**Archive of Processing Code and Support Files for NASA/RSS SMAP Salinity Version 5.0 Validated Release**

**07/25/2022**

**Thomas Meissner**

**Remote Sensing Systems (RSS)**

[**meissner@remss.com**](mailto:meissner@remss.com)**.**

# Overview

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Module | Driver Routine | Purpose | Input | Action / Output |
| 1 | Make\_SMAP\_L1B\_ingest\_V50.F90 | Prepare ingest.  Extract relevant fields from HDF files.  Write out as raw binary. | L1B asc (HDF)  L1B dsc (HDF) | L1B asc  L1B dsc  (raw binary) |
| 2 | Make\_SMAP\_L2A\_V50.F90 | L2A processor. Backus Gilbert OI of SMAP L1B TA onto fixed Earth grid. Concatenate asc/dsc orbits. | L1B asc  L1B dsc  (raw binary) | L2A  (raw binary) |
| 3 | Make\_SMAP\_L2B\_V50.F90 | L2B processor. Interface with ancillary fields. Append ancillary fields to L2A file. Write out as raw binary. | L2A  ancillary files | L2B  (raw binary) |
| 4 | PERFORM\_OCEAN\_ TARGET\_CALIBRATION.F90 | Perform ocean target calibration for orbit. This routine should be run BEFORE the L2C processor.  It calculates TA measured - expected for open ocean scenes and writes it to the binary direct access file *dtb\_statfile.* | L2B | update ocean target calibration table |
| 5 | Make\_SMAP\_L2C\_V50.F90 | L2C processor.  1. Read L2B files.  2. DTB correction from ocean target calibration.  3. Run RSS SMAP Salinity Algorithm. 4. Smooth SSS to 70km. 5. Run perturbed retrieval for uncertainty estimates. 6. Calculate orbital statistic and update direct access file *sss\_statfile*. 7. Create netCDF4 output. | L2B | L2C  (netCDF4)  update SSS  statistics table |
| 6 | Make\_SMAP\_L3\_8DAY\_V50.F90 | L3 8-day running average processor.  1. Read L2C files. 2. Time average for 8-day aggregates. 3. Formal error propagation from L2C to 8-day. 4. Create netCDF4 output. | L2C | L3 8-day  (netCDF4) |
| 7 | Make\_SMAP\_L3\_MONTHLY\_V50.F90 | L3 monthly average processor.  1. Read L2C files. 2. Time average for monthly aggregates. 3. Formal error propagation from L2C to month. 4. Create netCDF4 output. | L2C | L3 monthly  (netCDF4) |

# Instructions

## File Directory Structure

Text

Description automatically generated

Unzip all zip directories and put the files of each zip directory into a directory with the same name. This creates a directory structure under the main directory.

Each module can be run independently.

Sample input and output is collected in the subdirectory *sample\_data*.

## Land Tables

The large 1 GB binary file *land\_tables\_latlon\_12km.dat* needs to be placed into the *tables\_L2A* directory.

## Special Handling of Sun-Tables and Galaxy-Tables

Due to their large size the two tables *sun\_tables.dat* and *galaxy\_wind\_tables.dat* need to be split up into sub-tables and released separately.

See the corresponding releases and instructions how to generate the full tables form the smaller sub-tables.

Once the large sun and galaxy tables have been generated, both of them need to be placed into the *tables\_L2A* directory.

## Storage of NCEP Binary Files

The relevant variables of the NCEP GRIB files have been de-gribbed. They are stored as raw binary arrays with a header containing the time stamp (year, month, day, hour).

# Environment and Platforms

The processing code package has been implemented, tested, and run on Windows 8 64-bit.

Minimum memory requirement: 24 GB.

Microsoft Visual Studio 2008. Version 9.0.21022.8 RTM

Microsoft .NET Framework. Version 3.5 SP1

Installed Edition: Professional

Microsoft Visual Basic 2008 91605-130-0139771-60896

Microsoft Visual C# 2008 91605-130-0139771-60896

Microsoft Visual C++ 2008 91605-130-0139771-60896

Microsoft Visual Studio 2008 Tools for Office 91605-130-0139771-60896

Microsoft Visual Web Developer 2008 91605-130-0139771-60896

Crystal Reports AAJ60-G0MSA4K-68000CF

Intel(R) Visual Fortran Package ID: w\_fcompxe\_2011.4.196

Intel(R) Visual Fortran Composer XE 2011 Integration for Microsoft Visual Studio\* 2008, 12.0.3471.2008, Copyright (C) 2002-2011 Intel Corporation

\* Other names and brands may be claimed as the property of others.

Intel® C++ Composer XE 2011 Package ID: w\_ccompxe\_2011.4.196

Intel® C++ Composer XE 2011 Integration for Microsoft Visual Studio\* 2008, Version 12.0.1029.2008, Copyright© 2002-2011 Intel Corporation

\* Other names and brands may be claimed as the property of others.

# Compile and Link Options for Intel Compiler

## compile64

set linkerpath=C:\Program Files (X86)\Intel\ComposerXE-2011\bin

call "%linkerpath%\ifortvars.bat " intel64 vs2008

"%linkerpath%\intel64\ifort.exe" %1 /exe:%2 /module:C:\fortran\_tools\trash /fltconsistency /gen-interfaces:nosource /Qsave-temps- /Qsave /Qvec-report0 /heap-arrays:1000 /check:bounds,format,pointers,uninit /nologo /extend-source:132 /traceback /warn:unused,interfaces,uncalled /assume:byterecl x:\lib\x64\zlib64.lib /F40000000

## compile\_hdf5\_32

set d=x:\lib\x32\

set linkerpath=C:\Program Files (X86)\Intel\ComposerXE-2011\bin

call "%linkerpath%\ifortvars.bat " ia32 vs2008

"%linkerpath%\ia32\ifort.exe" %1 /exe:%2 /module:C:\fortran\_tools\trash /fltconsistency /gen-interfaces:nosource /Qsave-temps- /Qsave /Qvec-report0 /heap-arrays:1000 ^

/check:bounds,format,pointers,uninit /nologo /extend-source:132 /traceback /warn:unused,interfaces,uncalled /assume:byterecl %d%zlib32.lib %d%zlib.lib ^

%d%xdr.lib %d%libszip.lib %d%hdf\_fcstub.lib %d%hdf\_fortran.lib %d%mfhdf\_fortran.lib %d%mfhdf\_fcstub.lib %d%hd426.lib %d%hm426.lib %d%libjpeg.lib %d%WS2\_32.Lib /F40000000

## compile64\_netcdf

set fpath=%1

set fname=%2

set fname=%fname:"=%

set rss\_inc\_path=X:\fortran\_tools\include\latest\win

set rss\_lib\_path=X:\fortran\_tools\builds\latest\win

set nc\_path=X:\fortran\_tools\builds\latest\win\third\_party\netcdf4.4.3

set nc\_inc\_path=%nc\_path%\include

set nc\_lib\_path=%nc\_path%\lib

set ncf\_path=X:\fortran\_tools\builds\latest\win\third\_party\netcdf-fortran4.5.2

set ncf\_inc\_path=%ncf\_path%\include

set ncf\_lib\_path=%ncf\_path%\lib

set linkerpath=C:\Program Files (X86)\Intel\ComposerXE-2011\bin

echo In path: %fpath%

echo Out path: %fname%

call "%linkerpath%\ifortvars.bat" intel64 vs2008

REM CD to the root of the input file.

for %%A in (%fpath%) do (cd %%~dpA)

"%linkerpath%\intel64\ifort.exe" %fpath% /exe:%fname% /module:X:\intel11.1\fortran\_tools\trash /fltconsistency /gen-interfaces:nosource /Qsave-temps- ^

/Qsave /Qvec-report0 /heap-arrays:1000 /check:bounds,format,pointers,uninit /nologo /extend-source:132 /traceback /F40000000 ^

/warn:unused,interfaces,uncalled /assume:byterecl ^

/I %rss\_inc\_path% ^

/I %nc\_inc\_path% ^

/I %ncf\_inc\_path% ^

/link ^

/nodefaultlib:MSVCRTD ^

/nodefaultlib:LIBCMT ^

/libpath:%rss\_lib\_path% ^

/libpath:%nc\_lib\_path% ^

/libpath:%ncf\_lib\_path% ^

netcdf.lib ^

hdf5.lib ^

hdf5\_hl.lib ^

ZLIB64.lib ^

netcdff.lib ^

rssnetcdf.lib ^

erareader.lib ^

PyPlotter.lib ^

logger.lib ^

c\_integrations.lib

# Implementing and Running on Other Platforms

## Reading and Writing Raw Binary Files with gfortran.

In order to open raw binary files for read sequential access:

Rewrite:

open(unit=3,file=file\_header,action='read',form='binary',status='old').

To:   
open(unit=3,file=file\_header,action='read',form='unformatted',access='stream',status='old').   
  
In order to open raw binary files for write sequential access:

Rewrite:

open(unit=4,file=file\_heder, action='write’,form='binary', …).

To:   
open(unit=4,file=file\_header,action='write',form='unformatted',access='stream', …).