PDS4 in GDAL

PSIDA, April 2018

Trent Hare and Lisa Gaddis

thare@usgs.gov

What Domain?

- PDS4 supports dozens of domains
  - Atmospheres (GCMs)
  - Planetary Interiors
  - Astronomy
  - Astrobiology
  - etc., etc., etc.

GDAL is Geospatial – “tied to a planetary surface”

What is GDAL

Geospatial Data Abstraction Library

- GDAL is a “translator **library** for raster geospatial data formats”
- Open source (and **community** driven)
- Used in many “geo” applications: QGIS, UDIG, SAGA, ARCMAP, GMT, MapServer (WMS), Google Earth, AMES Stereo-pipeline, SOCET GXP, …
- Handles many image formats for read and slightly less for writing: PDS3, ISIS2/3, VICAR, **FITS** (via CFITSIO), ENVI, GeoTiff, Jpeg2000, PNG, cloud-based, NetCDF, … - **over 150 formats**

More info

What is GDAL

Geospatial Data Abstraction Library

- Started in 1998 by Frank Warmerdam
- A project of OSGeo since 2008
- MIT/X Open Source license (permissive)
- >1M lines of code for library and utilities
- > 150K lines of tests in Python

From: Even Rouault (link)

Adding PDS4 -- Contract

- Open bid awarded to Hobu Inc. https://hobu.co/ GDAL and LIDAR specialist.
- 3 months allocated. Delivered in 2 months (1 month for testing).

- Overview:
  - Incorporate PDS4 **read and write** capabilities into existing GDAL C/C++.
  - PDS4 keywords to be accessed programmatically (C++ or Python).
  - Allow for detached PDS4 label that is pointed into a “raw” GeoTiff.
  - All code available and supported under the standard GDAL release.

Beta released in trunk (Sep. 12th, 2017)

- Initial testing finalized on Linux RedHat
- Astro has built a **Anaconda** environment (for low-level Python API support)
- 18 Unit tests provided with delivery:
  https://trac.osgeo.org/gdal/browser/trunk/autotest/gdrivers/pds4.py
- Initial help page:
  - http://www.gdal.org/frmt_pds4.html

Reign in expectations

- A full "GDAL" solution relies on a **good PDS4 template**.
  - *Similar how FGDC metadata works in the real world. Metadata is never fully automatable.*
- Only the **image's physical parameters** (lines, samples, bit type) and **map projection** are <u>automated</u>.
- But GDAL supports template **variables**, so scripting can support a full EDR label using Python, PERL, etc.

Simple scripting examples on GitHub

- Written In Python, but shows how Bash, PERL, etc. could be used

https://github.com/USGS-Astrogeology/GDAL_scripts/tree/master/PDS4gdal

```
observeID = getkey(from_=inputlbl, keyword='InstrumentId', grp='Archive')

theLine = '-co VAR_OBSERVING_SYSTEM_NAME={}'.format(observeID)

fileConfig.write(theLine)
```

Pros

- Supports writing and **full XML schema validation** (via Xerces).
- Convert from most PDS3, ISIS2/3, VICAR, FITS, GeoTiff, Jpeg2000
- With XML template, GDAL can support fully **compliant** PDS4 label
- Templates can be used from an “http” address
    - e.g. mission website, PDS, github link
- Template **variables** allow for simple scripting
- Full XML access in Python using lxml

Pros (cont.)

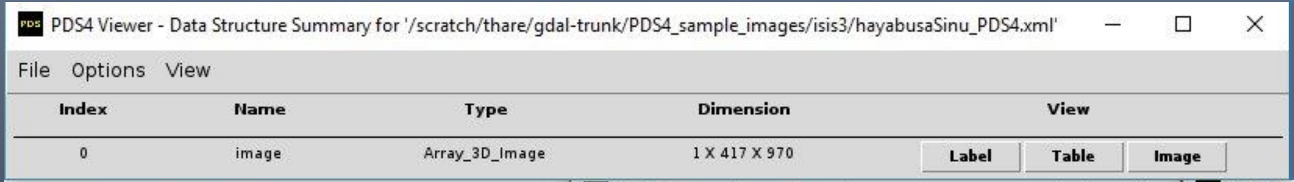
- Supports writing a PDS4 label pointing into a GeoTiff (`interoperability`)
- Array_xD_Type defaults to Array_3D_Image (but can be user defined)
- Builds on all OSs (even on cell phones ;-)
- PDS4 Driver will eventually show up in GIS apps like QGIS, ArcGIS Pro, SAGA, GMT, GRASS…

Cons

- Targeted specifically to support map projected data sets
- No PDS4 **table** support (yet).
- No support for writing multiple image arrays in one file (called sub-datasets).
- XML Templates requires more design up-front and more input from data provider
    - Leverages existing PDS4 label helpers (PLAID, OLAF, …).

Conversion examples (viewed in PDS4_Viewer)

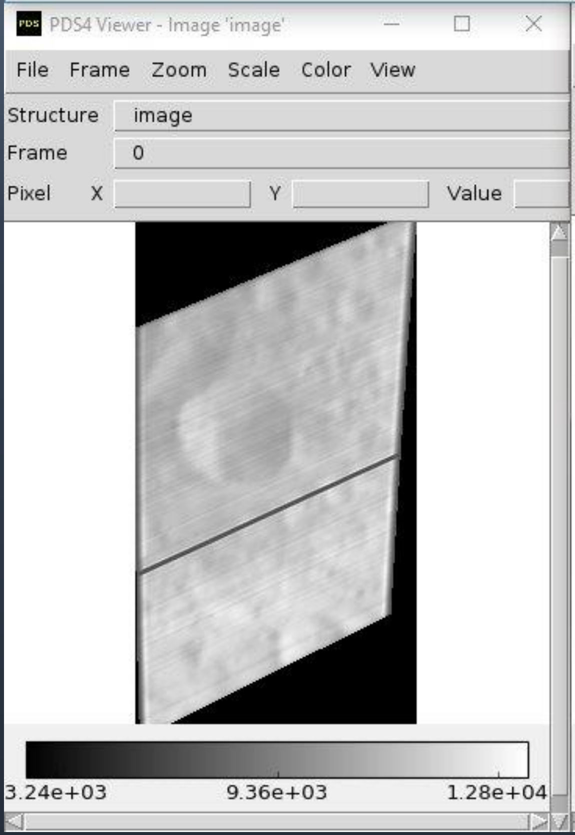
Dawn, Galileo, Hayabusa (Mercator, Sinusoidal)

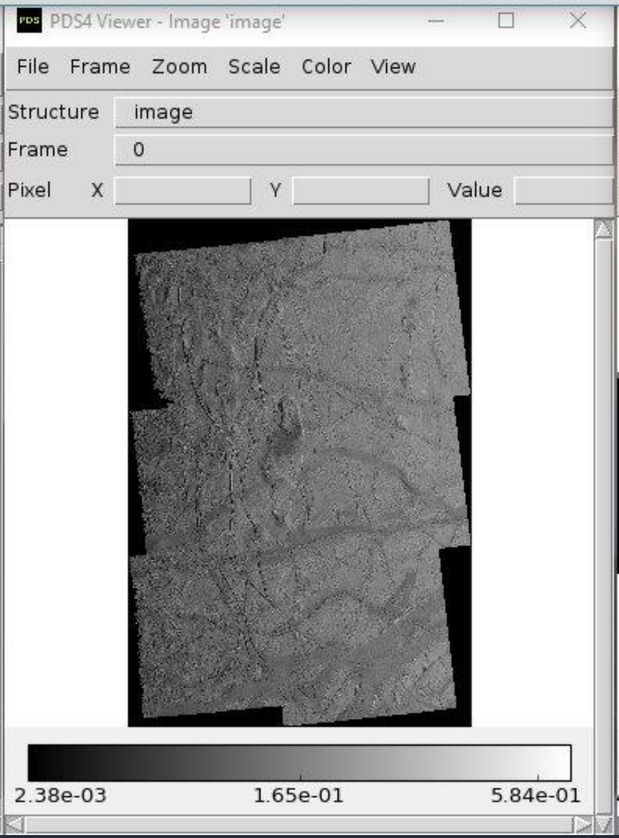


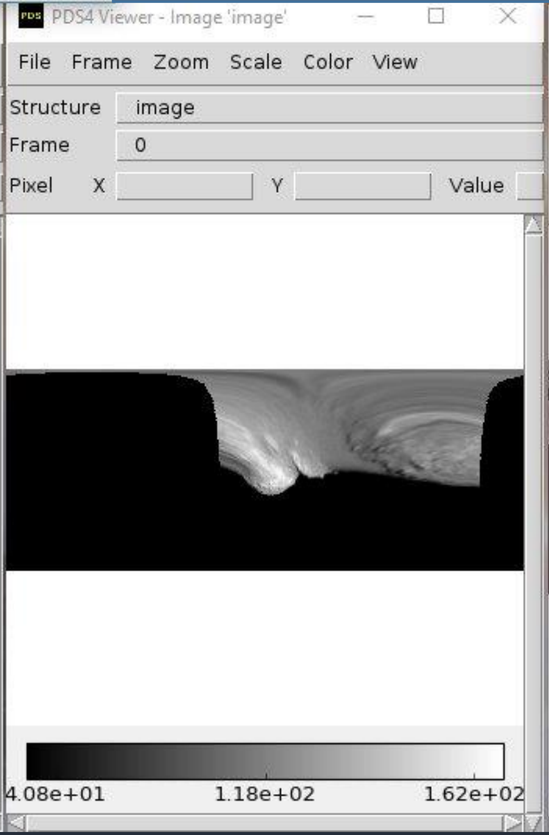
PDS4 Viewer - Data Structure Summary for '/scratch/thare/gdal-trunk/PDS4_sample_images/isis3/hayabusaSinu_PDS4.xml'

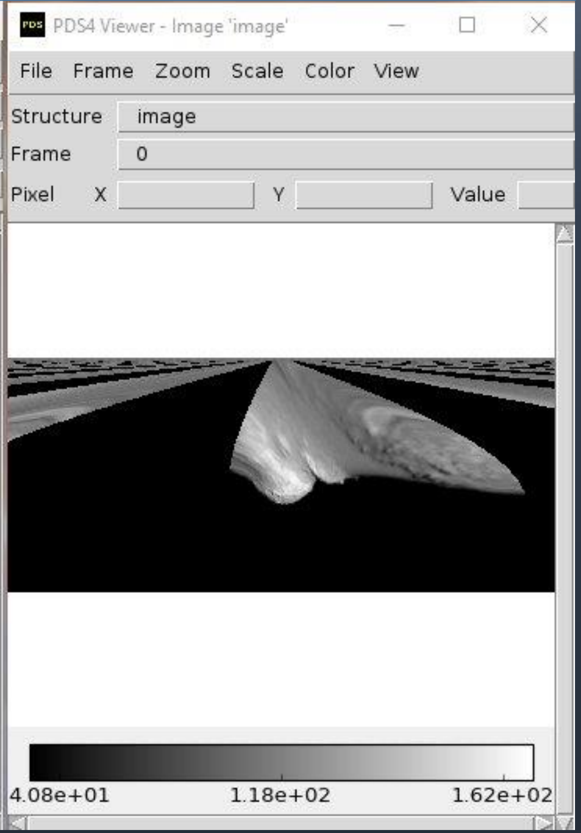
File Options View

Index	Name	Type	Dimension	View
0	image	Array_3D_Image	1 X 417 X 970	Label Table Image



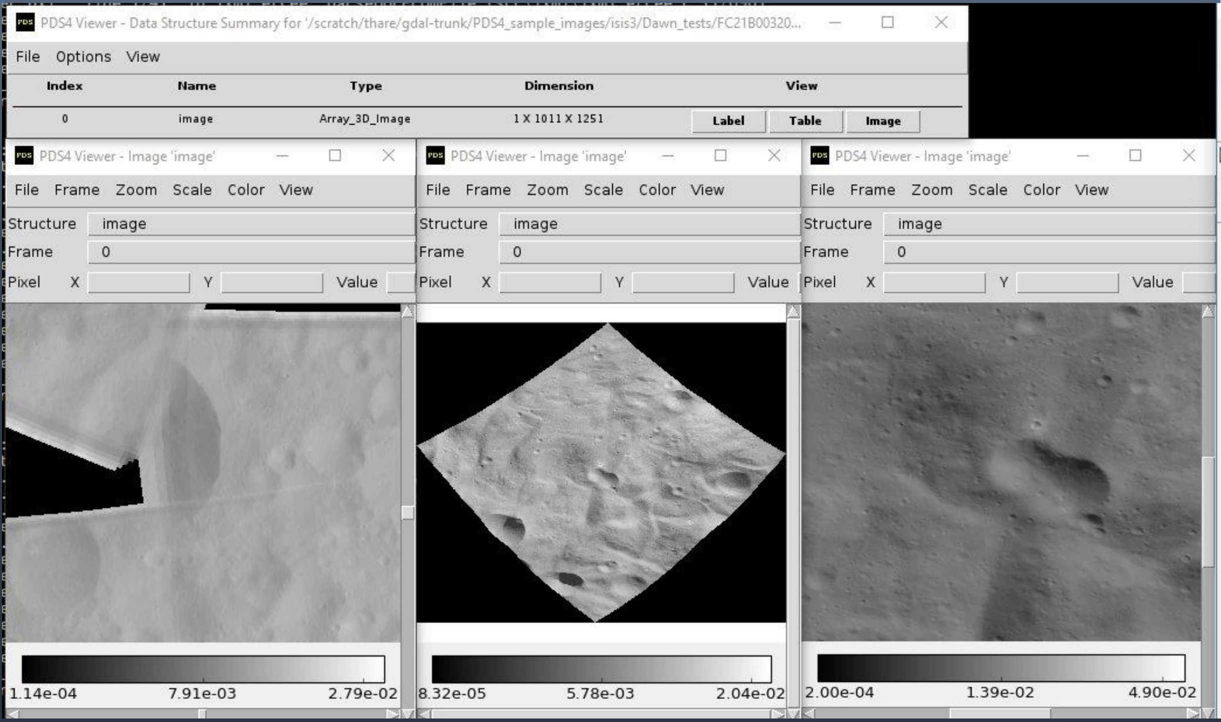






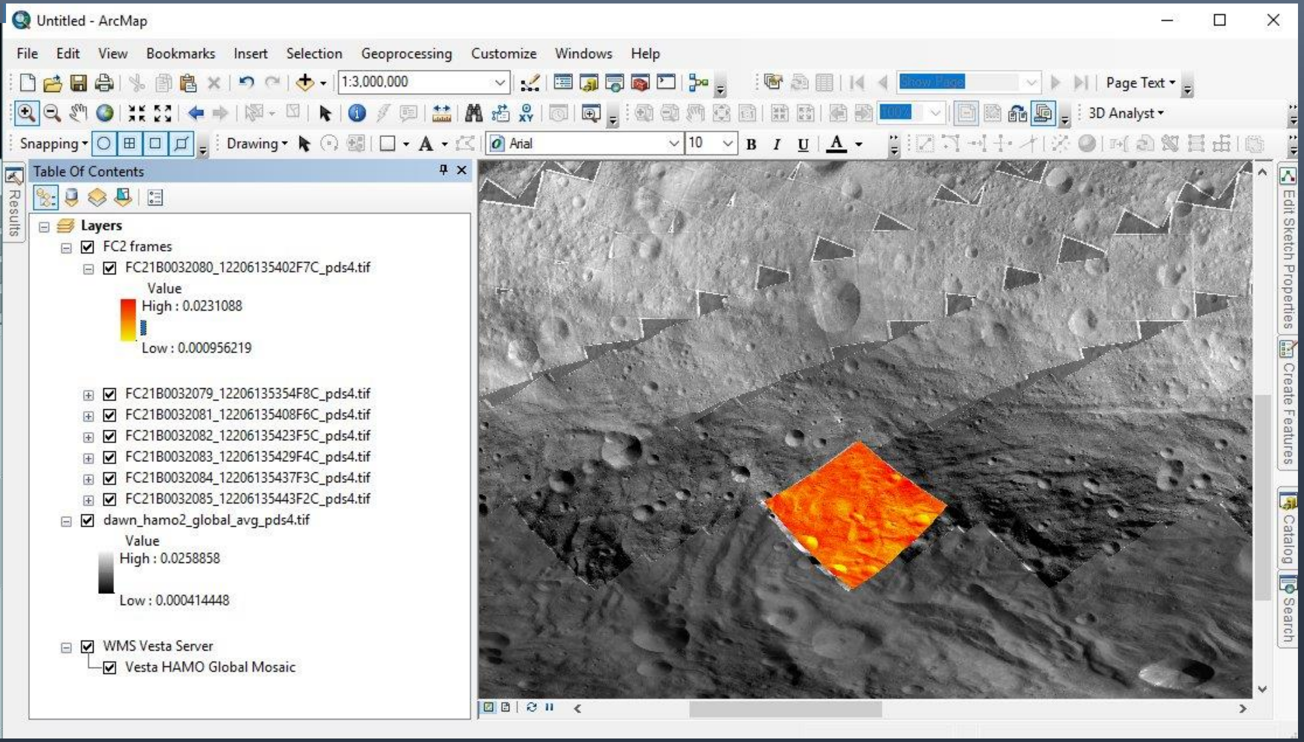
Conversion examples (PDS4_Viewer, ArcMap)

Dawn (Equirectangular)


PDS4 Viewer - Data Structure Summary for '/scratch/thare/gdal-trunk/PDS4_sample_images/isis3/Dawn_tests/FC21B00320...

Index	Name	Type	Dimension	View
0	image	Array_3D_Image	1 X 1011 X 1251	Label Table Image

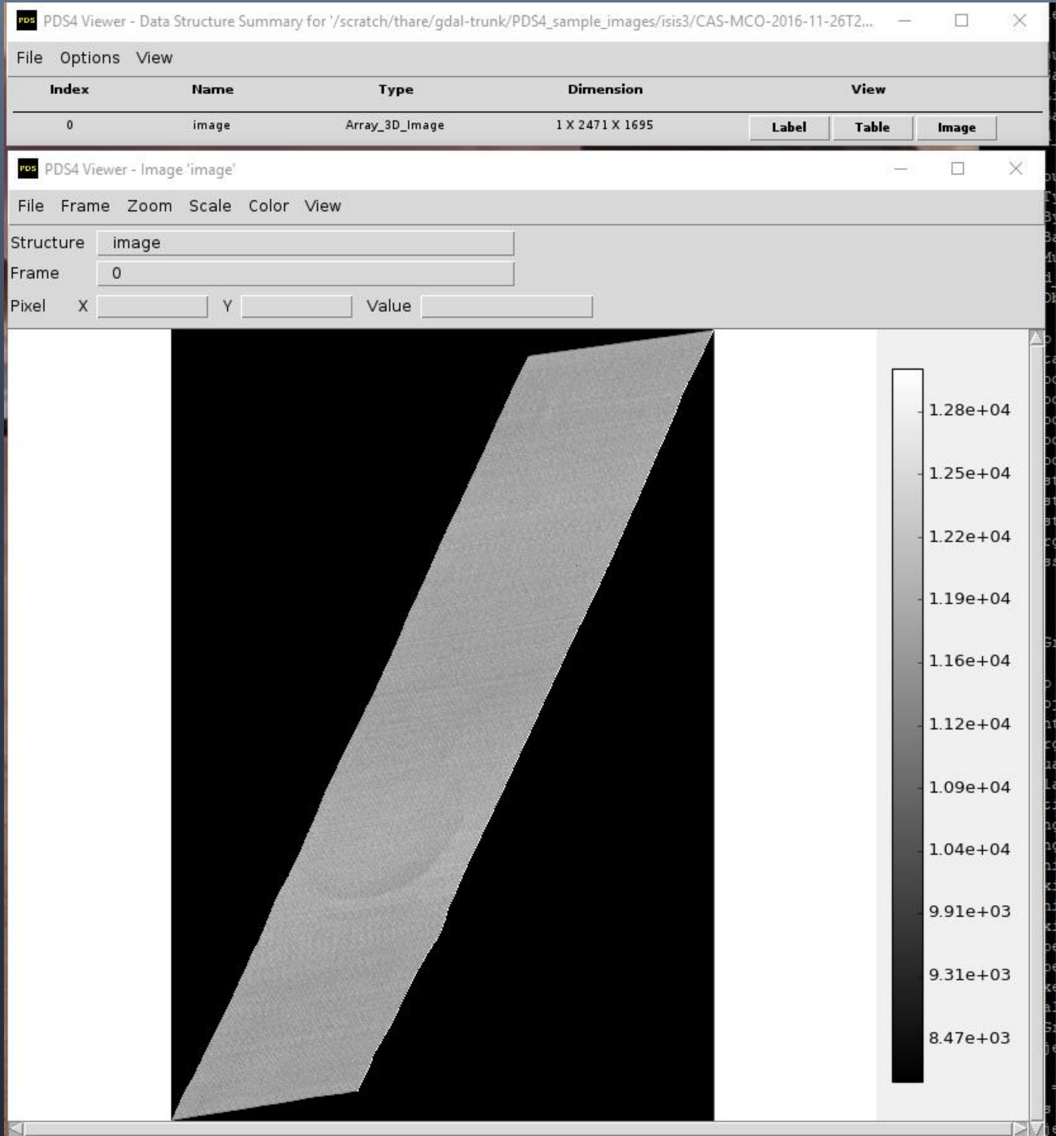
File Options View
PDS4 Viewer - Image 'image'
File Frame Zoom Scale Color View
Structure image
Frame 0
Pixel X Y Value
1.14e-04 7.91e-03 2.79e-02
PDS4 Viewer - Image 'image'
File Frame Zoom Scale Color View
Structure image
Frame 0
Pixel X Y Value
8.32e-05 5.78e-03 2.04e-02
PDS4 Viewer - Image 'image'
File Frame Zoom Scale Color View
Structure image
Frame 0
Pixel X Y Value
2.00e-04 1.39e-02 4.90e-02



Layers
- FC2 frames
  - FC21B0032080_12206135402F7C_pds4.tif
    - Value
    - High : 0.0231088
    - Low : 0.000956219
  - FC21B0032079_12206135354F8C_pds4.tif
  - FC21B0032081_12206135408F6C_pds4.tif
  - FC21B0032082_12206135423F5C_pds4.tif
  - FC21B0032083_12206135429F4C_pds4.tif
  - FC21B0032084_12206135437F3C_pds4.tif
  - FC21B0032085_12206135443F2C_pds4.tif
- dawn_hamo2_global_avg_pds4.tif
  - Value
  - High : 0.0258858
  - Low : 0.000414448
- WMS Vesta Server
  - Vesta HAMO Global Mosaic

Conversion examples (PDS4_Viewer, ArcMap)

CASSIS


PDS4 Viewer - Data Structure Summary for '/scratch/thare/gdal-trunk/PDS4_sample_images/isis3/CAS-MCO-2016-11-26T2...

Index	Name	Type	Dimension	View
0	image	Array_3D_Image	1 X 2471 X 1695	Label Table Image

PDS4 Viewer - Image 'image'

Structure: image
Frame: 0
Pixel X Y Value

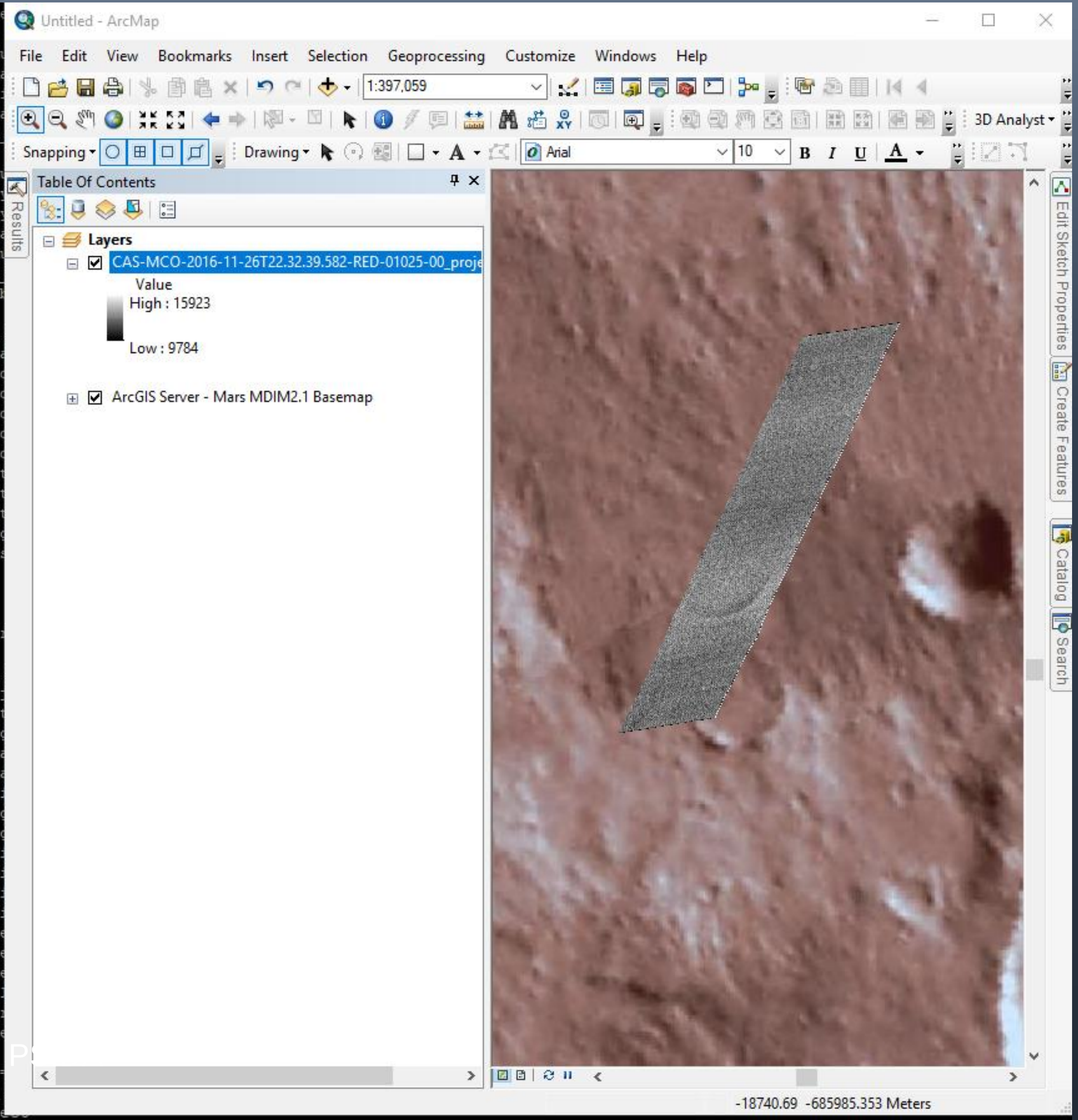

Untitled - ArcMap

Table Of Contents
Layers
CAS-MCO-2016-11-26T22.32.39.582-RED-01025-00_proje
Value
High : 15923
Low : 9784
ArcGIS Server - Mars MDIM2.1 Basemap

-18740.69 -685985.353 Meters

PDS4 XML examples

- Very minimal PDS4 XML template (ships with GDAL binaries)
  - https://www.dropbox.com/s/px97xm57n2q6a83/pds4_template.xml?dl=0

- Dawn output example w/ GDAL-added image section and map projection
  - https://www.dropbox.com/s/b69cejqvum181dl/dawnEqui_pds4.xml?dl=0

- LOLA output example with GDAL added image params and map projection
  - https://www.dropbox.com/s/pn2dwq5mdz702cx/ldem_4_pds4.xml?dl=0

Current testing environments (GDAL 2.3.0 beta1)

1. Build from GDAL trunk: http://www.gdal.org/daily

2. Anaconda (Python)
    - **$ conda install -c usgs-astrogeology gdal-pds4**

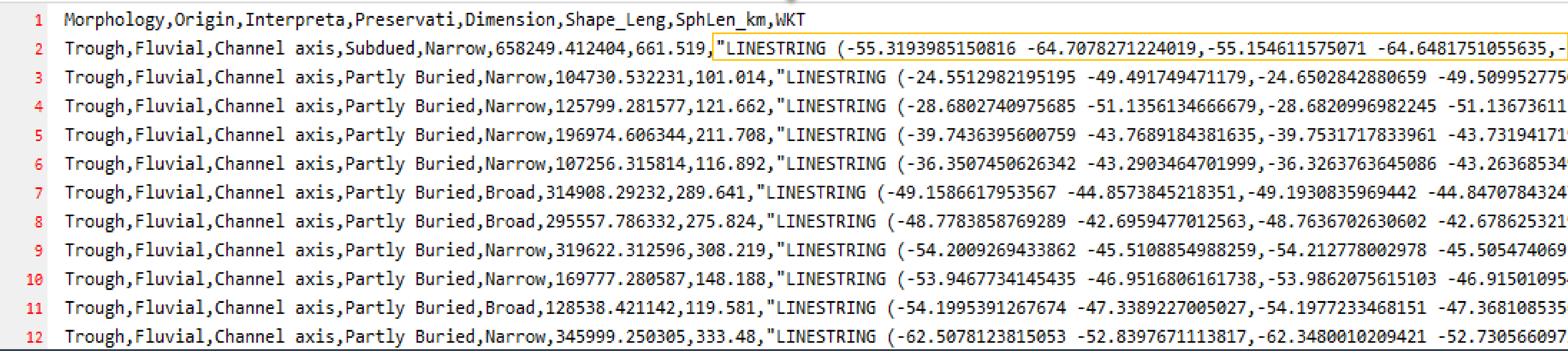
    - or install a whole environment with the required packages:
    - **$ conda env install usgs-astrogeology/astrogdal**

Next contract

- Add PDS4 **table** support.
- Write multiple image arrays in one file (called sub-datasets).
- Add units within “image” label section

- Vector support
  1. If PDS4 table has Latitude / Longitude field (acts as point)
  2. If PDS4 table has Well-Known-Text (WKT) field then supports points, lines, polygons

Proposed PDS4 ASCII Table example:



```
Morphology,Origin,Interpreta,Preservati,Dimension,Shape_Leng,SphLen_km,WKT
Trough,Fluvial,Channel axis,Subdued,Narrow,658249.412404,661.519,"LINESTRING (-55.3193985150816 -64.7078271224019,-55.154611575071 -64.6481751055635,-
Trough,Fluvial,Channel axis,Partly Buried,Narrow,104730.532231,101.014,"LINESTRING (-24.5512982195195 -49.491749471179,-24.6502842880659 -49.509952775
Trough,Fluvial,Channel axis,Partly Buried,Narrow,125799.281577,121.662,"LINESTRING (-28.6802740975685 -51.1356134666679,-28.6820996982245 -51.13673611
Trough,Fluvial,Channel axis,Partly Buried,Narrow,196974.606344,211.708,"LINESTRING (-39.7436395600759 -43.7689184381635,-39.753171783961 -43.73194171
Trough,Fluvial,Channel axis,Partly Buried,Narrow,107256.315814,116.892,"LINESTRING (-36.3507450626342 -43.2903464701999,-36.326376364508 6 -43.26368534
Trough,Fluvial,Channel axis,Partly Buried,Broad,314908.29232,289.641,"LINESTRING (-49.1586617953567 -44.8573845218351,-49.193083596944 2 -44.8470784324
Trough,Fluvial,Channel axis,Partly Buried,Broad,295557.786332,275.824,"LINESTRING (-48.7783858769289 -42.6959477012563,-48.7636702630602 -42.678625321
Trough,Fluvial,Channel axis,Partly Buried,Narrow,319622.312596,308.219,"LINESTRING (-54.2009269433862 -45.5108854988259,-54.212778002978 -45.505474069
Trough,Fluvial,Channel axis,Partly Buried,Narrow,169777.280587,148.188,"LINESTRING (-53.9467734145435 -46.9516806161738,-53.9862075615103 -46.91501095
Trough,Fluvial,Channel axis,Partly Buried,Broad,128538.421142,119.581,"LINESTRING (-54.1995391267674 -47.3389227005027,-54.197723468151 -47.368108535
Trough,Fluvial,Channel axis,Partly Buried,Narrow,345999.250305,333.48,"LINESTRING (-62.5078123815053 -52.8397671113817,-62.3480010209421 -52.730566097
```

- Vector support
  1. If PDS4 table has Latitude / Longitude field (acts as point)
  2. If PDS4 table has Well-Known-Text (WKT) field then supports points, lines, polygons

Wait - what…

- PDS4 supports Well-Known-Text (WKT) ?
    1. Not yet approved
    2. Need feedback

    Why WKT
    - ISO standard
    - Simple but allows for multiple shapes and holes
    - Broad use across applications

Please email if you have specific requirements for next contract.

thare@usgs.gov, Trent Hare