

## ATMOSPHERES DATA: GALILEO, MARINER, JUNO MIGRATION

L.F. Huber, T. Güth, J. Emmett, N. Chanover, L.D.V. Neakrase

Planetary Data System, Atmospheres Node, Dept. of Astronomy, P.O. Box 30001, MSC 4500, New Mexico State University, Las Cruces, NM 88003-8001

**Introduction.** The PDS Atmospheres Node is continuing the process of migrating its archived data holdings from the PDS3 to PDS4 archiving standard. The goal of this work is to bring all data holdings up to the PDS4 standard, allowing for better integration across all investigations. Here we discuss our recent migration work that encompasses the *Juno*, *Galileo* and *Mariner* missions. Our migration of the Cassini/CIRS data is part of a larger, multi-node Cassini migration effort that is beyond the scope of this presentation.

**PDS3-PDS4 and Python.** The Atmospheres Node has developed a collection of Python methods and scripts to make the migration process as automatic as possible, even when working with the more complex labels, such as those used by several of the *Juno* instruments. The Python codes handle reading the valid PDS3 labels (.lbl) using translation files to convert to equivalent PDS4 attributes. The resultant PDS4 output creates .xml labels for all the pieces of the bundle utilizing the various parts of the PDS3 structures. The archive bundles are created all at once through the code and PDS4 archive bundles are built with minimal human effort. Resultant bundles are then validated against the PDS4 standard and released alongside the certified PDS3 versions of the same data.

Initial testing of the code was done on *Phoenix Lander* atmospheric data in 2016 and has evolved over time alongside the PDS4 Information Model (IM) as refinements are released. Current migration codes convert PDS3 to PDS4 up to IM Version 1.15.0.0 (released December 2020).

With more complex datasets and the inclusion of the larger mission data, there has been a need to utilize local data dictionaries (LDDs). Local data dictionaries may incorporate mission attributes in a mission dictionary or discipline specific attributes (e.g., display, geometry, cartography, spectroscopy, etc.) that are defined outside the core PDS4 Information Model. Dictionary support within PDS4 labels is found in the Discipline\_Area section of any label and the core migration codes have needed to include local data dictionaries to complete migration of several datasets.

**Features and Capabilities.** Several different Python scripts and other supporting files work together to

properly execute a migration. A blank XML template that corresponds to the overall structure of the PDS3 labels is the starting point. For datasets containing Local Data Dictionaries, the XML declaration at the top of this template (the preamble) must be modified to include the appropriate targetNamespace and schemaLocation for those dictionaries in order for those tags to show up in the template. A keyword lookup file (or character definition dictionary) for translating keywords from PDS3 to PDS4 is used to fill in many of the repeated tags of the label, the LID's and/or LIDVID's, and points to the appropriate tags used within the LDDs. A compilation of Python methods (PDS\_Module) capable of handling different tables, file types, and tag-filling methods is continuously being updated/modified and maintained by our staff of undergraduate research assistants. One main Python script is generally used per dataset (and adapted for each dataset) to read through the PDS3 label, call the respective functions from the Module, and output the corresponding PDS4 .xml files.

The amount of human effort that goes into preparing a dataset for migration primarily depends on the datasets themselves. The more complex datasets (e.g., FITS files, files containing SPICE kernels, or datasets with multiple instruments) require more effort as the structure of these files is rather complicated and the methods used to handle these are more complex. As more migrations are done, this process becomes more intuitive. Node experience and continued refinement of the Python routines reduces the severity of errors and acts as an iterative process that in turn provides further examples for the base migration code. Files other than data files (e.g., documents) take more time to complete as they frequently require manual processing.

**Current Efforts.** The Atmospheres Node has completed the process of migrating all of its *Juno* holdings (Gravity, MWR, UVS and JIRAM instruments) to PDS4 and currently migrates each new delivery so that the PDS4 files can be released alongside the delivered PDS3 data.

Older data sets that have been completed or are undergoing migration efforts currently include the *Galileo Orbiter* UVS and PPR instruments, *Mariner 6/7/9* UVS instruments and all of the *Galileo Probe* data. At the time of writing, the *Galileo Orbiter* UVS

and PPR and the *Mariner 6/7* UVS were nearly complete, while the others were at about 50% completion. The *Galileo* UVS has provided some challenges due to varying formats between the data files and the geometry files.

**Expandability and Future Plans.** The Atmospheres Node plans to completely migrate all of its holdings in the next 5 years. The next target for migrations will be the active Mars missions *Mars Reconnaissance Orbiter* and *Mars Science Laboratory*. The migration code for these continuing missions as well as *Juno* will need to be maintained until those missions have ceased science operations.

As the PDS4 Information Model continues to mature, it is likely that we will “re-migrate” our holdings that were migrated very early on in order to make better use of improvements to the IM. Some of the more complex data sets will be reserved until the end of our efforts.