Worksheet for Build-A-Bundle Exercise

PDS4 Training Session, DPS Fall 2017 Meeting, Provo, UT, October 15-20, 2017

Learning Objectives: By the end of this exercise you should be familiar with the organization of a PDS4 Bundle including member Collections with their respective contents (inventories, basic products, etc.)

For today's exercise we are going to use some real data. Specifically, we will be using the *Mars Science Laboratory Entry, Descent, and Landing Atmospheric Reconstruction* data that was recently reviewed by the PDS Atmospheres Node.

The "Science_Files" folder within the "2_Bundle_Exercise" directory contains your base science files for this exercise. There should be three files listed here.

readme.txt
BU_PDS_EDLdata.txt
reconstruction.pdf

Your task is to build a PDS4 Bundle for the data presented in the Science Files folder.

You will need to use templates to complete your work. *The Template Package directory (3_Template_Package) contains PDS4 XML Templates to begin to construct your PDS4 Archive Bundle.*

These are templates for the labels you will make for the archive.
This directory should include **4** template files:

bundle_template_1800.xml
collection_template_1800.xml
document_template_1800.xml
table_delimited_template_1800.xml

All work you do should be saved in a common location either on your computer or in the "User_Exercise" space provided on the thumb drive in the "2_Bundle_Exercise" directory.

Before we start assembling the bundle (and using XML) you need to think about the organization of the collections. In order to do this we need to designate the order of the logical identifiers.

STEP 1: *What will you call this bundle?*

We need to determine what the bundle id will be to begin building the URN. *Remember URNs must be in all lowercase with no spaces (dots and underscores are legal separators)*

In order to help provide a unique id, we suggest an ordered approach.

We try not to include researchers names in the bundle id. For clarity we try to include information about mission, instrument, and type of science to round out the Bundle ID and create a unique identifier. Individual nodes may have specific rules for this so it is always a good idea to be in regular communication with your node representative.

Examples:

Mars Pathfinder Atmospheric Opacity Data	urn:nasa:pds:mpf_opacity
Voyager Calibrated IRIS Data	urn:nasa:pds:vgr_iris_calibrated
Ground-based IRTF observations of Jupiter	urn:nasa:pds:irtf_jupiter

Bundle LIDs (URNs) have this form:
urn:nasa:pds:<bundle_id>

For your reference enter your Bundle LID here:
urn:nasa:pds:__________________

<u>STEP 2:</u> ***What types of collections will you have in the bundle?***

Look at the data. The data are located in “2_Bundle_Exercise/Science_Files”

Ask yourself…
What type of data do you have? ______________________
In what format are the data? ______________________
Is there more than one processing level? ______________________
Is there documentation? ______________________
How many collections do we need? ______________________

Collections are typically subdivided by purpose or processing level in most cases. *Remember, Collection LIDs are a combination of the Bundle LID + an extra segment used as the Collection ID.*

In simple bundles Collection IDs tend to be uniform and limited to something like: ***document, data_raw, data_derived, context, etc.***

Examples:
LADEE UVS Document Collection

urn:nasa:pds:ladee_uvs:document
MAVEN IUVS Corona Scan Data Collection
urn:nasa:pds:maven.iuvs.raw:corona
Phoenix MET Raw Data Collection
urn:nasa:pds:phx_met:data_raw

For our example data, we have a data file, and a document. In the documentation we can see references to mission/instrument/target, namely Mars Science Laboratory, the Accelerometer instrument, with the target, Mars.

So from above, we should have two collections: a Document Collection and a Data Collection.

Collection LIDs (URNs) have this form:
urn:nasa:pds:<bundle_id>:<collection_id>

For your reference enter your Collection LID(s) here:
urn:nasa:pds:_______________:________________
urn:nasa:pds:_______________:________________
**urn:nasa:pds:_______________:________________
**urn:nasa:pds:_______________:________________

***Optional:**
Your PDS Node might also require a Context Collection and an XML_Schema Collection – these collections will contain only secondary products that are managed by the PDS Engineering Node. These two collections may be included for completeness and as a quick check for the node to see which mission/instrument/targets you have in your bundle or which versions of the PDS4 Information Model you use throughout your bundle. Your node representative will be able to advise you on how to set up these two collections if that node requires them. We have provided them in the completed example for you to look at later (4_Completed_Example).

STEP 3: *Begin working with the XML template files.*

Now that we have a plan for the referencing LIDs, let's begin to set up the different pieces of the bundle.

Open the directory "3_Template_Package" on the thumb drive.
This directory should include a group of tailored XML templates like you might receive from your node representative or from node-specific tools.

For this exercise this is the starting point for building your bundle.

You will need a bundle file, a collection file for ***each*** collection, and the appropriate product file for the tables and documents. You can open these in your favorite text editor to edit them. We suggest that you will probably need multiple files open at the same time.

We will start with the Bundle and Collection Files before moving on to the data and document files.

Bundle File
We can start to populate templates with LIDs (URNs) you constructed above.

Open the bundle_template_1800.xml file from the Template Package.

The <logical_identifier> field can be found at the top of each label in the <Identification_Area>.

REMEMBER:
Bundle LID: *urn:nasa:pds:<bundle_id>*
Collection LID: *urn:nasa:pds:<bundle_id>:<collection_id>*

Put the Bundle LID in the <logical_identifier> tag (the first line of the <Identification_Area>.

When you save the edited versions you will want to “Save As” and change the filenames in a bundle directory on your computer (or use the provided space on the thumb drive, “2_Bundle_Exercise/User_Exercise/”):
bundle_<bundle_id>.xml
collection_<bundle_id>_<collection_id>.xml

For Bundle labels there are a few more fields at the bottom of the label you can fill out at this time. Look for the section beginning with the <Bundle_Member_Entry> tag about 2/3 of the way to the bottom. This section designates all the member collections for this bundle.

From Step 2 above, you should be able to include your collections in the <Bundle_Member_Entry> section. You also need relationship information, which has already been provided in this section. You should be able to match your LIDs with the appropriate collection reference types.

<lid_reference> should be your collection LIDs from above

<member_status> should be “Primary” as this is the first time these collections have been put together for the archive.

<reference_type> should be “bundle_has_<collection type>_collection”
Examples: bundle_has_data_collection, bundle_has_context_collection, bundle_has_document_collection

We have included room for 4 collections. The first 2 lines should correspond to the collections you decided on from Step 2. The other 2 are the optional Context and XML_Schema Collections.

Optional Reading:
Context Products – Missions, Spacecraft, Instruments, Targets
In each label file, there is a section devoted to the inclusion of context product references. Remember, the LIDs for these products aid in providing basic information for search and retrieval and link mission/instrument/target information to the Bundle/Collection/Product structure in PDS4. All context products are managed by the PDS Engineering Node (EN) and must be part of their repository to be legal values in your labels.

Missions, their spacecraft and instruments, and their targets all have context products that typically get made during the archiving of their data.
Other classes of investigations and facilities must be made at the time of the archiving and will typically be constructed by your node representative and registered with EN when complete if they don’t already exist within the EN repository.

For this exercise we have provided the correct Context LIDs for your <Context_Area>. The <Context_Area> can be found just above the <Bundle_Member_Entry> section you edited in the first part of Step 3. These references tie your data to the selected missions, spacecraft, instruments, and target(s).

The Context Area consists of:
<Time_Coordinates> is a listing of start/stop times for the bundle.

<Investigation_Area> is used to describe how the data were collected by use of context products. <Internal_Reference> provides a linkage to the mission.

<Observing_System> provides links to the instrument host (spacecraft) and the instruments used in these data products.

<Target_Identification> lists all relevant targets in these data.

Fill in the <Context_Area> (blanks designate needed values). Use the following values where appropriate in the <Internal_Reference> tags:

Investigation: Mars Science Laboratory
urn:nasa:pds:context:investigation:mission.mars_science_laboratory

Observing System
Accelerometer
urn:nasa:pds:context:instrument:instrument.accelerometer.msl
Spacecraft
urn:nasa:pds:context:instrument_host:spacecraft.msl

Target Identification
Mars
urn:nasa:pds:context:target:planet.mars

*****This information in the context area can be propagated through all the labels as it will be applicable to every part of our Bundle example.***

This material can be added to your individual bundle and collections files if you choose to do it. We have left cues to where each of the context products belong within the Context Area of the labels.

Document Referencing

Because PDS is committed to providing usable data to the public, we ***require*** you as the data provider to include relevant documentation on how your submitted data be used and how they were generated. To increase the likelihood of users finding your documentation we have cross-referencing in the labels to ensure this.

The ***<Reference_List>*** section should be used for this. You are not limited in the number of files that could appear in this section, but most often this will be a handful of important informative documents including spacecraft and instrument references, Software Interface Specifications (SISs) or User’s Guides, and/or published refereed manuscripts. Documents are acceptable in many formats, however text documents are most common in either plain ASCII text or PDF/A files. (As a minimum requirement - one ASCII or PDF/A file must be included regardless of how many other formats may be included.)

For our example, we have the “reconstruction.pdf” file that is appropriate for these data. This file will reside in the ***Document Collection*** from the bundle we are creating and can be put in the <lid_reference> field of the ***<Reference_List>*** section. The <Reference_List> section immediately follows the <Context_Area>.

This basic product LID will have the form:
urn:nasa:pds:<bundle_id>:document:reconstruction

*****Again, this <Reference_List> section will remain the same for all products in the bundle, because we want to reference the User Guide document in every file. So the information here should be propagated through all the other files.***

Start with filling out the bundle and collection files for now.

STEP 4: ***Collections***

Now continue with editing the collection files. The first step is to remind ourselves of how many collections we will be working on. For today’s example we have two main collections to work on: the ***Data Collection*** and the ***Document Collection***.

Having two collections, we need two collection files.

We want to populate two copies of this file with the appropriate information from the Data and Document collections.

Open the collection_template_1800.xml file in the “3_Template_Package” directory.

Save 2 copies of this template.
These two files should be named like:
collection_<bundle_id>_document.xml
collection_<bundle_id>_data.xml

Just like with the Bundle file above, you can begin by inputting the collection LID into the <logical_identifier> tag at the top of the <Identification_Area>.

<title> should be used to title your collection. Something like, “Document Collection for the MSL EDL Bundle” should be used, using Title Case.

<product_class> for collections should be “Product_Collection”.

<Citation_Information> can be used to put the people responsible for creating these files or the archive bundle with some free-form fields for descriptions and keywords.

Optional for completeness:
Following that, moving down the label you should be able to use the information from the Bundle file to populate the <Context_Area> and <Target_Identification> sections. The <Internal_References> provide the LIDs (URNs) for your context products (see Context Products above for the values). *Hint: the <reference_types> provided will give you clues as to values you should put in the spaces above them.

After completing this we move on new types of fields for collection files.
First, we fill out the <Collection> section. This consists of a single field that designates what type of collection we have.

In <Collection>, find the <collection_type> tag.

Fill in the <collection_type> for each collection.
In our example bundle today, we should use the values “Document” and “Data” respectively.

The last part of the collection file is the <File_Area_Inventory> section. This section links the collection with its inventory file. The blanks listed in the <File> section provide the name of the file where the inventory is listed and a local identifier, and the date of its creation, for standard bookkeeping.

Fill in the <File> section of the File Area Inventory.
Remember Collection Inventory files should have names like:
collection_<bundle_id>_<collection_id>_inventory.txt
(with possible formats being .csv, .txt, or .tab)

The <local_identifier> in this section should be the filename of the inventory file without the extension.

The rest of the text below this section in the <Inventory> section designates the layout of the inventory table as a 2-column table listing member status and the LID::VID combination for the system to find every file in the collection. The field <records> here should have a number that matches the number of lines in the inventory file. In our example, our collections only contain 1 file each so the number of records in each collection is “1”. The rest of the fields in the <Record_Delimited> section here should never be changed as these denote the standard format of the Inventory File.

Now, it’s time to step through our collections to make sure everything is correctly populated and that the collections are now complete.

<u>STEP 5:</u> *Document Collection and Document Files*

Before we move on to the data files, the first collection we will look at will be the Document Collection. We’ve already decided (above) that the example project has one document file (*reconstruction.pdf*). Now we have to make sure we have the document collection built around that.

In any complete collection, the two required files are the collection label (.xml) and the inventory table (.csv/.txt/.tab). If all products are “Primary” members (i.e., they are being submitted to the PDS for the first time here) the inventory will likely list the files in this directory.

So for a typical simple document collection with one document file you should have:

collection_<bundle_id>_document.xml
collection_<bundle_id>_document_inventory.txt
<document_filename>.xml (reconstruction.xml)
<document_filename>.pdf (reconstruction.pdf)

<u>Document File Labels</u>

From Step 4 we have already set up the collection label for the document collection. We still need to make the inventory file and the document label. A general practice is to make the inventory file last, because we will need the LIDs (URNs) for each product in the collection in order to populate the list in the inventory. So first we will make the label for the document file.

Open the document template file (document_template_1800.xml).
This file should look similar to what we saw in the bundle and collection labels. We can start at the top and fill in the <logical_identifier> by appending the filename (without extension) as the last piece of the LID.

urn:nasa:pds:<bundle_id>:<collection_id>:<filename>

Similar to what we did above:

<title> should be filled in with the title of the document using title case.
<product_class> should be "Product_Document" for document files.

<Citation_Information> contains information about the authors, publication year, with freeform fields for keywords and descriptions of the file.

<Modification_History> is a place to record when this file gets updated complete with version updates and a description field for describing the changes.

Save this file as "<document_filename>.xml", filling in the name of your file. This XML file is now the label for your document.

<Reference_List> in document files can be used to reference a context product for the instrument that took the data we are archiving (MSL Accelerometer) and a published paper (with DOI). The best archiving practices should include as much metadata as possible to describe each file. Many times this is up to you and the node you are working with to decide how much is necessary. The MSL Accelerometer reference is an "Internal Reference" because this instrument has a reference within PDS already through the EN Context Products. The journal paper reference is an "External Reference" because it references something that is outside of PDS and won't be registered within PDS. Best practices for external references are to include as much metadata as possible to ensure users can easily find the reference outside of PDS.

Complete the <Document> section of the label.

<Document> is the place where all the specific metadata about the file goes. Document name, authors, publication date, document editions (multiple formats/languages could be other editions), file format information, DOI info, etc. all have places here. Some of the pertinent information has already been filled out for convenience in this exercise.

<document_name> is essentially the title of the document in Title Case.
Author information should be listed beneath this in the <author_list>.

<Document_File> is used to link up the various file names of various editions of the document. In our case we have 1 edition, therefore, 1 file name (reconstruction.pdf) – As before the <local_identifier> here is just the file name without the extension. (i.e., the filename_id portion of the LID (URN) at the top of the label.)

Document Collection Inventory File

Once the document files are labeled (in our exercise – one document) we have the LID necessary to build the *Collection Inventory File* for this collection.

The collection inventory is a listing of all products considered part of this collection. The inventory consists of a 2-column, comma-separated list. The first column is a designator designed to alert the registry system of whether or not the listed product is new to the PDS4 archive or could be found in some other preregistered place.

> ***"P"*** *(Primary)* designates that the product is being registered for the first time, and is present in this collection.
>
> ***"S"*** *(Secondary)* designates that the product has been already registered and may or may not be physically present in this collection.

The second column designates the ***LID::VID*** of each included product. The LID::VID is a combination of the logical identifier (LID) and the version identifier (VID) separated by a double colon.

So for our example in the Document Collection, the inventory file should consist of a 2-column table with only one entry – the entry for the PDF/A file, reconstruction.pdf.

Therefore the document collection inventory should look like:

P, urn:nasa:pds:msledl:document:reconstruction::1.0

This shows the registration system that the associated file is being registered as for the first time and is Version 1.0.

Make a new text file with the above format.
Save this file as (in the same place as the collection label (.xml):
collection_<bundle_id>_document_inventory.txt

Consequently, if this file were to be updated at some point the registry could be re-run for a new version of this product, where everything would be the same except for an incremented VID.

STEP 6: *Data Collection and Data Files*

Collection File

We have already discussed how to set up the collection label file in Step 4 (for both collections) and completed making the Document Collection in Step 5. The Data Collection is essentially the same process as we did in Step 5.

In any complete collection, as before, the two required files are the collection label (.xml) and the inventory table (.csv/.txt/.tab). If all products are "Primary" members (i.e., they are being submitted to the PDS for the first time here) the inventory will likely list the files in this directory.

So for a typical simple data collection with one document file you should have:

collection_<bundle_id>_data.xml
collection_<bundle_id>_data_inventory.txt
<data_filename>.xml (BU_PDS_EDLdata.xml)
<data_filename>.<ext> (BU_PDS_EDLdata.txt)

Data File Label

As was the case in the Document collection we should start with the data file(s). In our example today we have one data file (BU_PDS_EDL.txt) that needs a label. This is a delimited table, so we will use the template:

table_delimited_template_1800.xml.

Open the file, table_delimited_template_1800.xml.

This should look familiar, as it should resemble the document label we just completed. The LID can be completed at the top of the label – remembering that the construction should look like:

urn:nasa:pds:<bundle_id>:<collection_id>:<filename>.

The filename portion of this URN should be the filename of the data file in all lowercase, no spaces or extensions. (e.g., BU_PDS_EDLdata becomes bu_pds_edldata)

As before, **<title>** should be filled out with title for the data product in Title Case. This could be as simple as an expanded version of the file name or just "Data File" – something that serves as a title for your file.

<product_class> at the top should be listed as "Product_Observational" to denote that this is a data file.

Save this file as "<filename>.xml", using the data file's name. This XML file is now the label for your data product.

Next, the <Observational_Area> can be filled out. For our simple example, we've already filled in some of this information in the effort to save some time.

<Time_Coordinates> provides a place to put start and stop times for the observation.

<Primary_Results_Summary> includes keyword references to help with search capabilities after these data are registered. This includes keywords for purpose, processing level, and a section to list Science Facets. In our exercise today we have left these as:

```
<Primary_Result_Summary>
  <purpose>Science</purpose>
  <processing_level>Derived</processing_level>
       <Science_Facets>
    <domain>Atmosphere</domain>
    <discipline_name>Atmospheres</discipline_name>
         <facet1>Structure</facet1>
       </Science_Facets>
</Primary_Result_Summary>
```

There are enumerated lists for most of these values to help narrow down selections – your node representative can help you decide the best fit for your data.

Optional for completeness:
<Investigation_Area> contains information that should be copied from one of the other files to maintain all the proper information relating your data back to the mission/instrument/target that we set up earlier. These are the same context products we've referenced in each file.

<name> should be the related mission name (e.g., "Mars Science Laboratory")

In each case we need the EN context reference LID (URN) for each blank in these sections. (see optional section about context products from Step 3)

<Observing_System> contains the references to the instrument and spacecraft.

<Target_Identification> contains the reference to the target body. (e.g., Mars)

Because PDS is committed to providing good documentation in support of data in the archive, and you are required as data providers to include good documentation

<Reference_List> should be filled out with the URN link to the document file we used in the document collection. The LID reference should be the one for the reconstruction.pdf file and the ***<reference_type>*** should be “data_to_document”.

Next we need to fill out the **<File_Area_Observational>** section. This section provides the necessary metadata to describe the data file. In the case of a table file this provides filename information, creation time, file size in the ***<File>*** section, and all of the column header information in the ***<Table_Delimited>*** section. Again, we’ve left most of this information as it should be for your convenience and as an example of some of the ways to denote fields and units.

Data Collection Inventory File

Now that the data file label is complete we have all the relevant information to construct the collection inventory file just as we did with the Document Collection above.

Just as before in the Document Collection:
The collection inventory is a listing of all products considered part of this collection. The inventory consists of a 2-column, comma-separated list. The first column is a designator designed to alert the registry system of whether or not the listed product is new to the PDS4 archive or could be found in some other preregistered place.

> ***“P”*** *(Primary)* designates that the product is being registered for the first time, and is present in this collection.
>
> ***“S”*** *(Secondary)* designates that the product has been already registered and may or may not be physically present in this collection.

The second column designates the ***LID::VID*** of each included product. The LID::VID is a combination of the logical identifier (LID) and the version identifier (VID) separated by a double colon.

So for our example in the Data Collection, the inventory file should consist of a 2-column table with only one entry – the entry for the ASCII text file, BU_PDS_EDL.txt.

Therefore the data collection inventory should look like:

P, urn:nasa:pds:msledl:data:bu_pds_edldata::1.0

This shows the registration system that the associated file is being registered as for the first time and is Version 1.0.

Make a new text file with the above format.
Save this file as (in the same place as the collection label (.xml):

collection_<bundle_id>_data_inventory.txt

Consequently, again, if this file were to be updated at some point the registry could be re-run for a new version of this product, where everything would be the same except for an incremented VID.

STEP 7: Review the Completed Bundle

Now that we've walked through the entire bundle for our example we should take a step back and review the final bundle layout so you can double check your progress.
PDS4 Bundles should have a familiar directory structure but it helps if label files show up in predictable places. So we have included a checklist breakdown of the bundle we just completed for your convenience. We present it here with generalized names because our choices throughout may not have been the same.

SCIENCE BUNDLE
bundle.<bundle_id>.xml

DOCUMENT COLLECTION
collection.<bundle_id>_document.xml
collection.<bundle_id>_document_inventory.txt (.csv)
reconstruction.xml
reconstruction.pdf

DATA COLLECTION
collection.<bundle_id>_data.xml
collection.<bundle_id>_data_inventory.txt (.csv)
BU_PDS_EDLdata.xml
BU_PDS_EDLdata.txt

OPTIONAL:

CONTEXT COLLECTION
collection.<bundle_id>_context.xml
collection.<bundle_id>_context_inventory.txt (.csv)

XML SCHEMA COLLECTION
collection.<bundle_id>_xml_schema.xml

collection.<bundle_id>_xml_schema_inventory.txt (.csv)