MIGRATING THE CASSINI RADAR ARCHIVE TO PDS4. P. E. Geissler, U.S. Geological Survey, 2255 N. Gemini Drive, Flagstaff, AZ, USA (pgeissler@usgs.gov).

Introduction: Since 2011, NASA has required that new data submitted to the Planetary Data System (PDS) for inclusion in their archive be in the modern PDS4 format [1-3]. This has necessitated both migration of data formatted in the older PDS3 format and establishment of new methods and tools for creating data archives that are fully compliant with PDS4 formats [4-5]. The PDS provides a number of tools to support users in archive migration and creating and working with archived data in PDS4 formats (see https://pds.nasa.gov/tools/about/).

Here we describe the steps taken to migrate the Cassini RADAR archive to PDS4 compliance while preserving the PDS3 archive intact so that it can continue to be used as it has been in the past.

Archive Design and Approach: Cassini RADAR data are organized into separate volumes corresponding to distinct close flybys of Saturn's satellites. Each flyby was planned differently and documented as if it were a distinct mission. Therefore, it made sense to keep this organization and create PDS4 collections corresponding to each PDS3 volume. The structures of these collections are similar to one another, so only a single bundle was needed.

Cassini used a year-day_of_year system to designate start and stop times which must be converted to year-month-day for PDS4. We found that the python `time` function could do the task and decided to do the entire migration using python scripts.

We made a collection of sample data consisting of a handful of PDS3 volumes representative of the rest of the archive to test our scripts and validation.

We decided to migrate everything in the PDS3 archive, including data products, documentation, index tables and EXTRAS.

We altered the CART dictionary to include the Oblique Cylindrical map projection used by the BIDR images.

Generating PDS4 Labels: We used the PDS4 Generate tool (now known as MILabel) to produce PDS4 labels from PDS3 label input. The steps involved are detailed in [6]. First we used Oxygen XML Editor to write prototype labels for each of our 53 distinct file types, including all the information available from the PDS3 labels along with relevant citation and context information. After verifying that the labels were valid and correctly represented the data products in the PDS4 Viewer, we converted them to generalized label templates by replacing the product-specific information with Apache Velocity variables and expressions [6]. To test our templates, we used the PDS4 Generate tool together with our templates to re-create our prototype labels.

Next we wrote 53 python scripts to hunt down each file type in our test archive, match them with the correct template, and generate PDS4 labels. For data products with PDS3 labels, the scripts ran the Generate tool to extract specific information from the PDS3 labels for use in the corresponding PDS4 labels. For ancillary files without PDS3 labels, we made generic templates and used python to replace variables such as `$file_size` with the actual file size.

Generating Inventories, Collections, and Bundle: Inventories were produced by walking through each volume in our test archive, reading each xml file, and writing the urn to a collection_inventory_#volume file preceded by status and anteceded by the version number as determined from the file name.

Collection labels were next generated using a generic collection label template and replacing variables with the correct urn, inventory file name, number of inventory records, collection type, md5 checksum and current date, all computed with python.

The bundle was also produced from a generic template, with the bundle member entries appended by walking through the volumes in our test archive.

PDS Review: The Cassini RADAR data had already been through a science review, but needed another pair of eyes on the PDS4 implementation. The PDS Atmospheres Node provided a careful review. Their most important suggestion was to further subdivide our collections so that data, documents, and ancillary files were in separate collections. They also pointed out that certain strictly PDS3 files such as FMT files should not be migrated at all.

We made the changes suggested, reviewed the revisions and were set to migrate the entire archive.

Audit: Prior to the migration, a complete audit of the Cassini RADAR archive was carried out by the PDS Cartography and Imaging Node. They compared our holdings to those at JPL, eliminated superseded volumes, and added or updated volumes that had fallen behind to be sure that we were migrating all the data that should be migrated and no data that should not. The full audited archive was then copied to a scratch disk for migration and validation.

Migration and Validation: Migration simply consisted of running the scripts on the full archive, capturing the output into text files, and checking to see that everything completed correctly.

Validation was trickier. Many of the data products are stored as compressed .ZIP files for quicker download and had to be inflated for product level validation. Even with a relatively small archive of ~14,000 items, the PDS Validate tool took about 36 hours to complete. The Validate tool was run with the PDS4 Bundle rules applied to enforce referential integrity checking.

Errata and Oddballs: Validation of the complete archive found several errors among data products that were not included in our sample data set. For example, the length of the first field of INDEX.TAB changed twice during the course of the mission so a script was written to make them consistent. Many CSV files used '-Inf' as a marker for bad data, so a script was written to replace those markers with '-9999' to be PDS4 compliant. Two ASCII tables had a few obviously wonky values that were replaced with '-9999' by hand-editing.

Other errors could be fixed by modifying the PDS4 label rather than the data product. A couple of binary tables had PDS3 labels that incorrectly reported the number of records in the table. One binary image could only be accommodated by including <error_constant>NaN</error_constant> among the <Special_Constants>.

We also altered the archive deliberately, for example replacing the essential Software Interface Specifications with PDF/A versions to be PDS4 compliant. We replaced each volume's CUMINDEX.TAB with the final mission CUMINDEX.TAB for utility and consistency.

All these modifications are recorded in the text file PDS4_errata.txt to facilitate synching our altered archive with the archive currently offered online.

Last Steps: Now that we have a PDS4 archive that passes validation, we have two more steps to complete in order to pass the finish line. The first is to obtain a DOI number from the PDS Engineering Node (EN) and include it in our bundle.xml. Second is to have the archive harvested into the PDS4 database, a task that will also be undertaken by the EN.

Acknowledgments: This task could not have been accomplished without the help and advice of Lyle Huber and Nancy Chanover at the PDS Atmospheres Node, Lisa Gaddis, Trent Hare, David Mayer, John Blalock, Glen Cushing and Melody Hartke at the PDS Cartography and Imaging Node, and Jordan Padams, Richard Chen, Cristina de Cesare and Emily Law at JPL.

References: : [1] Crichton et al. (2011) *EPSC Abstracts*, 6, #1733. [2] Beebe et al. (2010) AAS-DPS meeting #42, id.37.02; Bulletin of the American Astronomical Society, Vol. 42, p.967. [3] Hughes et al. (2018) Planetary & Space Science 150, pp. 43-49. [4] The PDS4 Data Provider's Handbook, (https://pds.nasa.gov/datastandards/documents/dph/current/PDS4_DataProvidersHandbook_1.11.0.pdf). [5] Planetary Data System Standards Reference, 1.9.0 (https://pds.nasa.gov/pds4/doc/sr/current/). [6] Geissler (2019) 4th Planetary Data Workshop, abstract 7103.