Planetary Data System

Harvest-PDAP Tool

Software Requirements and Design Document (SRD/SDD)

Sean Hardman

April 18, 2012
Version 0.1

Jet Propulsion Laboratory
Pasadena, California

CHANGE LOG

Revision	Date	Description	Author
0.1	2012-04-18	Initial draft.	S. Hardman

TABLE OF CONTENTS

1.0 INTRODUCTION

The PDS 2010 effort will overhaul the PDS data architecture (e.g., data model, data structures, data dictionary, etc) and deploy a software system (online data services, distributed data catalog, etc) that fully embraces the PDS federation as an integrated system while leveraging modern information technology.

This tool provides functionality for capturing and registering data set metadata from a service that supports the Planetary Data Access Protocol (PDAP). Although the tool should support any service with a PDAP interface, the service of interest is the Planetary Science Archive (PSA) of the European Space Agency (ESA). The tool will run locally at the Engineering Node to query the PSA data set registry in order to discover data sets and register associated metadata with the Registry (Inventory) service.

1.1 Document Scope and Purpose

This document addresses the use cases, requirements and software design of the Harvest-PDAP tool within the PDS 2010 data system. This document is intended for the reviewer of the tool as well as the developer and tester of the tool.

1.2 Method

This combined Software Requirements and Software Design Document (SRD/SDD) represents the software by defining use cases and requirements and by using architecture diagrams, functional descriptions, context diagrams and data flow diagrams for the high-level design. UML diagrams will illustrate the detailed design.

1.3 Notation

The numbering of the requirements in this document will be formatted as **LX.HVT.AA.X**, where:

- **LX** represents the requirements level where X is a number.
- **HVT** is an abbreviation representing the harvest requirement section for the specified level.
- **AA** is a two-letter abbreviation representing the requirement sub-category (optional).
- **X** is a unique number within the section and optional sub-category for the requirement.

Following the text of a requirement may be a reference to the requirement or use case from which it was derived. The reference will be in parenthesis. A

paragraph following a requirement, which is indented and has a reduced font size, represents a comment providing additional insight for the requirement that it follows. This comment is not part of the requirement for development or testing purposes.

1.4 Controlling Documents

[1] Planetary Data System (PDS) Level 1, 2 and 3 Requirements, March 26, 2010.

[2] Planetary Data System (PDS) 2010 Project Plan, February 2010.

[3] Planetary Data System (PDS) 2010 System Architecture Specification, Version 1.2, May 25, 2011.

[4] Planetary Data System (PDS) 2010 Operations Concept, February 2010.

[5] Planetary Data System (PDS) General System Software Requirements Document (SRD), Version 1.0, June 11, 2011.

1.5 Applicable Documents

[6] Planetary Data Access Protocol (PDAP), Version 1.1, September 9, 2011.

[7] VOTable Format Definition, Version 1.2, November 30, 2009.

[8] Planetary Data System (PDS) Registry Service Software Requirements and Design Document (SRD/SDD), Version 1.0, June 12, 2011.

[9] PDS4 Information Model Specification, PDS4 Information Model Specification Team.

[10] PSA to PDS4 Metadata Mapping, April 18, 2012.

1.6 Document Maintenance

The component design will evolve over time and this document should reflect that evolution. This document is limited to design content because the specification content will be captured in separate documentation (e.g., Installation Guide, Operation Guide, etc.). This document is under configuration control.

2.0 COMPONENT DESCRIPTION

This tool provides functionality for capturing and registering data set metadata from a service that supports the Planetary Data Access Protocol (PDAP). Although the tool should support any service with a PDAP interface, the service of interest is the Planetary Science Archive (PSA) of the European Space Agency (ESA). The tool will run locally at the Engineering Node to query the PSA data set registry in order to discover data sets and register associated metadata with the Inventory service. The following diagram details the context of the Harvest-PDAP tool within the system:

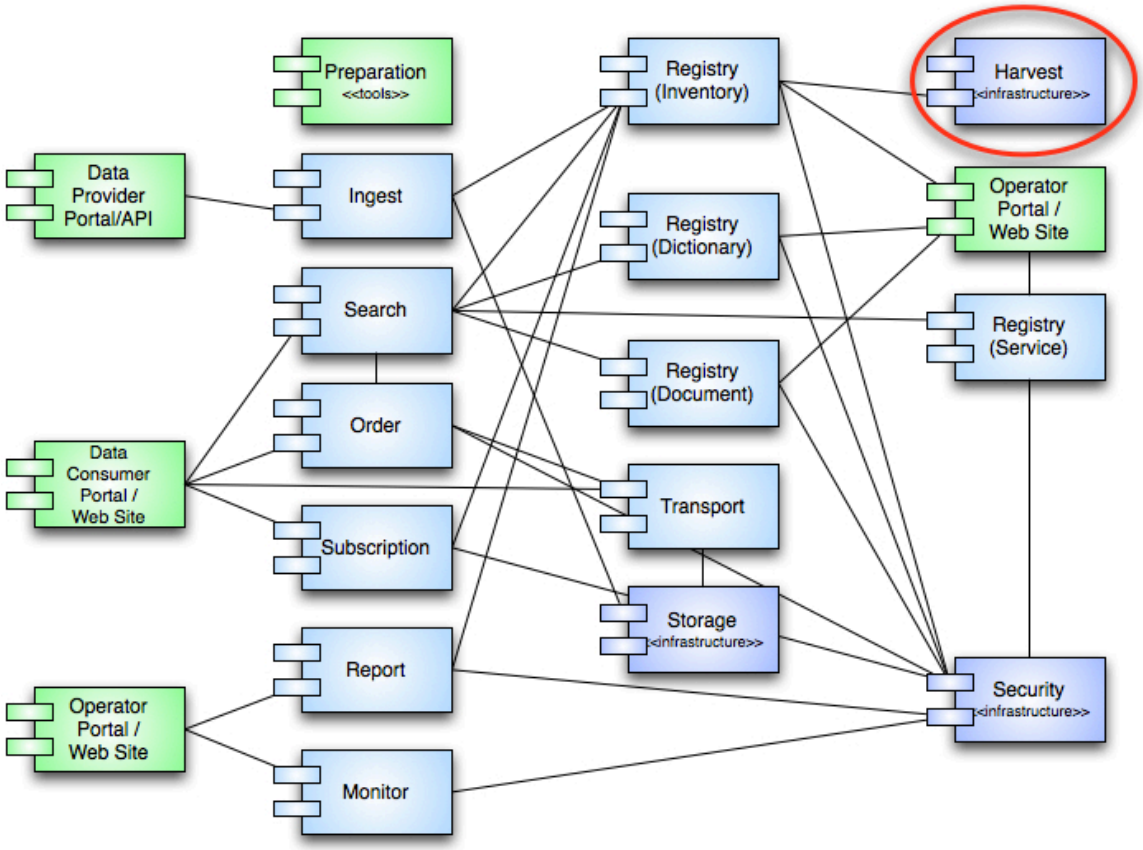

Figure 1: Harvest-PDAP Tool Context

Within the PDS 2010 system (referred to as the “system” from this point forward), the Harvest-PDAP tool is an infrastructure component. This means that there will not be any external interfaces to the tool. As depicted in the diagram above, the Harvest-PDAP tool supports a single interface with the Inventory (Registry) service. This interface has a single purpose and that is to register products from ESA’s PSA registry to its associated Inventory service instance. The Inventory service is an instance of the Registry service that offers an Application Programming Interface (API) for interacting with that service. The details regarding the tool interface can be found in section 6.2.

Unlike the original Harvest tool that crawls a local repository and extracts metadata from product labels, the Harvest-PDAP tool queries the PSA registry via the Planetary Data Access Protocol (PDAP) [6] to retrieve data set metadata which is then registered with a Registry Service instance.

3.0 USE CASES

A use case represents a capability of the component and why the user (actor) interacts with the component. It should be at a high enough level so as not to reveal or imply the internal structure of the system. An actor is an object (e.g., person, application, etc.) outside the scope of the component but interacts with the component. This section captures the use cases for the Harvest-PDAP tool based on the description of the component from the previous section. These use cases will be used in the derivation of requirements for the component. The following diagram details the use cases:

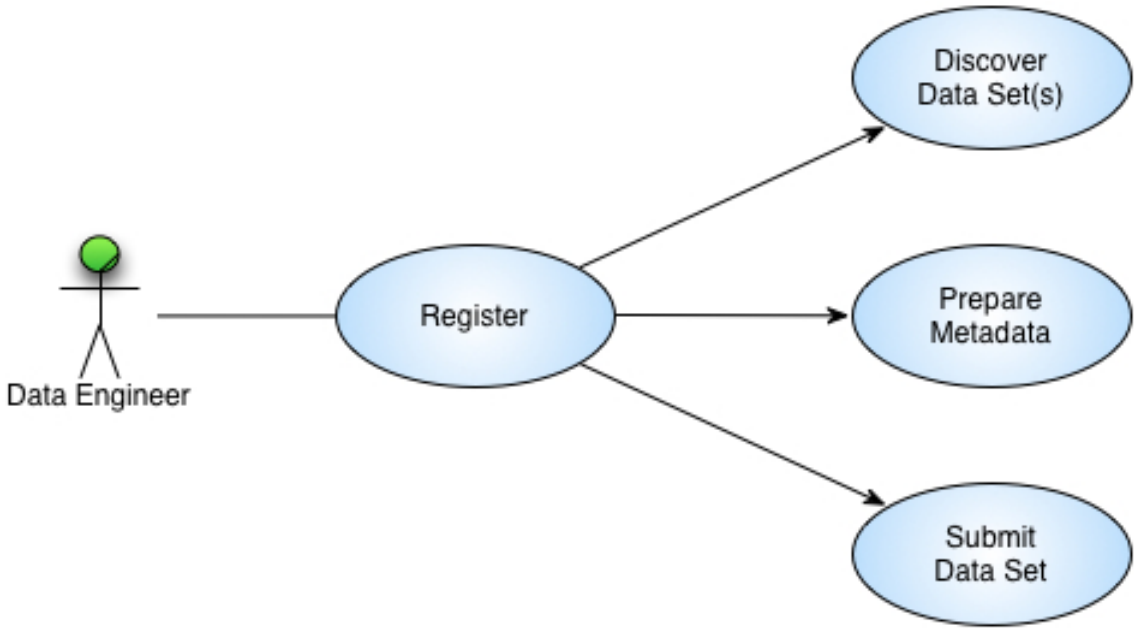

Figure 2: Harvest-PDAP Tool Use Cases

The above diagram identifies the following actor (represented as a stick figure):

Data Engineer
This actor represents a portion of the PDS Technical group that curates the data before and after it enters the PDS system.

The following sections detail the use cases identified in the above diagram.

3.1 Register

The tool runs in a mode where it performs one query against the PSA registry and registers the data sets discovered. This use case pertains to the Data Engineer actor.

1. Data Engineer executes the Harvest-PDAP tool specifying the configuration file.
2. Harvest-PDAP tool queries for data sets in the PDS catalog (include Discover Data Set(s) use case).
3. Harvest-PDAP tool prepares metadata for each discovered data set (include Prepare Metadata use case).
4. Harvest-PDAP tool registers each data set with the target Registry service instance (include Submit Data Set use case).

3.2 Discover Data Set(s)

Data sets are discovered based on the results returned from a query to the PSA registry. This use case is included as part of the Register use case.

1. Harvest-PDAP tool obtains criteria for accessing the PDS catalog from the configuration file.
2. Harvest-PDAP tool queries the PDS catalog for the list of data sets and their associated metadata.
3. Harvest-PDAP tool discovers candidate data set(s).

Alternative: Previously Discovered Data Set
At step 3, the tool has already registered the discovered data set product.

a. Harvest-PDAP tool determines a previous registration for a candidate data set product and skips it.
b. Return to primary scenario at step 2.

3.3 Prepare Metadata

Metadata is prepared for a discovered data set based on the associated metadata returned from the PSA registry. This use case is included as part of the Register use case.

1. Harvest-PDAP tool determines the metadata for a data set based on the associated metadata returned from the PSA registry.
2. Harvest-PDAP tool retrieves the *Dataset.cat* file from the PSA repository and extracts additional metadata from that file for the data set.
3. Harvest-PDAP tool formats the metadata for submission to the Registry service.

3.4 Submit Data Set

A data set and its associated metadata are submitted to the target instance of the Registry service. This use case is included as part of the Register use case.

1. Harvest-PDAP tool authenticates for access to the Registry service API (include Security service Authenticate User use case).
2. Harvest-PDAP tool submits the associated metadata for a product for registration via the Registry service API.
3. Registry service responds with a successful status regarding the registration and the global unique identifier for the product.
4. Harvest-PDAP tool logs the registration.

Alternative: Product Registration Fails

At step 2, the product registration fails for any number of reasons.

a. Registry service returns an exception with cause of failure.
b. Harvest-PDAP tool logs the exception.

4.0 REQUIREMENTS

It has been determined that the Use Cases above are sufficient for capturing the desired capabilities of the Harvest-PDAP tool. Therefore, requirements will not be specified in addition to the Use Cases.

5.0 DESIGN PHILOSOPHY, ASSUMPTIONS, AND CONSTRAINTS

The intent of the Harvest-PDAP tool is to provide a simple solution for querying the PSA registry for the purpose of registering data set products into the federated system of registries.

The PSA registry offers a couple of different interfaces for obtaining metadata, but we chose to use the PDAP interface. PDAP is a REST-based API for interacting with the PSA registry and is based on an International Planetary Data Alliance standard.

6.0 ARCHITECTURAL DESIGN

The architectural design covers the component breakdown within the tool, external/internal interfaces and the associated data model.

6.1 Component Architecture

The following diagram details the architecture for the Harvest-PDAP tool:

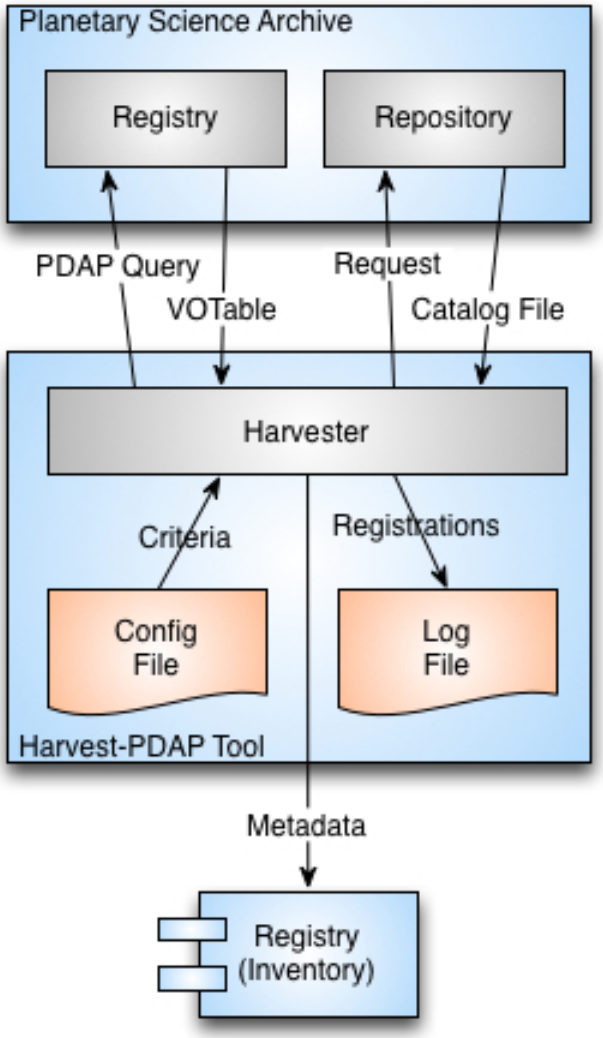

Figure 3: Harvest-PDAP Tool Architecture

The Harvest-PDAP tool consists simply of a Harvester component that receives its configuration from a local configuration file as well as from command-line parameters. It queries the PSA registry via the PDAP interface and receives metadata for candidate data set products in VOTable [7] format. See Appendix B for an example VOTable structure. The Harvester may also retrieve the

associated *Dataset.cat* file from the PSA repository to gather additional metadata regarding the data set. The Harvester then registers those data set products with the target Registry service instance using the REST-based Registry service API. The Registry Service SRD/SDD [8] documents the API in detail. A local log file captures each registration.

6.2 External Interface Design

The external interface for the Harvest-PDAP tool is limited to the command-line interface and the configuration file. The tool utilizes Apache's CLI (Command-Line Interface) library for accepting options on the command-line. The command-line interface accepts the following options:

- File specification for the configuration file
- File specification for the log file
- User name and password for registering with a secured Registry service

The configuration file utilizes an XML structure for specifying additional information pertaining to a specific harvest execution. The following information is typical:

- End point for the PDAP interface.
- End point for the Registry Service.
- Static metadata to be registered with each Data Set.

6.3 Internal Interface Design

The Harvest-PDAP tool does not have any internal interfaces of consequence.

6.4 Data Model

The Harvest-PDAP tool does not have an associated data model but the metadata that passes to the Registry service for data set product registration is subject to the PDS4 Information Model Specification [9].

7.0 ANALYSIS

Detailed analysis was not necessary given the scope of this component.

8.0 IMPLEMENTATION

The PDS 2010 system is a phased implementation with increasing capabilities delivered in three planned builds. The builds are as follows:

- **Build 1** – This build consists of the Ingestion subsystem including the Security, Harvest, Registry (Inventory, Dictionary, Document, Service) and Report components along with the Data Provider tool suite.
- **Build 2** – This build consists of the Distribution subsystem including the Search and Monitor components along with a revised web site and general portal applications.
- **Build 3** – This build consists of enhanced user capabilities include the Order and Subscription components along with integration of Discipline Node applications and science services.

The Harvest-PDAP tool, implemented in conjunction with the Registry service, is scheduled for delivery in Build 2. This initial delivery will support test collection generation and registration. Additional capabilities are planned for follow-on deliveries as testing progresses and the data model matures.

The implementation platform for the Harvest-PDAP tool is the Java 2 Platform Standard Edition 6.0. In addition, development will utilize publically available libraries for command-line option handling, message handling and HTTP interfacing. It will also utilize the PDS common library for parsing and reading catalog files for metadata extraction.

The scenario for the preferred deployment is to run an instance of the Harvest-PDAP tool and an instance of the Registry service locally at the Engineering Node. The following diagram depicts this deployment scenario:

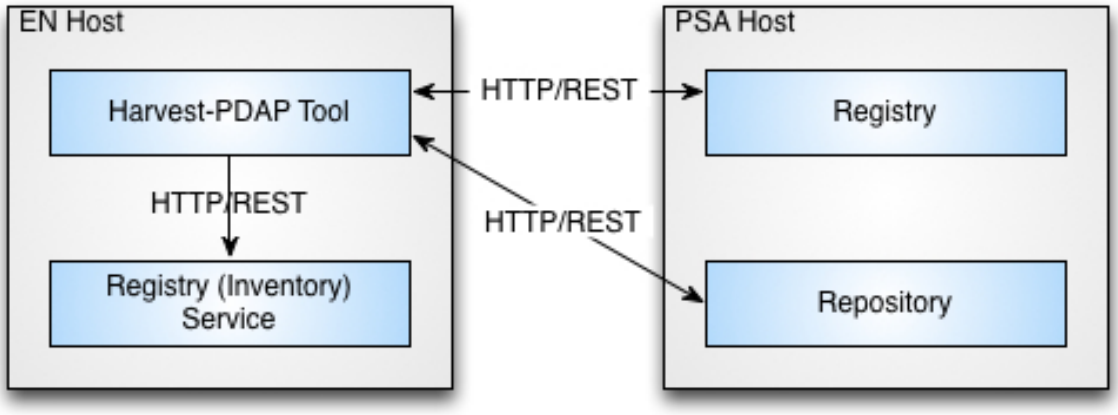

Figure 4: Harvest-PDAP Tool Deployment

The Harvest-PDAP tool provides for execution from the command-line. The Harvest-PDAP tool queries and retrieves metadata/data from PSA registry/repository via a REST-based interface using the Hypertext Transfer Protocol (HTTP). The interface with the PSA registry utilizes PDAP over HTTP. The Harvest tool submits data set product registrations to the Registry service via its REST-based interface also using HTTP.

9.0 DETAILED DESIGN

This section offers a more detailed look at certain aspects of the Harvest-PDAP Tool design. The following diagram details the activity flow of the software:

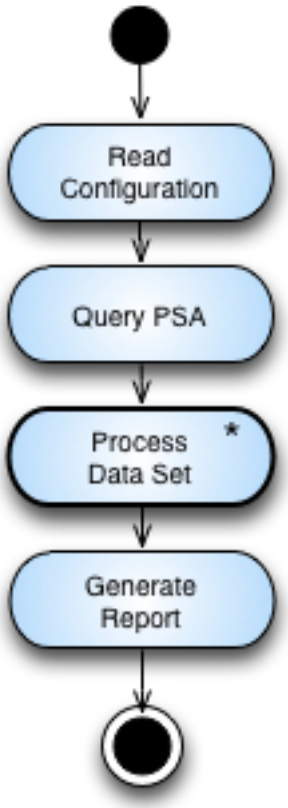

Figure 5: Harvest-PDAP Tool Activity (Overview)

The activity titled “Process Data Set *” in the diagram above represents an iteration over all candidate data set products identified in the query results from the “Query PSA” activity. Example URLs for the PSA interface can be found in Appendix C.

9.1 Process Data Set

The following diagram details the activity flow for this activity:

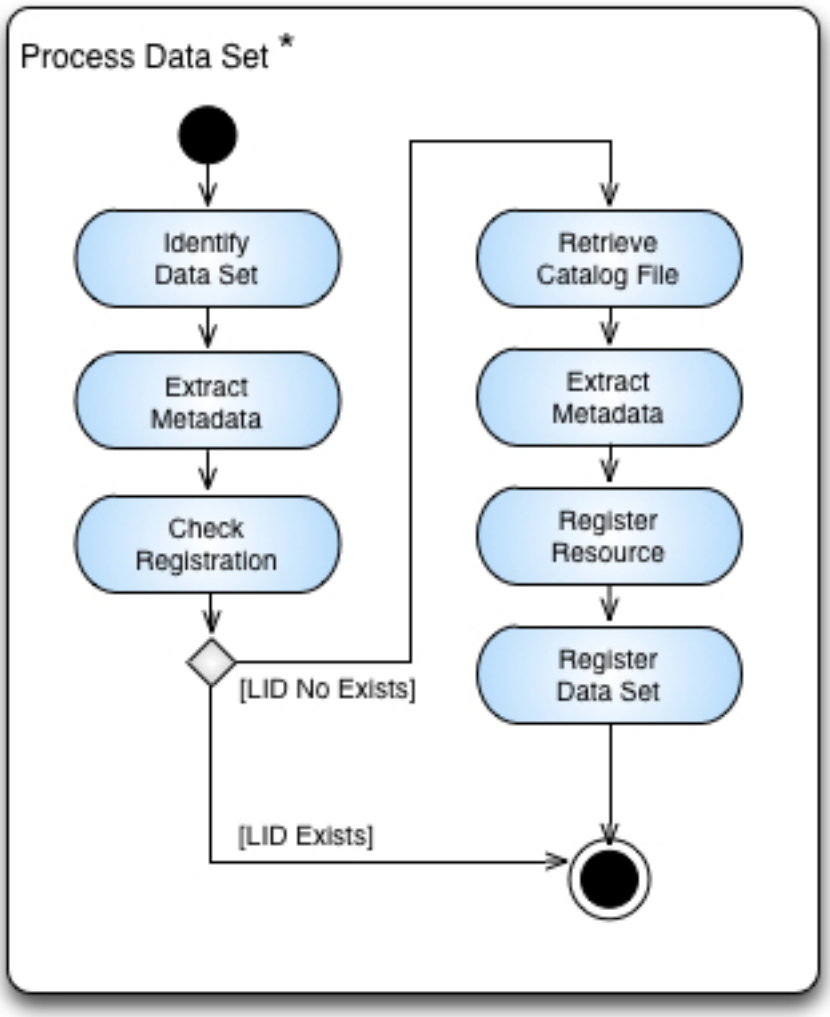

Figure 6: Harvest-PDAP Tool Activity (Process Data Set)

The first “Extract Metadata” activity above involves reading the VOTable entry for the target data set and mapping [10] the appropriate fields to the Product_Data_Set_PDS3 product from the PDS4 Information Model [9]. The second “Extract Metadata” activity above involves reading the Data Set catalog file and mapping the additional fields to the Product_Data_Set_PDS3 product as well.

In order to support the current PDS Data Search capability, the tool also needs to register an associated Product_Resource product containing the URL to the HTML page for a given data set.

APPENDIX A ACRONYMS

The following acronyms pertain to this document:

API	Application Programming Interface
CLI	Command-Line Interface
ESA	European Space Agency
HTTP	Hypertext Transfer Protocol
JPL	Jet Propulsion Laboratory
LID	Logical Identifier
NASA	National Aeronautics and Space Administration
PDS	Planetary Data System
PDS4	Version 4 of the PDS Standards
PSA	Planetary Science Archive
REST	Representational State Transfer
SDD	Software Design Document
SRD	Software Requirements Document
URL	Uniform Resource Locator
XML	Extensible Markup Language

APPENDIX B VOTABLE EXAMPLE

This is an example VOTable structure returned from the PSA registry that contains a single data set. Structures with multiple data sets will have additional <TR> blocks under the <TABLEDATA> block.

```
<?xml version="1.0"?>
<!DOCTYPE VOTABLE SYSTEM "http://us-vo.org/xml/VOTable.dtd">
<VOTABLE version="1.1">
  <RESOURCE type="results">
    <DESCRIPTION>PSA Metadata Query Service</DESCRIPTION>
    <INFO name="QUERY_STATUS" value="OK" />
    <TABLE>
      <FIELD ID="DATA_SET.DATA_SET_ID" ucd="DATA_SET_ID"
utype="pds:DATA_SET.DATA_SET_ID" datatype="char" arraysize="*"/>
      <FIELD ID="DATA_SET.DATA_SET_NAME" ucd="DATA_SET_NAME"
utype="pds:DATA_SET.DATA_SET_NAME" datatype="char" arraysize="*"/>
      <FIELD ID="DATA_SET.DATA_ACCESS_REFERENCE"
ucd="DATA_ACCESS_REFERENCE" datatype="char" arraysize="*"/>
      <FIELD ID="DATA_SET.XML_DESCRIPTION" ucd="XML_DESCRIPTION"
datatype="char" arraysize="*"/>
      <FIELD ID="DATA_SET.PRODUCER.FULL_NAME" ucd="FULL_NAME"
utype="pds:DATA_SET.PRODUCER.FULL_NAME" datatype="char" arraysize="*"/>
      <FIELD ID="DATA_SET.PRODUCER.INSTITUTION_NAME"
ucd="INSTITUTION_NAME" utype="pds:DATA_SET.PRODUCER.INSTITUTION_NAME"
datatype="char" arraysize="*"/>
      <FIELD ID="DATA_SET.PRODUCER.NODE_NAME" ucd="NODE_NAME"
utype="pds:DATA_SET.PRODUCER.NODE_NAME" datatype="char" arraysize="*"/>
      <FIELD ID="DATA_SET.START_TIME" ucd="START_TIME"
utype="pds:DATA_SET.START_TIME" datatype="char" arraysize="*"/>
      <FIELD ID="DATA_SET.STOP_TIME" ucd="STOP_TIME"
utype="pds:DATA_SET.STOP_TIME" datatype="char" arraysize="*"/>
      <FIELD ID="DATA_SET.NPRODUCTS" ucd="NPRODUCTS"
utype="pds:DATA_SET.NPRODUCTS" datatype="char" arraysize="*"/>
      <FIELD ID="DATA_SET.MISSION_NAME" ucd="MISSION_NAME"
utype="pds:DATA_SET.MISSION_NAME" datatype="char" arraysize="*"/>
      <FIELD ID="DATA_SET.INSTRUMENT_ID" ucd="INSTRUMENT_ID"
utype="pds:DATA_SET.INSTRUMENT_ID" datatype="char" arraysize="*"/>
      <FIELD ID="DATA_SET.INSTRUMENT_NAME" ucd="INSTRUMENT_NAME"
utype="pds:DATA_SET.INSTRUMENT_NAME" datatype="char" arraysize="*"/>
      <FIELD ID="DATA_SET.TARGET_NAME" ucd="TARGET_NAME"
utype="pds:DATA_SET.TARGET_NAME" datatype="char" arraysize="*"/>
      <FIELD ID="RESOURCE_CLASS" ucd="RESCLASS" datatype="char"
arraysize="*"/>
      <DATA>
        <TABLEDATA>
          <TR>
            <TD>AIRUB-C-PHOTOCAM-2-EDR-HALLEY-1986-V1.0</TD>
            <TD><![CDATA[AIRUB-HALLEY-PHOTOGRAPHIC-PROJECT-EDR-1986-
V1.0]]></TD>
            <TD><![CDATA[http://psa.esac.esa.int:8000/aio/jsp/ \
product.jsp?dataSetID=AIRUB-C-PHOTOCAM-2-EDR-HALLEY-1986-
V1.0&compression=tar&protocol=HTTP]]></TD>
            <TD><![CDATA[http://psa.esac.esa.int:8000/aio/jsp/ \
fileXML.jsp?DATA_SET_ID=AIRUB-C-PHOTOCAM-2-EDR-HALLEY-1986-V1.0]]></TD>
            <TD><![CDATA[WERNER E. CELNIK]]></TD>
```

```
            <TD><![CDATA[ASTRONOMISCHES INSTITUT DER RUHR-UNIVERSITAET
BOCHUM]]></TD>
            <TD></TD>
            <TD>1986-02-16 00:00:00.0</TD>
            <TD>1986-04-18 00:00:00.0</TD>
            <TD>1833</TD>
            <TD><![CDATA[EARTH]]></TD>
            <TD>300,FFC,HBL,HUV,RUV</TD>
            <TD><![CDATA[HASSELBLAD-ZEISS-PLANAR-F2-110MM,HASSELBLAD-
ZEISS-UV-SONNAR-F4.3-105MM,LICHTENKNECKER-FLAT-FIELD-CAMERA-F4-
760MM,PENTACON-OPTICS-F4-300MM,ROLLEI-ZEISS-UV-SONNAR-F4.3-
105MM]]></TD>
            <TD>1P/HALLEY,M83</TD>
            <TD>DATA_SET</TD>
          </TR>
        </TABLEDATA>
      </DATA>
    </TABLE>
  </RESOURCE>
</VOTABLE>
```

APPENDIX C PSA ACCESS

This appendix provides example URLs for accessing the PSA registry and repository.

- Main page for the PSA Archive InterOperability System
  http://psa.esac.esa.int:8000/aio/doc/

- PDAP query that returns all data sets in VOTable XML format
  http://psa.esac.esa.int:8000/aio/jsp/metadata.jsp?RETURN_TYPE=VOTABLE

- PDAP query that returns a single data set in VOTable XML format
  http://psa.esac.esa.int:8000/aio/jsp/metadata.jsp?DATA_SET_ID=AIRUB-C-PHOTOCAM-2-EDR-HALLEY-1986-V1.0&RETURN_TYPE=VOTABLE

- PDAP query that returns a single data set in HTML format (this will be the resource link)
  http://psa.esac.esa.int:8000/aio/jsp/metadata.jsp?DATA_SET_ID=AIRUB-C-PHOTOCAM-2-EDR-HALLEY-1986-V1.0&RETURN_TYPE=HTML

- HTTP request for a data set catalog file
  http://psa.esac.esa.int:8000/aio/jsp/product.jsp?dataSetID=AIRUB-C-PHOTOCAM-2-EDR-HALLEY-1986-V1.0&productID=&path=CATALOG/&fileName=DATASET.CAT&protocol=HTTP