The following are for the PGS_GCT tool only.
# The IDs are #defined in the PGS_GCT.h file
# -----------------------------------------------------------------------
10200|nad27sp|~/runtime||||1
10201|nad83sp|~/runtime||||1
# ------------------------------------------------------------------
# The following are for the PGS_AA_DCW tool only.
# The IDs are #defined in the PGS_AA_DCW.h file
# ------------------------------------------------------------------
10990|eurnasia/|||||1
10991|noamer/|||||1
10992|soamafr/|||||1
10993|sasaus/|||||1
#
# ------------------------------------------------------------------
# End Toolkit product input files
# ------------------------------------------------------------------
#
?   PRODUCT OUTPUT FILES
# [ next line is for default location ]
! ~/runtime
#
# ------------------------------------------------------------------
# Pathfinder AVHRR/Land main output file
# ------------------------------------------------------------------
301|test11.hdf|||||1
#
?   SUPPORT INPUT FILES
# [ next line is for default location ]
! ~/supportinput
#
# ------------------------------------------------------------------
# Toolkit support input files
# ------------------------------------------------------------------
#
# ------------------------------------------------------------------
# This ID is #defined in PGS_AA_Tools.h . This file contains
```

```
# the IDs for all support and format files shown.
# ----------------------------------------------------------------
10900|indexFile|~/runtime||||1
#
# ----------------------------------------------------------------
# These are support files for the data set files - to be created
#  by user (not necessarily a one-to-one relationship)
# The IDs must correspond to the logical IDs in the index file
# ----------------------------------------------------------------
10901|mowe13aSupport|~/runtime||||1
10902|owe13aSupport|~/runtime||||1
10903|owe14Support|~/runtime||||1
10904|etop05Support|~/runtime||||1
10905|fnoc1Support|~/runtime||||1
10906|fnoc2Support|~/runtime||||1
10907|zobler1Support|~/runtime||||1
10908|zobler2Support|~/runtime||||1
10909|nmcRucSupport|~/runtime||||1
10915|tbaseSupport|~/runtime||||1
10916|tbase1Support|~/runtime||||1
10917|tbase2Support|~/runtime||||1
10918|tbase3Support|~/runtime||||1
10919|tbase4Support|~/runtime||||1
10740|usatile1Support|~/runtime||||1
10741|usatile2Support|~/runtime||||1
10742|usatile3Support|~/runtime||||1
10743|usatile4Support|~/runtime||||1
10744|usatile5Support|~/runtime||||1
10745|usatile6Support|~/runtime||||1
10746|usatile7Support|~/runtime||||1
10747|usatile8Support|~/runtime||||1
10748|usatile9Support|~/runtime||||1
10749|usatile10Support|~/runtime||||1
10750|usatile11Support|~/runtime||||1
10751|usatile12Support|~/runtime||||1
#
# ----------------------------------------------------------------
# The following are format files for each data set file
# (not necessarily a one-to-one relationship)
# The IDs must correspond to the logical IDs in the index file
# ----------------------------------------------------------------

10920|mowe13a.bfm|~/runtime||||1
10921|owe13a.bfm|~/runtime||||1
10922|owe14d.bfm|~/runtime||||1
10923|owe14dr.bfm|~/runtime||||1
10924|etop05.bfm|~/runtime||||1
10925|fnocAzm.bfm|~/runtime||||1
10926|fnocOcm.bfm|~/runtime||||1
10927|fnocPt.bfm|~/runtime||||1
10928|fnocRdg.bfm|~/runtime||||1
10929|fnocSt.bfm|~/runtime||||1
10930|fnocUrb.bfm|~/runtime||||1
10931|fnocWat.bfm|~/runtime||||1
```

```
10932|fnocMax.bfm|~/runtime||||1
10933|fnocMin.bfm|~/runtime||||1
10934|fnocMod.bfm|~/runtime||||1
10935|srzArea.bfm|~/runtime||||1
10936|srzCode.bfm|~/runtime||||1
10937|srzPhas.bfm|~/runtime||||1
10938|srzSlop.bfm|~/runtime||||1
10939|srzSoil.bfm|~/runtime||||1
10940|srzText.bfm|~/runtime||||1
10941|nmcRucSigPotPres.bfm|~/runtime||||1
10942|tbase.bfm|~/runtime||||1
10943|tbase1.bfm|~/runtime||||1
10944|tbase2.bfm|~/runtime||||1
10945|tbase3.bfm|~/runtime||||1
10946|tbase4.bfm|~/runtime||||1
10700|usatile1.bfm|~/runtime||||1
10701|usatile2.bfm|~/runtime||||1
10702|usatile3.bfm|~/runtime||||1
10703|usatile4.bfm|~/runtime||||1
10704|usatile5.bfm|~/runtime||||1
10705|usatile6.bfm|~/runtime||||1
10706|usatile7.bfm|~/runtime||||1
10707|usatile8.bfm|~/runtime||||1
10708|usatile9.bfm|~/runtime||||1
10709|usatile10.bfm|~/runtime||||1
10710|usatile11.bfm|~/runtime||||1
10711|usatile12.bfm|~/runtime||||1
#
# ----------------------------------------------------------------
# leap seconds (TAI-UTC) file
# ----------------------------------------------------------------
10301|leapsec.dat|~/lib/database/TD||||1
#
# ----------------------------------------------------------------
# polar motion and UT1-UTC file
# ----------------------------------------------------------------
10401|utcpole.dat|~/lib/database/CSC||||1
#
# ----------------------------------------------------------------
# earth model tags file
# ----------------------------------------------------------------
10402|earthfigure.dat|~/lib/database/CSC||||1
#
# ----------------------------------------------------------------
# directory where spacecraft ephemeris files are located
# NOTE: This line is used to specify a directory only!
#       The "file" field should not be altered.
# ----------------------------------------------------------------
10501|.|~/lib/database/EPH||||1
#
# ----------------------------------------------------------------
# JPL planetary ephemeris file (binary form)
# ----------------------------------------------------------------
10601|de200.eos|~/lib/database/CBP||||1
```

```
# ---------------------------------------------------------------
# End Toolkit support input files
# ---------------------------------------------------------------
#
#
?   SUPPORT OUTPUT FILES
# [ next line is for default location ]
! ~/supportoutput
#
#
# ---------------------------------------------------------------
# Toolkit 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.
# ---------------------------------------------------------------
10100|LogStatus|~/runtime||||1
10101|LogReport|~/runtime||||1
10102|LogUser|~/runtime||||1
10103|TmpStatus|~/runtime||||1
10104|TmpReport|~/runtime||||1
10105|TmpUser|~/runtime||||1
10110|MailFile|~/runtime||||1
#
# ------------------------------------------------------------------------
# ASCII file which stores pointers to runtime SMF files in lieu of
# loading them to shared memory.
# ------------------------------------------------------------------------
10111|ShmMem|~/runtime||||1
#
# ---------------------------------------------------------------
# End Toolkit support output files
# ---------------------------------------------------------------
#
#
?   USER DEFINED RUNTIME PARAMETERS
#
# ---------------------------------------------------------------
# Pathfinder AVHRR/Land runtime parameters
# ---------------------------------------------------------------
601|requested_size_x|409
602|requested_size_y|128
603|wait_time|3
#
# ---------------------------------------------------------------
# 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.
# ----------------------------------------------------------------
10109|TransmitFlag; 1=transmit,0=disable|1
10106|RemoteHost|fire@eos.hitc.com
10107|RemotePath|/fire2/toma/inbox
10108|EmailAddresses|toma@eos.hitc.com
#
# -------------------------------------------------------------------------
# Default location for processing host IP address.
# This is overridden by the environment variable PGS_HOST_PATH.
# -------------------------------------------------------------------------
10099|Local IP Address of 'ether'|155.157.31.87
#
#
?   INTERMEDIATE INPUT
# [ next line is for default location ]
! ~/runtime
#
#
?   INTERMEDIATE OUTPUT
# [ next line is for default location ]
! ~/runtime
#
#
?   TEMPORARY IO
# [ next line is for default location ]
! ~/runtime
#
# ----------------------------------------------------------------
# Pathfinder AVHRR/Land temporary file
# ----------------------------------------------------------------
901|test10.bin|||||
#
#
?   END
```