

# PDS4 Viewer and Python Library

Lev Nagdimunov

NASA PDS – Small Bodies Node  
University of Maryland, College Park

# PDS4 Viewer

- A read-in and visualization tool for PDS4 data
- Available stand-alone on Windows, Mac and Linux
- Supports all\* PDS4 data (images, spectra, arrays, tables)

Able to display 2-D and 3-D images and spectra, allowing for zoom, rotation, pan, axis-inversion, colormap manipulation, scaling, etc.

Able to display PDS4 character, delimited and binary data tables, including arbitrarily nested GROUP fields (sub-tables).

Allows plotting PDS4 table columns against each other, including error bars for uncertainties.

Consistent interface allows for easy access to meta-data (such as object labels, display settings and spectral characteristics, etc.)

PDS4 Viewer Demo ...

# Python PDS4 Tools

[\*\*http://tinyurl.com/pds4-viewer\*\*](http://tinyurl.com/pds4-viewer)

[\*\*http://tinyurl.com/pds4-reader\*\*](http://tinyurl.com/pds4-reader)

[\*\*http://tinyurl.com/pds4-data\*\*](http://tinyurl.com/pds4-data)

# Python PDS4 Tools

- Written with both scientists and software developers in mind
- Supports all\* PDS4 observational data (images, spectra, arrays, tables)
- Provides both data and label meta-data in convenient formats

# Python PDS4 Tools

- Compatible with Python 2.6+ and 3.3+
- Has only one dependency (NumPy)
- Completely cross-platform

```
>>> from pds4_tools import pds4_read
```

```
>>> structures = pds4_read('/path/to/label.xml')
```

```
>>> structures.info()
```

0	Array_3D_Spectrum	'data_Primary'	3 axes, 21 x 10 x 36
1	Table_Binary	'data_Integration'	9 fields x 21 records
2	Table_Binary	'data_Engineering'	38 fields x 1 records
3	Table_Binary	'data_Binning'	9 fields x 1 records
4	Table_Binary	'data_PixelGeometry'	12 fields x 21 records
5	Table_Binary	'data_SpacecraftGeometry'	36 fields x 21 records
6	Table_Binary	'data_Observation'	22 fields x 1 records

```
>>> label = structures.label
```

```
>>> mission_area = label.find('./Mission_Area')
```

```
>>> print( mission_area.findtext('./mvn:filter') )  
        'HB UV Continuum'
```

```
>>> label.to_dict()
```

```
>>> label.to_string()
```

```
>>> label.getroot()
```

```
>>> array = structures[0]
```

```
>>> label = array.label
```

```
>>> data = array.data
```

```
>>> meta_data = array.meta_data
```

```
>>> data = array.data
```

```
>>> print( data.min(), data.max() )
```

```
1023.12, 92313.3
```

```
>>> section = array.section[0:50000, 50000:100000]
```

```
>>> print( section.min(), section.max() )
```

```
502.9, 608.2
```

```
>>> from pds4_tools import pds4_read
```

```
>>> structures = pds4_read('/path/to/label.xml')
```

```
>>> structures.info()
```

0	Array_3D_Spectrum	'data_Primary'	3 axes, 21 x 10 x 36
1	Table_Binary	'data_Integration'	9 fields x 21 records
2	Table_Binary	'data_Engineering'	38 fields x 1 records
3	Table_Binary	'data_Binning'	9 fields x 1 records
4	Table_Binary	'data_PixelGeometry'	12 fields x 21 records
5	Table_Binary	'data_SpacecraftGeometry'	36 fields x 21 records
6	Table_Binary	'data_Observation'	22 fields x 1 records

```
>>> table = structures['Data_Integration']
```

```
>>> label = table.label
```

```
>>> data = table.data
```

```
>>> meta_data = table.meta_data
```

```
>>> table = structures['Data_Integration']
```

```
>>> field = table['field_name']
```

```
>>> field = table.field('field_name')
```

```
>>> field = table.field(0)
```

```
>>> print( field )
```

```
array([486390954.2, 486390343.6, 486390864.3, ... 474321864.8, 47432172.5])
```

```
>>> records = table[0:100]
```

```
>>> print( records )
```

```
array([(486390954.2, 486391050, '2015/152 Jun 01 00:36:23.03709UTC', 24932,  
32.93, 65.87, 1, 15960, 2602),  
( ... )])
```

```
>>> table.data
```

```
>>> table = structures['Data_Integration']
```

```
>>> field = table.field(...)
```

```
>>> print( field.meta_data.keys() )
```

```
['name', 'location', 'data_type', 'length', 'unit', 'description']
```

```
>>> print( field.meta_data['unit'] )
```

```
'Microns'
```

```
>>> print( field.meta_data['description'] )
```

```
'Wavelength of ... '
```

# Additional Features

- Masked data

```
>>> structure = structure.as_masked()
```

- Lazy Load

```
>>> structures = pds4_read(..., lazy_load=True)
```

**THANK YOU FOR YOUR ATTENTION!**

**<http://tinyurl.com/pds4-viewer>**

**<http://tinyurl.com/pds4-reader>**