**Process for Submission of PDS4 Data to NSSDCA**

**V1.0**

**2020-07-01**

Table of Contents

[1 Introduction 1](#_Toc42781287)

[1.1 Purpose 1](#_Toc42781288)

[1.2 Actors 1](#_Toc42781289)

[2 Data Submission Process 2](#_Toc42781290)

[3 Online Status Check 4](#_Toc42781291)

[4 Data Retrieval Process 4](#_Toc42781292)

[5 Glossary 5](#_Toc42781293)

[Appendix A – Checks Performed by NSSDCA 6](#_Toc42781294)

# Introduction

## Purpose

This document describes the step-by-step procedures for submitting a PDS4 bundle from the Planetary Data System (PDS) to the Deep Archive at the NASA Space Science Data Coordinated Archive (NSSDCA). It also describes the procedure for requesting the NSSDCA to return PDS4 data from the Deep Archive The process addresses network data delivery.

## Actors

* Submitter: PDS Discipline Node that wishes to submit a PDS4 bundle to the NSSDCA. The Submitter may request the NSSDCA retrieve and return PDS4 products that have been ingested. This document uses the terms Submitter and PDS Discipline Node interchangeably.
* NSSDCA: Institution that receives and ingests the PDS4 bundle into its Deep Archive for long-term storage and for retrieving the products if requested by a Submitter.
* PDS Engineering Node Operator (PDS Operator): Engineering Node (EN) staff member responsible for registering the Submission Information Package (SIP) and the Archive Information Package (AIP), which are generated and supplied by the Submitter. After registration, the PDS Operator submits the SIP to the NSSDCA to initiate the delivery. ENR has responsibility over EN Tracking Service.

# Data Submission Process

The step-by-step procedures and protocols for submitting a PDS4 bundle to the NSSDCA Deep Archive are as follows:

1. PDS Discipline Node creates an immutable bundle, either complete or accumulating, that is stored in an online accessible repository.
2. PDS Discipline Node successfully validates the bundle using the PDS4 Validate Tool (<https://nasa-pds.github.io/validate/>).

Note:

* + A bundle must successfully validate against the PDS4 Information Model (IM) and adhere to any additional requirements specified in the PDS4 Standards Reference (<https://pds.nasa.gov/datastandards/about/>) before it can be delivered to the NSSDCA.

1. PDS Discipline Node uses the PDS Deep Archive Tool (<https://nasa-pds.github.io/pds-deep-archive/>) to create a Submission Information Package (SIP) and an Archive Information Package (AIP) for the bundle.

Notes:

* + The components of the SIP and AIP are collectively called the PDS Deep Archive Delivery package. The Glossary describes the SIP and AIP and their components in more detail.
  + The PDS Deep Archive Tool assigns a unique LIDVID to the SIP product and another one to the AIP product.
  + The NSSDCA requires only the SIP when submitting a bundle to its Deep Archive. The PDS system requires the AIP for internal registration and tracking purposes.

1. PDS Discipline Node validates the SIP and AIP products contained in the PDS Deep Archive Delivery package to ensure compliance with the most recent release of the PDS4 IM.
2. PDS Discipline Node emails the PDS Deep Archive Delivery package to the PDS Operator.
3. PDS Operator validates the SIP and AIP products contained in the PDS Deep Archive Delivery package:
   1. Upon failure:
      1. The PDS Operator emails the bundle failure status to the PDS Discipline Node, and
      2. PDS Discipline Node fixes the problems and resubmits new PDS Deep Archive Delivery packages until success (i.e., repeats steps 2-5).
   2. Upon success, the PDS Operator:
      1. Registers the SIP and AIP products into the PDS system (Registry and Tracking Service),
      2. Places the contents of the PDS Deep Archive Package in an online accessible repository at EN (<https://pds.nasa.gov/data/pds4/manifests/>), and
      3. Emails the PDS Discipline Node that the bundle is under “delivery to NSSDCA” and must remain immutable until the transfer to NSSDCA is complete.
4. To initiate the delivery to the NSSDCA, the PDS Operator uses the Planetary Submission Interface (PSI) (<https://nssdc.gsfc.nasa.gov/psi/>) to submit the URL of the SIP product to the NSSDCA, selecting the appropriate node as the Provider Site and node personnel as the Transfer Contact.

Note:

* + The URLs of multiple SIP products can be submitted simultaneously through the PSI.

1. NSSDCA initiates the transfer, registration, and ingest of the SIP product, the SIP manifest, and the files identified the SIP manifest into the Deep Archive, while simultaneously checking for anomalies that would impede ingest of the SIP product or any product identified in the manifest.

Anomalies, such as malformed XML for the SIP product will cause the NSSDCA to abort and reject the entire delivery, while other errors such as a stale file URL for a basic product in the SIP manifest will stop the NSSDCA from ingesting that entire product but the processing of the remaining products will continue. Appendix A provides a summary of the checks the NSSDCA performs and resulting actions.

Notes:

* + Successful checks ensure the delivery is incorporated into the Deep Archive holdings, where each product is registered with a unique NSSDCA identifier.
  + The NSSDCA processes and stores the SIP product and manifest in the Deep Archive repository such that the SIP can be returned unaltered upon request.

1. Upon completion the NSSDCA records the status of SIP processing in the PSI (see section 3, Online Status Check) and alerts the PDS Operator that processing is complete.
2. The PDS Operator updates the status of the EN-registered SIP to delivered and sends an email to the PDS Discipline Node confirming the delivery has been completed.

Note:

* + When a delivery is rejected or partially ingested, the PDS Discipline Node fixes the problems and resubmits new PDS Deep Archive Delivery packages until success (i.e., repeats steps 2-5). When resubmitting, the SIP product’s version identifier (VID) must be incremented to ensure uniqueness.

# Online Status Check

The PDS Discipline Node and the PDS Operator can use the PSI to check the status of a submission (e.g., rejected, pre-ingest, ingested, partially ingested with exceptions):

<https://nssdc.gsfc.nasa.gov/psi/>

Select the PDS4 Reporting tab in the left navigation bar to query by SIP LID / LIDVID, bundle LID / LIDVID, or collection LID / LIDVID.

Note:

* + Presently the NSSDCA emails the PDS Operator about specific rejections and failures. To eliminate this manual step, the NSSDCA intends to enhance PDS4 Reporting to identify which products in the SIP were ingested, which ones failed along with remarks, and which ones were not downloaded because the NSSDCA has already ingested those LIDVIDs as part of a previous submission.

# Data Retrieval Process

Upon request, NSSDCA will retrieve and return PDS4 data from the Deep Archive to the submitting PDS Discipline Node. The step-by-step procedures and protocols for requesting and retrieving PDS4 products for network distribution to the Submitter are as follows:

1. PDS Discipline Node sends an email request to the NSSDCA Coordinated Request and User Support Office (CRUSO) at:

[nssdc-request@lists.nasa.gov](mailto:nssdc-request@lists.nasa.gov)

<https://nssdc.gsfc.nasa.gov/about/about_cruso.html>

indicating one or more PDS4 products to be retrieved by listing the specific LID(s) or LIDVID(s). To ensure the correct instance of each product is retrieved, the LIDVID(s) for the associated Product\_SIP\_Deep\_Archive product(s) should be provided.

NSSDCA can retrieve PDS4 data on the basis of:

* Bundle LID or LIDVID
* Collection LID or LIDVID
* Basic Product LID or LIDVID
* SIP LID or LIDVID

If the LID is specified, the NSSDCA will retrieve the highest version (LIDVID) of that product from the Deep Archive.

1. CRUSO receives the email request and acknowledges the request by email with a reply of estimated time of preparation for network delivery to the PDS Discipline Node.
2. NSSDCA retrieves and stages the requested product(s) online.
3. CRUSO sends an email to the PDS Discipline Node with instructions for downloading the data.

# Glossary

**Archive Information Package (AIP)**

The digital information package for submitting a PDS4 bundle from a PDS Discipline Node to the PDS Operator for registration and tracking purposes within PDS system. The AIP consists of three components:

* AIP product: An XML label containing Product\_AIP, which is defined in the PDS4 Information Model. The AIP product identifies the bundle to be submitted to EN for registration in the PDS System.
* Checksum manifest: A two-column tab-delimited ASCII table that identifies all files for *all products* within that bundle. Each record in this manifest provides the file specification name and the checksum for that file.
* Transfer manifest: A two-column tab-delimited ASCII table that identifies all files for *all products* within that bundle. Each record in this manifest provides the file specification name and the product LIDVID.

**Submission Information Package (SIP)**

The digital information package for delivering a PDS4 bundle from a PDS Discipline Node to the NSSDCA Deep Archive. SIP consists of two components:

* SIP product: An XML label containing Product\_SIP\_Deep\_Archive, which is defined in the PDS4 Information Model. The SIP product identifies the bundle to be submitted to NSSDCA and the submitting PDS Discipline Node; and specifies the URL of the SIP manifest and the checksum for that file.
* SIP manifest: A four-column tab-delimited ASCII table that identifies all files for *all primary products* within that bundle to be transmitted and ingested. Each record in this manifest provides the information the NSSDCA needs to download, check fixity, and ingest one file: its checksum, checksum type, URL, checksum, and LIDVID, respectively.

**LIDVID**

The logical identifier (LID) and version identifier (VID) used to uniquely identify a PDS4 product within the PDS domain.

# Appendix A – Checks Performed by NSSDCA

The appendix summarizes the checks performed NSSDCA:

* The LIDVID for the SIP product must be unique (i.e., was not previously submitted to the NSSDCA). Otherwise the submission (i.e., the SIP) is rejected.
* The SIP product must have well-formed XML and must adhere to the schema specified in the XML. Otherwise the submission is rejected.
* The checksum of the SIP product recorded by the PSI upon submission must match the checksum computed as the NSSDCA downloads that product. Otherwise the submission is rejected.
* The checksum of the SIP manifest specified in the SIP product must match the checksum computed as the NSSDCA downloads the SIP manifest. Otherwise the submission is rejected.
* The checksum specified for the bundle product in the SIP manifest must match the checksum computed as the NSSDCA downloads that product. Otherwise the submission is rejected.
* The downloaded bundle product must have well-formed XML and must adhere to the schema specified in the XML. Otherwise the submission is rejected.
* Every URL listed in the SIP manifest must be unique. Otherwise the submission is rejected.
* The checksum specified for each collection product in the SIP manifest must match the checksum computed as the NSSDCA downloads that collection product. If not, the anomaly is recorded for that collection product but processing continues.
* Each downloaded collection product must have well-formed XML and must adhere to the schema specified in the XML. If not, the anomaly is recorded for that collection product but processing continues.
* Each collection product listed in the SIP manifest must also have an inventory table file with an extension of ‘.csv’ listed. If not, the anomaly is recorded for that collection product; that collection product is not ingested but processing of the submission continues.
* Each collection product listed in the SIP manifest must have a label file with an extension of ‘.xml’. If not, the anomaly is recorded for that collection product; that collection product is not ingested but processing of the submission continues.
* Each basic product listed in the SIP manifest must have a label file with an extension of ‘.xml’. If not, the anomaly is recorded for that basic product; that basic product is not ingested but processing of the submission continues.
* For each collection product listed in the SIP manifest, its LIDVID must have the bundle LID. If not, the anomaly is recorded for that collection product; that collection product is not ingested but processing of the submission continues.
* For each basic product listed in the SIP manifest, its LIDVID must have the collection LID and the collection LID must include the bundle LID. If not, the anomaly is recorded for that basic product; that basic product is not ingested but processing of the submission continues.