Algorithm Delivery and Integration

Carol C Davidson

MODIS/VIIRS Software Testing and Integration Team Lead

Science Systems and Application Inc.

Overview

- GitLab
- Software Delivery Process
- Software Integration
- VIIRS Collections C2 vs C1
- MODIS Collections C61 vs C7
- Migration of MODIS C61 PGEs to C7

GitLab

- Changing from subversion to GitLab for housing source code repository
  - MODIS C6.1 and VIIRS C2: Transition to Gitlab in progress (~80% completed)
  - Future MODIS C7 and VIIRS C2/C3: Science teams to deliver through Gitlab
  - In house Libraries, science codes, and third-party libraries
  - CentOS7 and Ubuntu supported currently, but will become all Ubuntu
- Science Team on-boarding for future deliveries (MODIS C61/C7 and VIIRS C2/C3) through GitLab
  - STIG will hold on-boarding sessions for science teams
  - GitLab access via EarthData Login (EDL) and NASA Launchpad (*more details available in backup slides*)
  - Interface for STMs to use for making code deliveries.
  - Learning Resources
    - Account access/setup go to: Accounts · Wiki · infrastructure / Help · GitLab (nasa.gov).
    - Basics on working with PGE source codes within GitLab MODAPS / Documentation · GitLab (nasa.gov)
    - More detailed User’s Guide pdf available as well.

Gitlab Organization

- Code packages are organized into projects within GitLab and projects are organized into groups.
- Top level group is named “MODAPS”
  - Contains projects for third-party libraries, codes shared between MODIS and VIIRS, and documentation.
  - Contains “MODIS” and “VIIRS” sub-groups
- “MODIS” group contains sub-groups “ATMOS”, “Library”, “shared_src”, “LAND”, and “INHOUSE, each of which hold the projects for PGEs and/or shared codes.
- “VIIRS” group contains projects for all PGEs, shared codes, and VIIRS specific libraries, like IDPS and OPS.

Software Delivery

- Overview of the basic steps for updating/delivering a PGE. (*actual sequence of steps for an example of PGE11 provided in Back up slides*)
  - Determine Project name and branch (i.e, PGE and collection) you want to update. For example, MODIS PGE11 for C6.1.
  - Clone the project to the command line
  - Switch to the C61 branch.
  - Create a new branch for your updates; the new branch will be based on the C61 branch you are sitting on.
  - Switch to the new branch you created and integrate your changes. Build and test the updates and when satisfied, send them back to the repository.
  - Verify code pushed back built successfully within GitLab and resolve issues if it did not.
  - Submit update for merging/integration by SSTG by creating a Merge Request

Software Integration and Unit Test

- SSTG steps with STM delivery:
  - Create a forked copy of the project to work in.
  - Pull the STM delivery branch to forked copy/check that branch out.
  - Review the STM delivery. Implement updates needed in PGE perl scripts for inputs, outputs, runtime parameter settings or production rules.
  - Document changes in PGE HISTORY file.
  - Commit and push any changes made to working copy in GitLab and verify successful build/deploy.
  - Create a tag of the new version and use to execute command line and MODAPS tests with.
  - If all tests complete successfully and no concerns, update original merge request with test information and approve merge request as the “reviewer”
  - CM completes merge and tagging of PGE

VIIRS C1 vs C2

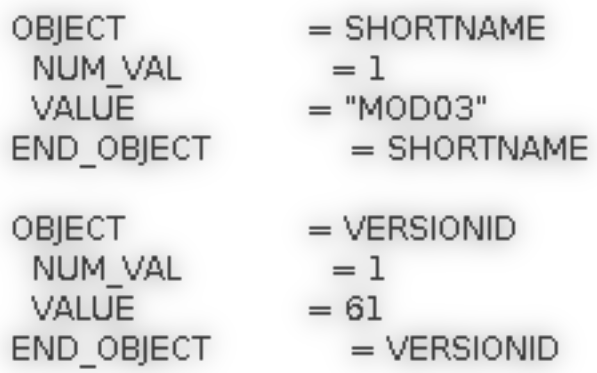
- C1 PGEs currently in subversion, (planned for migration to GitLab at a later time ?)
- C2 PGEs ported to Gitlab
  - 85% of existing land PGEs migrated to GitLab and successfully build to execute on Ubuntu system.
  - Ubuntu version of SNPP C2 and JPSS1 C2.1 Geolocation deployed from Gitlab in operational processing
  - All C1 land PGES updated to use NASA Geolocation and L1B data, both NetCDF4.2 format, as input.
  - Standard products generated in netcdf4/HDF5/HDFEOS5 format
  - Includes improvements to Land SIPS CloudMask, changes to metadata to be consistent with MODIS products.

MODIS C61 VS C7

- Output file format: NetCDF (HDF-EOS5) vs HDF4 (HDF-EOS2)
  - C61 : HDF4 format, support HDF-EOS Swath and Grid
  - C7 : HDF5 format, CF-1.6 compliant, support HDF-EOS Swath and Grid
- Metadata:
  - C61: nested within ‘CoreMetadata.x’ attribute; generated using SDP Toolkit and MCF
  - C7: global attributes for each metadata, Doesn’t use SDP Toolkit.
- CF-compliant
  - C7 output files will be CF compliant and can be georeferenced by NetCDF data viewers or tools
  - L2 Swath data: latitude and longitude are resampled and added for different sizes of data in the file by the C7 libraries.
  - L3 Grid data: projection parameter attributes will be added by the C7 libraries.
- PCF file
  - C7: ”key = value” format
  - C7: LUN number -> LUN string. E.g. 600000 -> MxD03
- MCF file
  - C7: “name|mandatory|type|num_val|value” format
- Toolkit
  - C61: Uses SDP Toolkit
  - C7: New API-trans and SDP-utils libraries developed by LDOPE/SSTG.

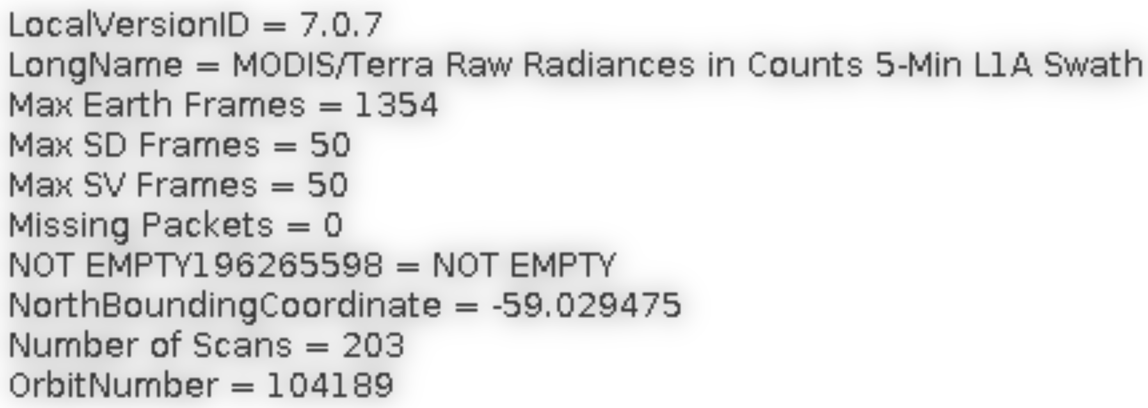
MODIS C7 VS C61

C61 metadata



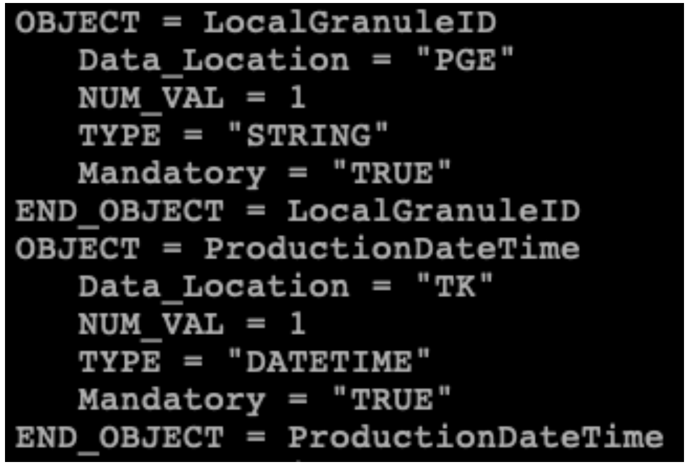
```
OBJECT              = SHORTNAME
  NUM_VAL             = 1
  VALUE             = "MOD03"
END_OBJECT            = SHORTNAME

OBJECT              = VERSIONID
  NUM_VAL             = 1
  VALUE             = 61
END_OBJECT            = VERSIONID
```

C7 metadata



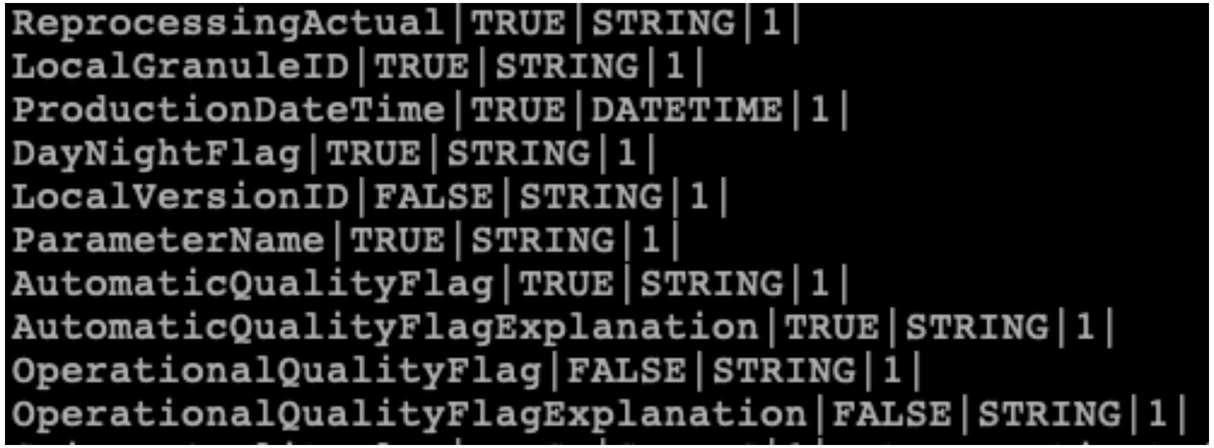
```
LocalVersionID = 7.0.7
LongName = MODIS/Terra Raw Radiances in Counts 5-Min L1A Swath
Max Earth Frames = 1354
Max SD Frames = 50
Max SV Frames = 50
Missing Packets = 0
NOT EMPTY196265598 = NOT EMPTY
NorthBoundingCoordinate = -59.029475
Number of Scans = 203
OrbitNumber = 104189
```

C61 MCF File



```
OBJECT = LocalGranuleID
   Data_Location = "PGE"
   NUM_VAL = 1
   TYPE = "STRING"
   Mandatory = "TRUE"
END_OBJECT = LocalGranuleID
OBJECT = ProductionDateTime
   Data_Location = "TK"
   NUM_VAL = 1
   TYPE = "DATETIME"
   Mandatory = "TRUE"
END_OBJECT = ProductionDateTime
```

C7 MCF File

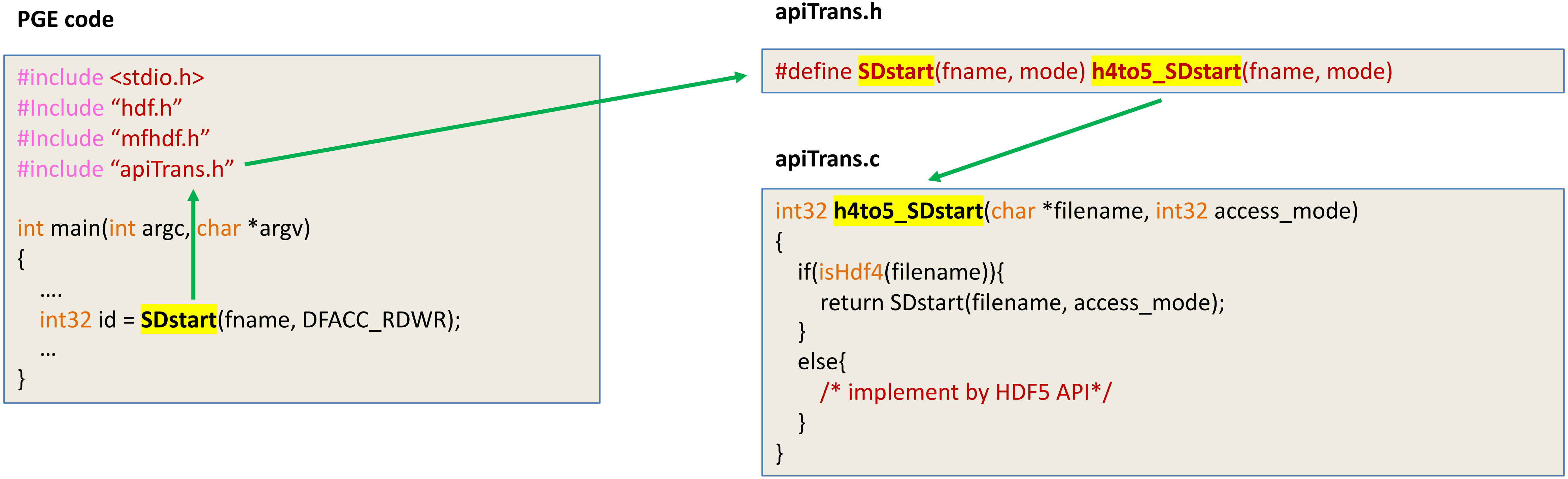


```
ReprocessingActual|TRUE|STRING|1|
LocalGranuleID|TRUE|STRING|1|
ProductionDateTime|TRUE|DATETIME|1|
DayNightFlag|TRUE|STRING|1|
LocalVersionID|FALSE|STRING|1|
ParameterName|TRUE|STRING|1|
AutomaticQualityFlag|TRUE|STRING|1|
AutomaticQualityFlagExplanation|TRUE|STRING|1|
OperationalQualityFlag|FALSE|STRING|1|
OperationalQualityFlagExplanation|FALSE|STRING|1|
```

Migration of MODIS C61 to C7

How C7 libraries work

The New API-trans and SDP-util libraries developed by LDOPE/SSTG provides HDF4 to HDF5 API implementation for every HDF4 call in the current operational MODIS PGEs.



PGE code

```
#include <stdio.h>
#Include “hdf.h”
#Include “mfhdf.h”
#include “apiTrans.h”

int main(int argc, char *argv)
{
   ….
   int32 id = SDstart(fname, DFACC_RDWR);
   …
}
```

apiTrans.h

```
#define SDstart(fname, mode) h4to5_SDstart(fname, mode)
```

apiTrans.c

```
int32 h4to5_SDstart(char *filename, int32 access_mode)
{
   if(isHdf4(filename)){
      return SDstart(filename, access_mode);
   }
   else{
      /* implement by HDF5 API*/
   }
}
```

Migration of MODIS C61 to C7

How to convert C61 PGE to C7

- Add ”apiTrans.h” to C files which include HDF4 or HDF-EOS2 calls;
- Add compile and link flags to makefile for C7 Libraries (API-Trans and SDP-Utils);
- Modify the Perl script to use new Perl libraries for C7 PCF format;
- Use provided tool to convert MCF files to C7 format;

Note: C61 metadata are nested in a “CoreMetadata” while C7 are saved as global attributes for each of them. <u>If PGE code uses SDP-Toolkit API to read/write metadata then nothing needs to be done; If PGE code parses the “CoreMetadata” string by itself then it will not work in C7.</u>

Migration of MODIS C61 to C7
How to update C7 PGE code by science team

- The PGE code is basically the same as C61 – still using HDF4/HDF-EOS2 function calls. Continue using HDF4/HDF-EOS2/SDP-Toolkit APIs if new attributes or SDS are added.
- LDOPE and SSTG will create baseline C7 PGEs by taking operational C7 PGEs and integrating API related changes. STMs focus on scientific changes.

Note: C7 PGEs will be managed by Gitlab. Here is a Wiki how to update a PGE in Gitlab and run tests: https://gitlab.modaps.eosdis.nasa.gov/modaps/documentation/-/wikis/home

Backup Slides

GitLab Access

- URL for MODAPS GitLab repository is Projects · Dashboard · GitLab (nasa.gov)
- Preferred access method is NASA LaunchPad.
- LaunchPad account holder should request access to the GitLab area via NAMS at https://idmax.nasa.gov/nams/asset/254154/670356850
- STMs without LaunchPad accounts may connect using EarthData login.
- For LaunchPad or EarthData, users must be added to appropriate user groups in order to access content.
- SSTG can contact appropriate admin for gaining access needed.

Software Delivery (1 of 4)

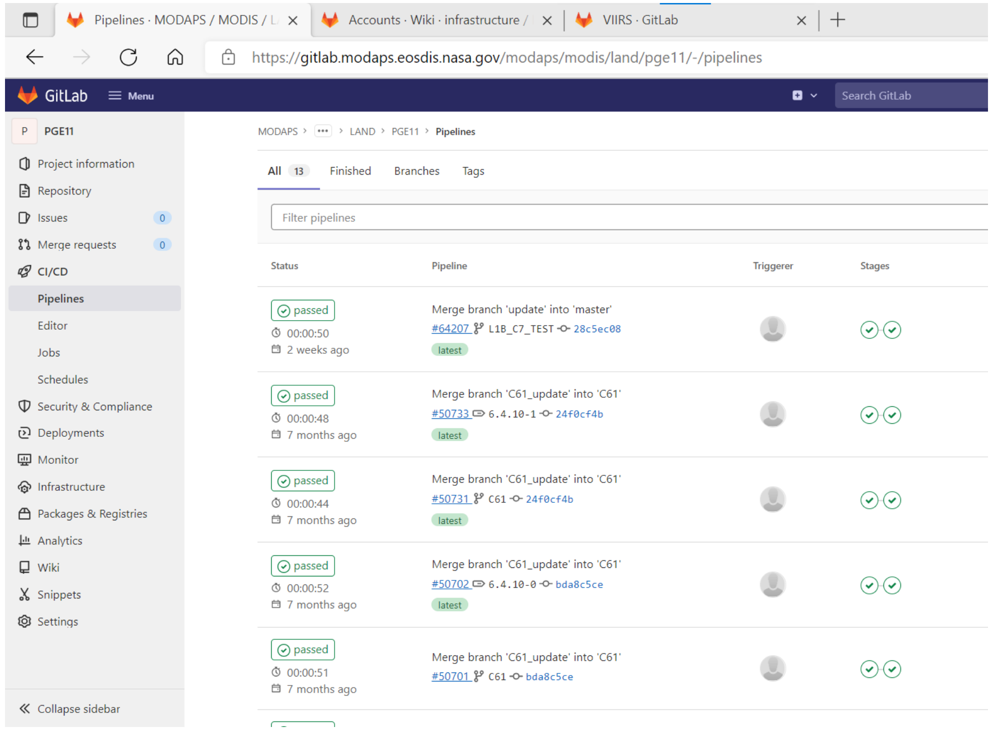
- Actual steps for update and delivery using mix of command line and web interface.
  - Clone the MODIS PGE11 project to the command line:

    git clone https://GitLab.modaps.eosdis.nasa.gov/modaps/modis/land/pge11.git
  - Switch to the pge11 directory created, go to the C61 branch so you start with the C61 code stored there:
    - git checkout C61
  - Create a new branch identical to the C61 branch for your updates using:
    - git checkout –b pge11update0422
  - Integrate your changes and then send them back to the repository using:
    - git add . (to add any new files to the repository)
    - git commit –am “<note about files modified/added/deleted>”
    - git push origin pge11update0422

Continued on next slide

Software Delivery (2 of 4)

- Via GitLab web interface, go to project https://GitLab.modaps.eosdis.nasa.gov/modaps/modis/land/pge11.git
- From left hand menu click on “CI/CD” to view pipelines – this is where you verify the code you pushed built successfully.

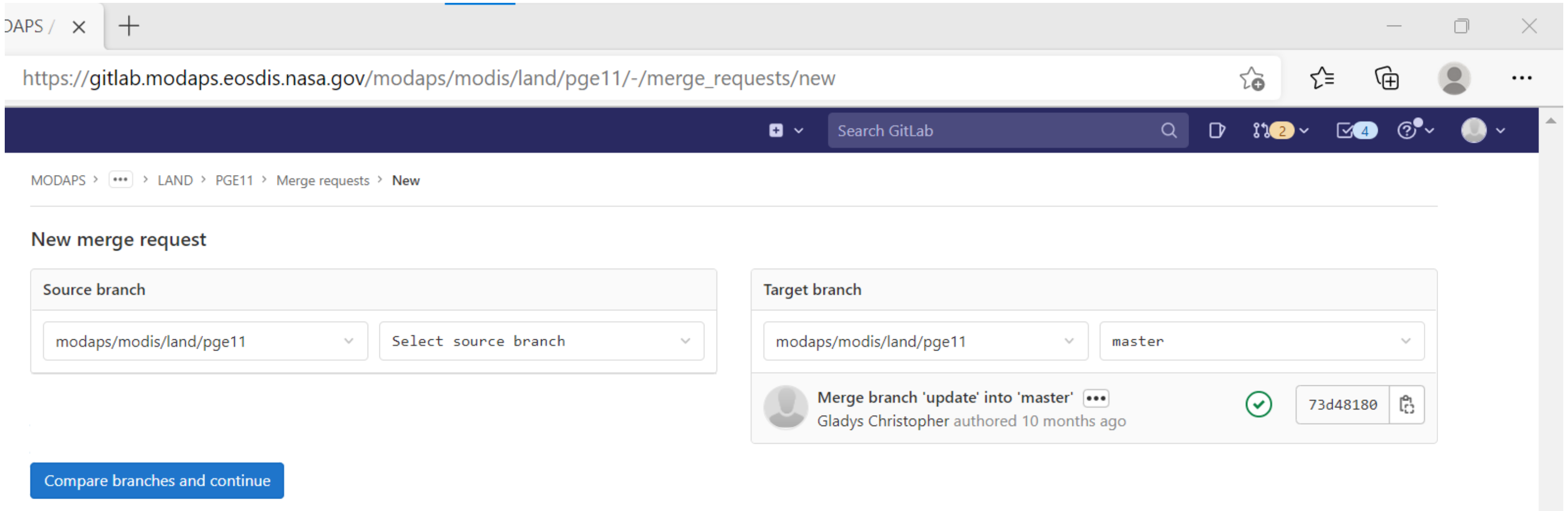


Status	Pipeline	Triggerer	Stages
passed 00:00:50 2 weeks ago	Merge branch 'update' into 'master' #64207 L1B_C7_TEST 28c5ec08 latest		✓ ✓
passed 00:00:48 7 months ago	Merge branch 'C61_update' into 'C61' #50733 6.4.10-1 24f0cf4b latest		✓ ✓
passed 00:00:44 7 months ago	Merge branch 'C61_update' into 'C61' #50731 C61 24f0cf4b latest		✓ ✓
passed 00:00:52 7 months ago	Merge branch 'C61_update' into 'C61' #50702 6.4.10-0 bda8c5ce latest		✓ ✓
passed 00:00:51 7 months ago	Merge branch 'C61_update' into 'C61' #50701 C61 bda8c5ce		✓ ✓

Continued on next slide

Software Delivery (3 of 4)

- Once pipeline is successful, submit your changes via a Merge Request
- From left hand menu click on "Merge Requests"
- On next page select "New merge request" button.
- On "New merge request" page, select the "source" and "target" branches. For our example, "source" is "pge11update0422", and "target" is "C61".
- Click on "Compare branches and continue"



https://gitlab.modaps.eosdis.nasa.gov/modaps/modis/land/pge11/-/merge_requests/new

MODAPS › ••• › LAND › PGE11 › Merge requests › New

New merge request

Source branch: modaps/modis/land/pge11 | Select source branch

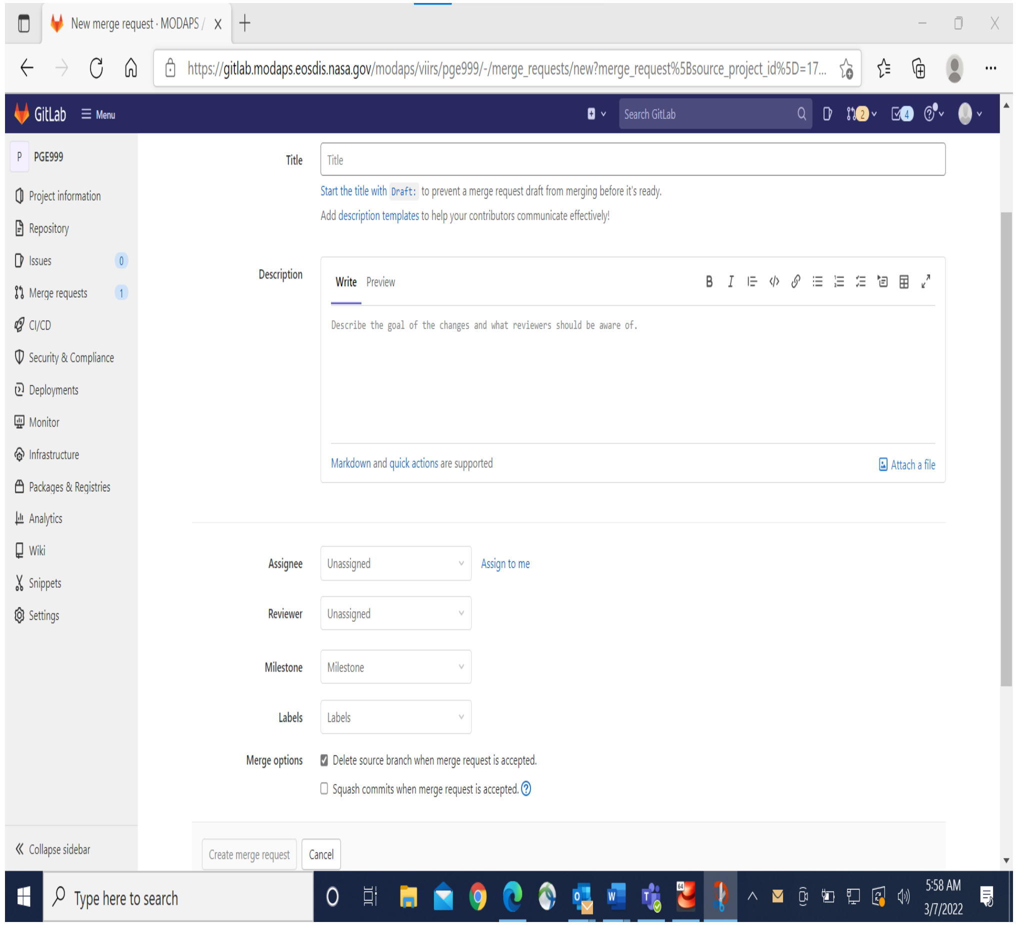
Target branch: modaps/modis/land/pge11 | master

Merge branch 'update' into 'master'
Gladys Christopher authored 10 months ago
73d48180

Continued on next slide

Software Delivery (4 of 4)

- On next “New merge request” page, fill out information as requested.
  - Title: this should indicate PGE and/or process name and version, for example PGE11 v6.1.20 update
  - Description: summary of changes made and what testing was done.
  - Assignee: select “modiscm”
  - Reviewer: Can be left blank or set to STIG member you communicated with regarding change.
  - Merge options: the “Delete source branch when merge request accepted” is selected by default. Deselect this option if you wish to retain the branch.
  - Click on “Create merge request”



https://gitlab.modaps.eosdis.nasa.gov/modaps/viirs/pge999/-/merge_requests/new?merge_request%5Bsource_project_id%5D=17...

Title: Title

Start the title with `Draft:` to prevent a merge request draft from merging before it's ready.

Add description templates to help your contributors communicate effectively!

Description: Describe the goal of the changes and what reviewers should be aware of.

Markdown and quick actions are supported

Assignee: Unassigned

Reviewer: Unassigned

Milestone: Milestone

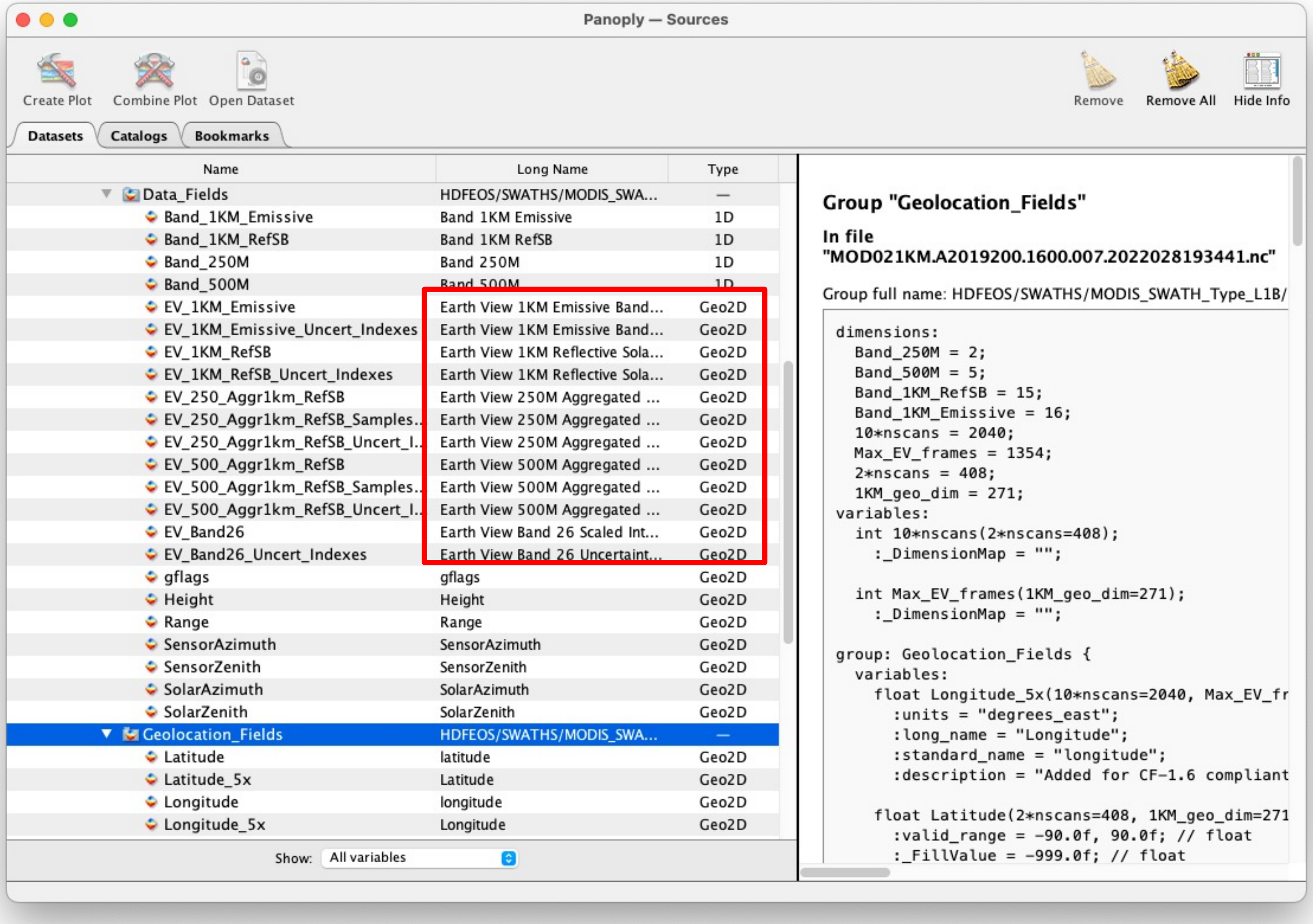
Labels: Labels

Merge options: ☑ Delete source branch when merge request is accepted.

□ Squash commits when merge request is accepted.

Display C7 L2 data in Panoply

Data with different resolution are all georeferenced



Name	Long Name	Type
Data_Fields	HDFEOS/SWATHS/MODIS_SWA...	—
Band_1KM_Emissive	Band 1KM Emissive	1D
Band_1KM_RefSB	Band 1KM RefSB	1D
Band_250M	Band 250M	1D
Band_500M	Band 500M	1D
EV_1KM_Emissive	Earth View 1KM Emissive Band...	Geo2D
EV_1KM_Emissive_Uncert_Indexes	Earth View 1KM Emissive Band...	Geo2D
EV_1KM_RefSB	Earth View 1KM Reflective Sola...	Geo2D
EV_1KM_RefSB_Uncert_Indexes	Earth View 1KM Reflective Sola...	Geo2D
EV_250_Aggr1km_RefSB	Earth View 250M Aggregated ...	Geo2D
EV_250_Aggr1km_RefSB_Samples...	Earth View 250M Aggregated ...	Geo2D
EV_250_Aggr1km_RefSB_Uncert_I...	Earth View 250M Aggregated ...	Geo2D
EV_500_Aggr1km_RefSB	Earth View 500M Aggregated ...	Geo2D
EV_500_Aggr1km_RefSB_Samples...	Earth View 500M Aggregated ...	Geo2D
EV_500_Aggr1km_RefSB_Uncert_I...	Earth View 500M Aggregated ...	Geo2D
EV_Band26	Earth View Band 26 Scaled Int...	Geo2D
EV_Band26_Uncert_Indexes	Earth View Band 26 Uncertaint...	Geo2D
gflags	gflags	Geo2D
Height	Height	Geo2D
Range	Range	Geo2D
SensorAzimuth	SensorAzimuth	Geo2D
SensorZenith	SensorZenith	Geo2D
SolarAzimuth	SolarAzimuth	Geo2D
SolarZenith	SolarZenith	Geo2D
Geolocation_Fields	HDFEOS/SWATHS/MODIS_SWA...	—
Latitude	latitude	Geo2D
Latitude_5x	Latitude	Geo2D
Longitude	longitude	Geo2D
Longitude_5x	Longitude	Geo2D

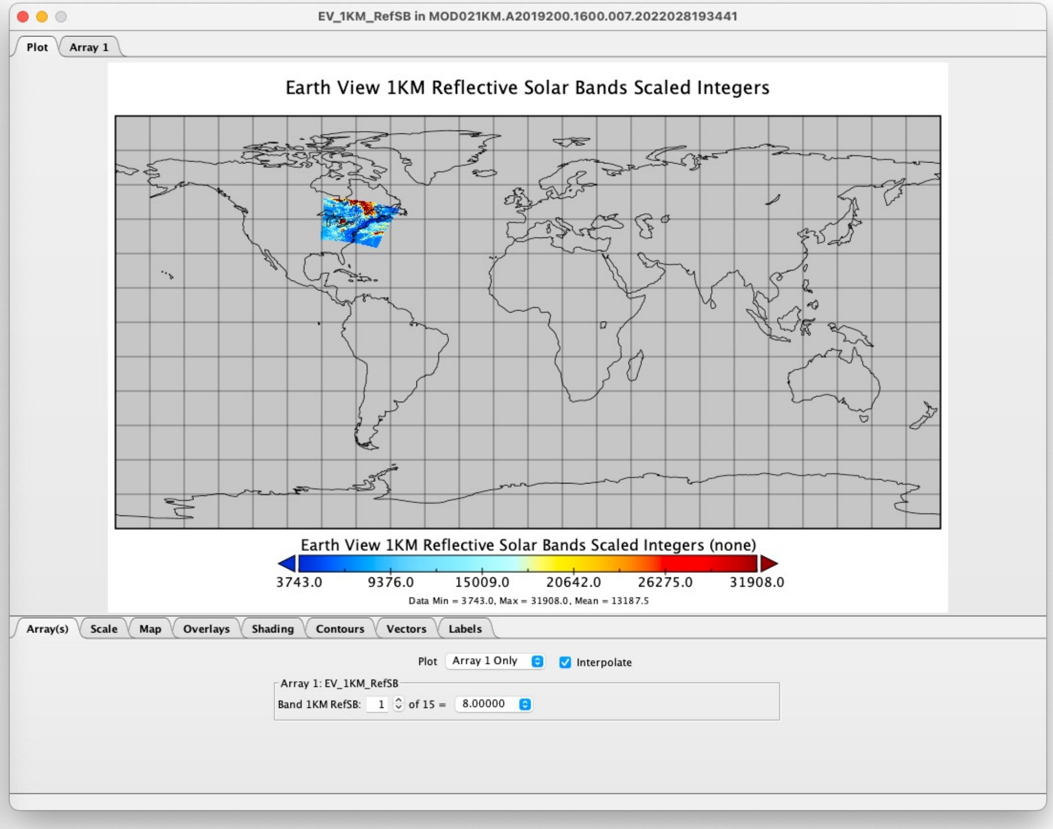
Show: All variables

Group "Geolocation_Fields"

In file "MOD021KM.A2019200.1600.007.2022028193441.nc"

Group full name: HDFEOS/SWATHS/MODIS_SWATH_Type_L1B/

```
dimensions:
  Band_250M = 2;
  Band_500M = 5;
  Band_1KM_RefSB = 15;
  Band_1KM_Emissive = 16;
  10*nscans = 2040;
  Max_EV_frames = 1354;
  2*nscans = 408;
  1KM_geo_dim = 271;
variables:
  int 10*nscans(2*nscans=408);
    :_DimensionMap = "";

  int Max_EV_frames(1KM_geo_dim=271);
    :_DimensionMap = "";

group: Geolocation_Fields {
  variables:
    float Longitude_5x(10*nscans=2040, Max_EV_fr
      :units = "degrees_east";
      :long_name = "Longitude";
      :standard_name = "longitude";
      :description = "Added for CF-1.6 compliant

    float Latitude(2*nscans=408, 1KM_geo_dim=271
      :valid_range = -90.0f, 90.0f; // float
      :_FillValue = -999.0f; // float
```


EV_1KM_RefSB in MOD021KM.A2019200.1600.007.2022028193441
Plot
Array 1
Earth View 1KM Reflective Solar Bands Scaled Integers
Earth View 1KM Reflective Solar Bands Scaled Integers (none)
3743.0
9376.0
15009.0
20642.0
26275.0
31908.0
Array(s)
Scale
Map
Overlays
Shading
Contours
Vectors
Labels

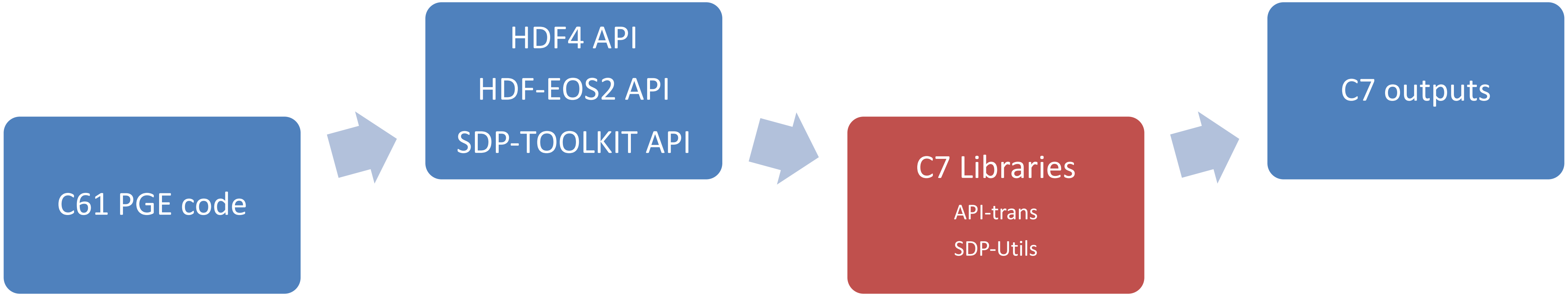
Data Min = 3743.0, Max = 31908.0, Mean = 13187.5

Plot: Array 1 Only ☑ Interpolate

Array 1: EV_1KM_RefSB

Band 1KM RefSB: 1 of 15 = 8.00000

How C7 libraries work



- C7 Libraries will ”translate” HDF4 / HDF-EOS2 API function calls to HDF5 / HDF-EOS5;
- C7 Libraries will handle the new format of PCF and MCF;
- C61 PGE code need no change – except to include a “apiTrans.h” header in some C files.