

---

# ***European Space Agency***

Research and Scientific Support Department

Planetary Missions Division

---

Planetary Science Archive  
Experiment Data Release Concept  
Technical Proposal

SOP-RSSD-TN-015

Issue 1.16

12 May 2005

---

Prepared by: J. Zender



**CHANGE RECORD SHEET**

Date	Iss.	Rev.	P.	Description/Authority	CR No.
2003-07-11	1	1		inserted new chapter 3 on interface details for data producer	
2003-10-13	1	3		Updated index part; introduced doc_index and others; Editorial changes;	
2004-04-16	1	5		<ul style="list-style-type: none"> <li>- Changed all release and revision indices from 3 characters to 4 characters to be compliant with the PDS dictionary</li> <li>- add information that the description part of the release object shall cover all previous revisions</li> <li>- add some information what the PSA end user will see when data products are deleted or updated</li> </ul>	
2004-05-21	1	6		Major change to the document;	
2004-06-15	1	8			
2004-06-16	1	9		Change to index file handling Updates of all example diagrams Updates of FAQ section	
2004-07-08	1	10		Editorial changes and corrections in diagrams after discussion with PSA-Vilspa	
2004-09-02	1	11		Added section 2.3 on handling of index files Added section 2.4 on handling of files below extras directory Added section 2.5 on handling the *INF.TXT files	
2004-09-03	1	12		Editorial changes to sections 2.4 and 2.5 (delete replaced by superseded)	
2004-10-15	1	13		added guide to use the pvv with the release concept general updates to ensure consistency between this and the PVV UM.	
2005-05-02	1	15		Updated guide to using PVV after development of the 'pvv build' command.	
2005-05-12	1	16		Updated guide to using the 'pvv build' and new 'pvv freeze' command, plus a description of how to manually construct a delta archive when using 'pvv freeze'.	

**DISTRIBUTION LIS**

Recipient	Organisation	Recipient	Organisation
All Mars Express Archive contact points			



## TABLE OF CONTENTS

<b>1 INTRODUCTION .....</b>	<b>5</b>
1.1 Purpose of the document.....	5
1.2 Background .....	5
1.3 Naming Convention .....	5
1.4 Overview .....	5
1.4.1 The First Release .....	6
1.4.2 The Second Release.....	7
1.4.3 Usage of Revisions .....	9
1.4.4 Updating Catalogue Information.....	11
1.4.5 Updating Documentation, Software, Geometry, Label, Extras and Calibration Directory Information .....	11
<b>2 OVERVIEW .....</b>	<b>13</b>
2.1 Glossary: Archive Volume, Delivery, Data Set, Release, Revision .....	13
2.1.1 Archive Volume.....	13
2.1.2 Delivery.....	13
2.1.3 Data Set.....	13
2.1.4 Data Set Release .....	13
2.1.5 Data Set Revision .....	13
2.2 Usage of the VOLDESC.CAT File.....	14
2.2 The DATA_SET_RELEASE Object .....	15
2.3 The INDEX Directory and INDEX Files .....	18
2.4 Content of the EXTRAS Directory .....	18
2.5 The *INFO.TXT Files .....	18
<b>3. GENERAL RULES.....</b>	<b>19</b>
<b>4. THE PVV TOOL AND THE RELEASE OBJECT: A QUICK GUIDE .....</b>	<b>19</b>
4.1 Preparation and Ingestion of Release 0001/Revision 0000 .....	21
4.2 Preparation and Ingestion of Release 0002/Revision 0000 .....	22
4.3 Preparation and Ingestion of Release 0001/Revision 0001 .....	24



<b>4.4 Preparation and Ingestion of Release 0001/Revision 0001 .....</b>	<b>25</b>
4.4.1 Manually constructing your first Release [Release0001 Revision0000] .....	25
4.4.2 Manually constructing your second Release [Release0002 Revision0000].....	25
4.4.3 Manually constructing a Revision [e.g. Release0001 Revision0001].....	26
<b>5. FREQUENTLY ASKED QUESTIONS .....</b>	<b>27</b>



## 1 Introduction

### 1.1 Purpose of the document

This document is a guideline to the experimenters on ESA's planetary missions that prepare the scientific archive using the release concept.

This document describes a data release concept to be used for instrument teams to ingest data into the Planetary Science Archive (PSA).

### 1.2 Background

Until June 2003 it was planned that experiment data are always ingested into the PSA as a complete PDS compatible *data set*, one on each *archive volume*. To remind the reader: an archive volume consists of all directories (index, document, label, calib, software, catalog) and the required and optional files below these directories.

A regular delivery of all of this information with new instrument data is repetitive, redundant and may be considered unnecessary.

A new approach that allows the delivery of experiment data **without** the delivery of all supplementary information is presented in this proposal. This approach reflects a delivery option for a data producer and will (tbc) be transparent for the end user.

In addition, an individual delivery from a data producer, a release, might not fulfill all PDS standard requirements, however the full data set itself is PDS compatible.

### 1.3 Naming Convention

As the concept of a *volume* as defined in the PDS standard is based on physical media, e.g. CD-Rs, the PSA will not use the name *volume*. Instead, the concept of deliveries is defined for the PSA and the term *delivery* is used for the PSA.

Within this document the term 'PSA compatible' is used for a dataset that is compliant with the PDS standard version 3.6 and successfully validated by the PVV (PSA Volume Verifier).

### 1.4 Overview

This chapter gives an overview and a step-by-step approach to using the release concept for testing and ingesting data into the PSA archive.

One dataset can be split up in several releases that together form a complete PSA compatible dataset.

A data set *release* is a set of data from a fixed, consecutive time period.

It must be ensured that:

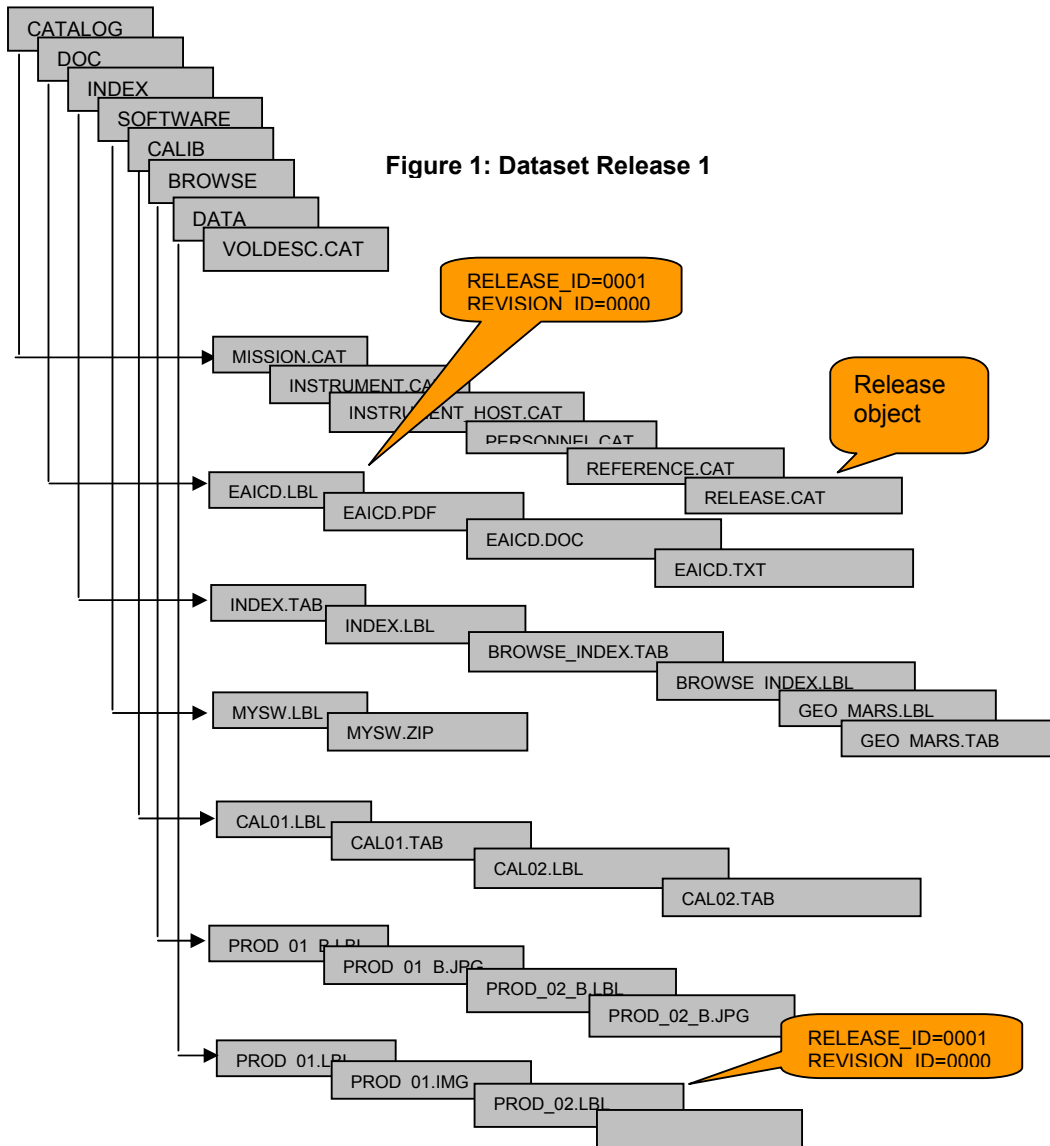
- the first release (RELEASE\_ID=0001, REVISION\_ID=0000) is a fully validated, complete PSA dataset
  - o all directories and files required by the PDS standard need to be present;
  - o The PVV tool needs to be able to validate the data set successfully (see Section 4);
  - o **All** labels within the first release must contain the two keywords
    - RELEASE\_ID = 0001
    - REVISION\_ID = 0000
  - o This really means all labels, also those in the CATALOG, INDEX, SOFTWARE, GEOMETRY, etc directories. The only exception to this is for the labels in the EXTRAS directory, which do not require RELEASE\_ID and REVISION\_ID

- keywords.
  - o Containing initial releases of all catalog files
- Any further release (e.g. RELEASE\_ID=0002, REVISION\_ID=0000) contains
  - o Only the data products covering a subsequent period to the first release
  - o An update of the release catalog file
  - o All updated index tables and labels in the INDEX directory.

### 1.4.1 The First Release

The first release of a dataset distinguishes itself from a 'typical' PDS dataset by

- containing a DATA\_SET\_RELEASE object within a release catalogue file (RELEASE.CAT)
- containing the keywords RELEASE\_ID and REVISION\_ID with ALL of its labels.





The first release can be verified with the PVV tool by the following sequence (see Section 4 for details):

- `pvv scan -Drelease=1`
- `pvv verify -Drelease=1`

To automatically construct a delta archive for the release:

- `pvv build -Drelease=1`
- `pvv upload FULL_PATH/RELEASE1_FILENAME`

Or, to manually construct your own delta archive:

- `pvv freeze -Drelease=1`
- use your preferred ftp client to upload the file

#### *1.4.2 The Second Release*

The second release of a dataset distinguishes itself from the first release by having data covering a consecutive time period appended to the original dataset release. It is important to mention that the structure and the files of the first release must be available for the PVV tool to run properly and to verify the dataset structure. Further,

- the `DATA_SET_RELEASE` objects within the release catalogue file must be updated, and the `RELEASE_ID` of the release catalogue must be updated;
- the new data labels must contain the keywords `RELEASE_ID=0002` and `REVISION_ID=0000`;
- the `INDEX.TAB` and `INDEX.LBL` files will need to be updated to include all new products, and the `INDEX.LBL` should have a `RELEASE_ID` updated to the current value.

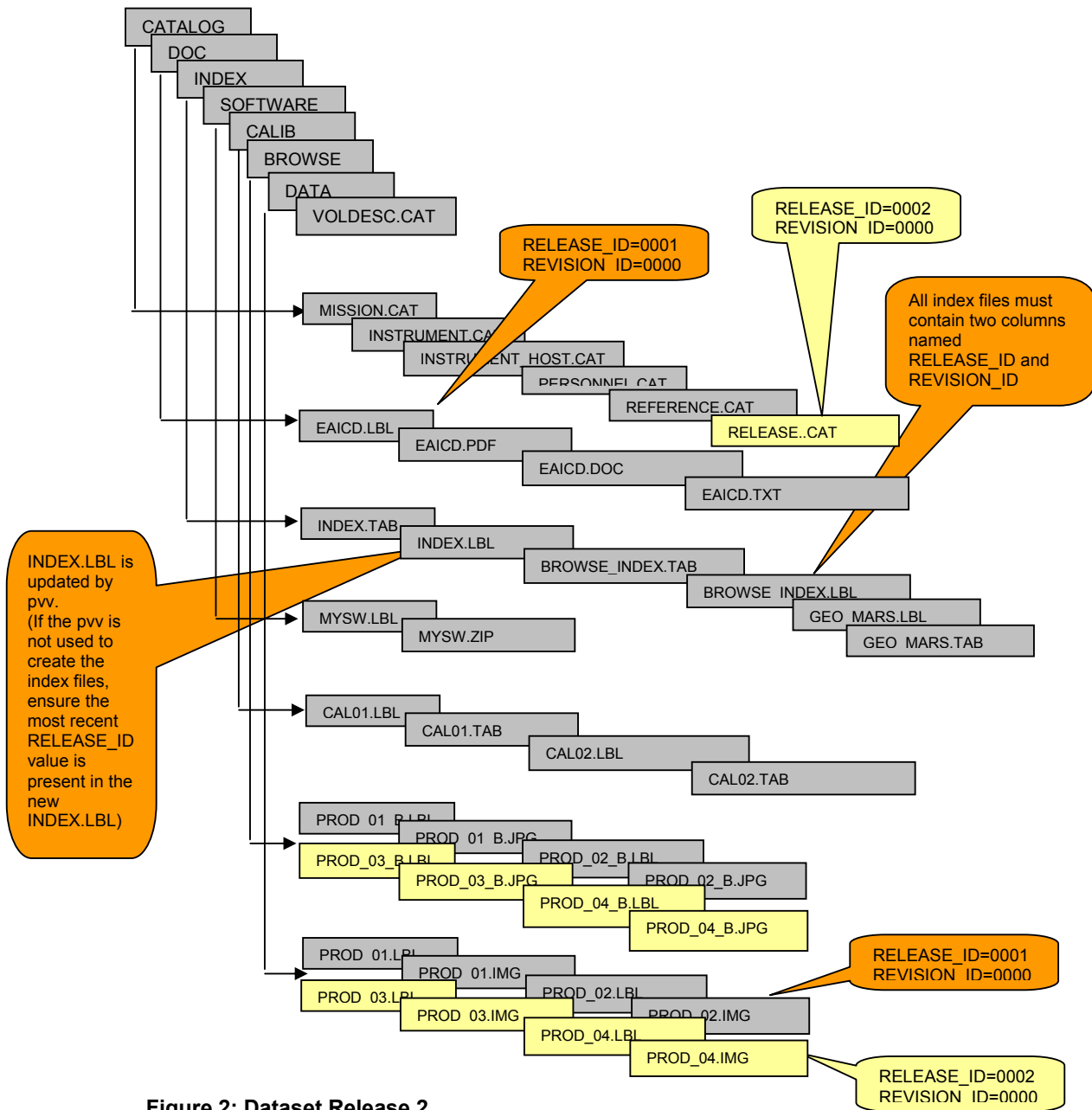


Figure 2: Dataset Release 2



The second release can be verified with the PVV tool by the following sequence (see Section 4 for details):

- pvv scan -Drelease=2
- pvv verify -Drelease=2

To automatically construct a delta archive for the release:

- pvv build -Drelease=2
- pvv upload FULL\_PATH/RELEASE2\_FILENAME

Or, to manually construct your own delta archive:

- pvv freeze -Drelease=2
- use your preferred ftp client to upload the file

### 1.4.3 Usage of Revisions

Revisions are used to

- update/replace files from a former release, and
- add files that belong to a time period for which data has been delivered in a previous release

As an example, to update data product PROD\_04 from Release number 2, the following steps are necessary:

- update the correct DATA\_SET\_RELEASE object in the RELEASE.CAT by adding a REVISION sub-object
- update the PROD\_04.IMG
- update the PROD\_04.LBL (RELEASE\_ID=0002, REVISION\_ID=0001)
- update the browse product
  - update the PROD\_04\_B.JPG
  - update the PROD\_04\_B.LBL (RELEASE\_ID=0002, REVISION\_ID=0001)
- update the relevant index tables and labels. As a minimum, the INDEX.TAB should be re-created to include the new revision and the INDEX.LBL should be updated with the new RELEASE\_ID and REVISION\_ID

The pvv can then be run as follows (see Section 4 for details):

- pvv scan -Drelease=2 -Drevision=1
- pvv verify -Drelease=2 -Drevision=1

To automatically construct a delta archive for the release:

- pvv build -Drelease=2 -Drevision=1
- pvv upload FULL\_PATH/RELEASE2\_REVISION1\_FILENAME

Or, to manually construct your own delta archive:

- pvv freeze -Drelease=2 -Drevision=1
- use your preferred ftp client to upload the file

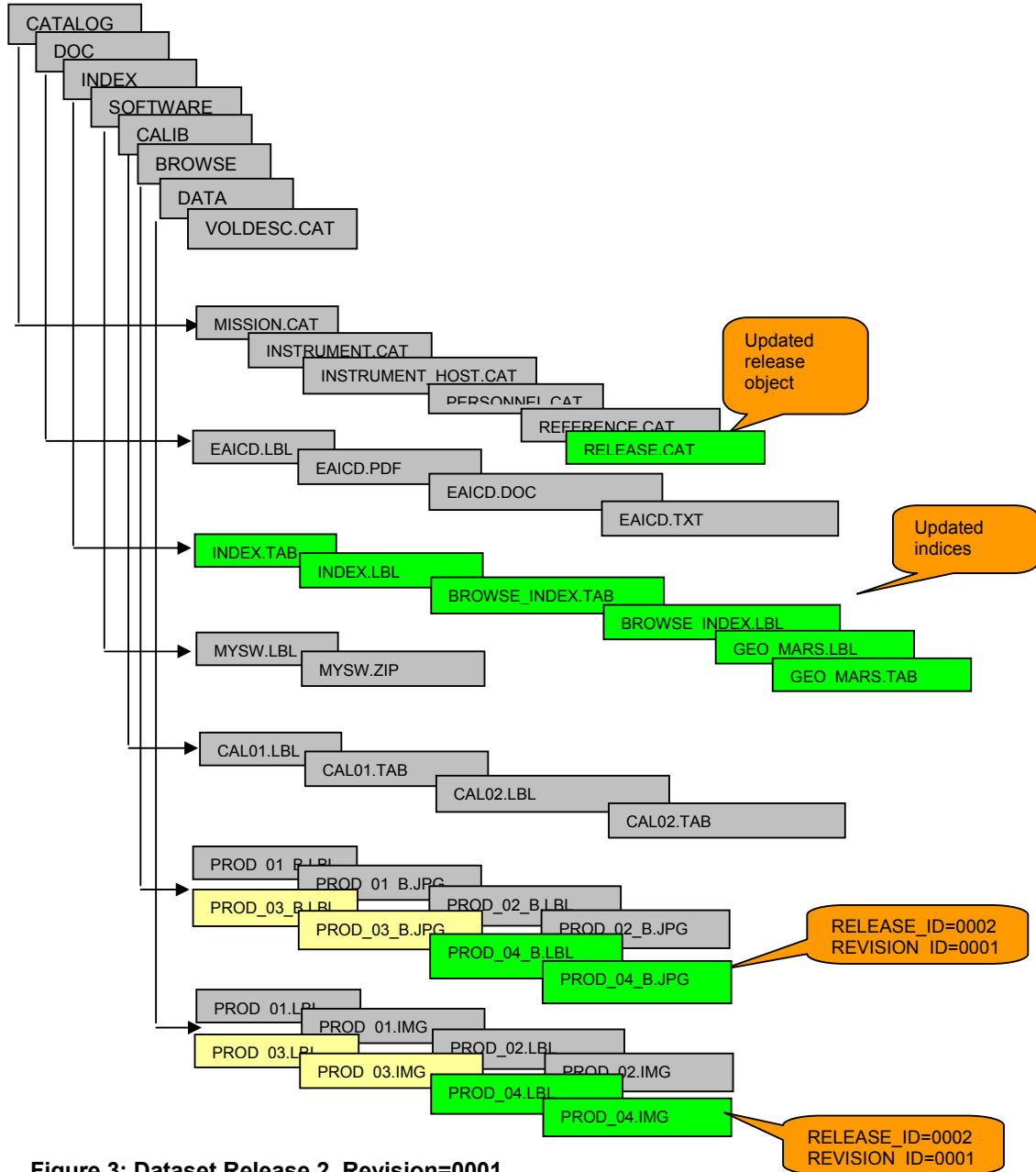


Figure 3: Dataset Release 2, Revision=0001



#### ***1.4.4 Updating Catalogue Information***

Catalogue files are ingested with the first release, first revision. Any further update on the catalogue information inside the PSA is done via the PSA administrator using a dedicated PSA administration tool. The data producer should contact his PSA contact directly and make sure that all catalogue information is properly ingested into the PSA.

The RELEASE.CAT is the only exception to this rule. You can use any release or revision ID values for the RELEASE.CAT, but we would recommend keeping it identical to the highest release and revision. If for example you have delivered RELEASE\_ID=0004, REVISION\_ID=0000, and you then decide you want to add a revision to RELEASE\_ID=0001, we recommend that the RELEASE.CAT values should remain RELEASE\_ID=0004, REVISION\_ID=0000.

#### ***1.4.5 Updating Documentation, Software, Geometry, Label, Extras and Calibration Directory Information***

Files that have been ingested in release 0001 must be updated by revisions to the release 0001. This normally covers the files in the following directories: GEOMETRY, DOC, LABEL, EXTRAS, CALIB and SOFTWARE.

For example, a new software version can be ingested via RELEASE\_ID=0001 and REVISION\_ID=0001. In this situation a user would have to:

- ❖ Update the software file and update the software label to include the RELEASE\_ID=0001 and REVISION\_ID=0001
- ❖ Update the INDEX.TAB and INDEX.LBL file (re-create these using the PVV)
- ❖ Update the DATASET\_RELEASE\_OBJECT for the first release in the RELEASE.CAT to include a new REVISION=0001 sub-object.
- ❖ Update the RELEASE\_ID and REVISION\_ID in the RELEASE.CAT. You are free to use any release or revision ID values in the RELEASE.CAT, but we strongly recommend using the highest release and revision values.

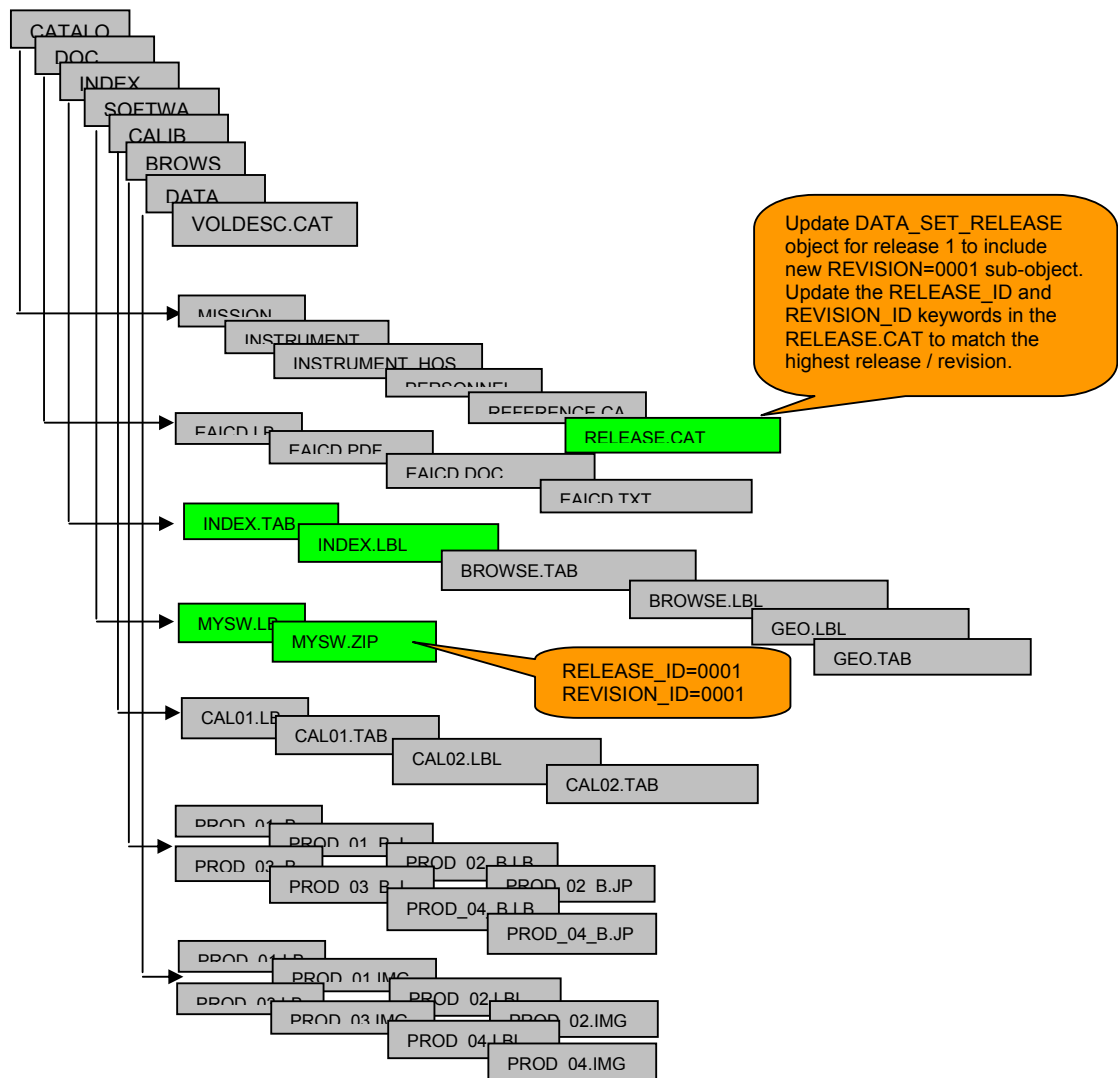


Figure 4: Dataset Release 0001, Revision 0001



## 2 Overview

### 2.1 Glossary: Archive Volume, Delivery, Data Set, Release, Revision

#### 2.1.1 Archive Volume

Within the PSA the *archive volume* represents an online delivery. This might not be fully compliant with the PDS standard 3.6, but the PSA itself can after the end of the mission produce fully PDS compatible archive volumes.

#### 2.1.2 Delivery

The term *delivery* is synonymous to the term *archive volume*. Deliveries are based on the concept of a *release* and a *revision*.

#### 2.1.3 Data Set

A *data set* is defined as in the PDS standard. For the PSA we restrict the PDS standard definition to “one volume – one dataset”.

#### 2.1.4 Data Set Release

A *data set release* (or volume release) contains data from a well-defined period of time. A data set release is made of at least one revision. The initial revision of a data set release is a fully PDS compatible volume including the VOLDESC.CAT and all necessary directories and their content.

Any further release is a concatenation of the new data to the already existing data set. One might consider the release concept as an incremental delivery of a data set in smaller data chunks.

#### 2.1.5 Data Set Revision

A data set revision contains the initial data of a data set release (REVISION\_ID is 0000) or it might comprise

- supplementary files, that shall be appended to the data set release
- updated files, that shall replace (overwrite) existing files in the data set release
- files, existing in the release, that shall be deleted from the data set release

For each update to a release, the REVISION\_ID will rise incrementally



## 2.2 Usage of the VOLDESC.CAT File

The VOLDESC.CAT file must contain a DATA\_SET\_RELEASE\_CATALOG pointer to a RELEASE catalog file. The DATA\_SET\_RELEASE objects within this catalog file must then fully define the releases and revisions.

The figure 5 gives an example of an initial VOLDESC.CAT file.

```
PDS_VERSION_ID          = PDS3.6
RECORD_TYPE             = STREAM
RELEASE_ID              = 0001
REVISION_ID             = 0000
OBJECT                  = VOLUME
  VOLUME_SERIES_NAME    = "MARS EXPRESS
  VOLUME_SET_NAME       = "MARS EXPRESS SPICAM"
  VOLUME_SET_ID         = FR_LSPI_SA_SPI
  VOLUME_NAME           = "MARS EXPRESS SPICAM-RAW VOLUME"
  PUBLICATION_DATE      = 2004-09-01
  VOLUMES                = 1
  MEDIUM_TYPE           = "ONLINE"
  VOLUME_FORMAT         = "ISO-9660"
  DATA_SET_ID          = "MEX-M-SPI-2-RAWEDR-V1.0"
  DESCRIPTION           = "THIS IS AN EXAMPLE OF A VOLDESC.CAT
  FOR THE SOLE PURPOSE OF DEMONSTRATING THE USE OF A RELEASE OBJECT"

OBJECT                  = DATA_PRODUCER
  INSTITUTION_NAME      = "SERVICE D'AERONOMIE CNRS"
  FACILITY_NAME         = N/A
  FULL_NAME             = UNK
  DISCIPLINE_NAME       = N/A
  ADDRESS_TEXT          = "91371 Verrieres le Buisson Cedex\n
  France"
END_OBJECT              = DATA_PRODUCER

OBJECT                  = CATALOG
  ^MISSION_CATALOG     = "MISSION.CAT"
  ^INSTRUMENT_HOST_CATALOG = "INSTHOST.CAT"
  ^INSTRUMENT_CATALOG  = "INST.CAT"
  ^DATA_SET_CATALOG     = "SPI_RAW_DS.CAT"
  ^PERSONNEL_CATALOG   = "PERSON.CAT"
  ^DATA_SET_RELEASE_CATALOG = "RELEASE.CAT"
  ^REFERENCE_CATALOG   = "REF.CAT"
  ^TARGET_CATALOG      = "NULL"
  ^SOFTWARE_CATALOG    = "NULL"
END_OBJECT              = CATALOG

END_OBJECT              = VOLUME
END
```

**Figure 5:**  
**VOLDESC.CAT example for release=0001, revision=0000**



## 2.2 The DATA\_SET\_RELEASE Object

The *data\_set\_release object* contains all *keyword-value* pairs that are necessary to identify a release and revision within this release.

This object is not PDS-compliant!

An initial release of the SPICAM data could be presented in the following catalog object:

```
PDS_VERSION_ID      = PDS3
LABEL_REVISION_NOTE = "TEST RELEASE CATALOG OBJECT"
RECORD_TYPE         = STREAM
RELEASE_ID          = 0001
REVISION_ID         = 0000
OBJECT              = DATA_SET_RELEASE
DATA_SET_ID         = "MEX-M-SPI-2-RAW-DATA-ARCHIVE-V1"
RELEASE_ID          = 0001
DESCRIPTION         = "
This release contains the digital numbers (DN) contained in the telemetry
(TM) packages of the SPICAM instrument on board of spacecraft Mars Express.
Data not been further converted or calibrated. This release contains data
from the cruise phase of the s/c including all data from the Near Earth
Verification (NEV) phase from July 2003 until December 2003. Data from
FIRST LIGHT (Earth Moon Scenario) and Interference Scenario are contained."

OBJECT              = REVISION
REVISION_ID         = 0000
REVISION_DATE       = 2003-01-01
REVISION_MEDIA      = "ONLINE"
DESCRIPTION         = "
First delivery of Mars Express SPICAM data:
RELEASE 0001, REVISION: 0000.
The data cover the period of the Near Earth Verification
Phase, 1 July 2004 till 1 November 2004. The value of
This delivery are the measurements of the following
Stars ...

"
END_OBJECT          = REVISION
END_OBJECT          = DATA_SET_RELEASE
END
```

Description of this release: This text is the entry point for the end user.

Description of this revision: This text is the entry point for the end user.

**Figure 8: Release and revision object within a release catalogue file**

The object revision is defining the necessary information about this revision. All description parts of one release will finally give the history information of a release.

In the DATASET.CAT, the value of the keyword archive status is set to **accumulating** to indicate that this volume is still growing.



A revision might be added for this release after improvement of the data within the release.

Here is an example of the release catalog object.

```
PDS_VERSION_ID      = PDS3
LABEL_REVISION_NOTE = "TEST RELEASE CATALOG OBJECT"
RECORD_TYPE         = STREAM
RELEASE_ID          = 0001
REVISION_ID         = 0001
OBJECT              = DATA_SET_RELEASE
DATA_SET_ID        = "MEX-M-SPI-2-RAW-DATA-ARCHIVE-V1.0"
RELEASE_ID         = 0001

DESCRIPTION         = "
  This release contains the digital numbers (DN) contained in the telemetry
  (TM) packages of the SPICAM instrument on board of spacecraft Mars Express.
  Data not been further converted or calibrated. This release contains data
  from the cruise phase of the s/c including all data from the Near Earth
  Verification (NEV) phase from July 2003 until December 2003. Data from
  FIRST LIGHT (Earth Moon Scenario) and Interference Scenario are contained."

OBJECT              = REVISION
REVISION_ID        = 0000
REVISION_DATE      = 2004-12-10
REVISION_MEDIA     = "ONLINE"
DESCRIPTION        = "
  First delivery of Mars Express SPICAM data:
  RELEASE 0001, REVISION: 0000.
  The data cover the period of the Near Earth Verification
  Phase, 1 July 2004 till 1 November 2004. The value of
  this delivery are the measurements of the following
  stars ....
"
END_OBJECT         = REVISION

OBJECT              = REVISION
REVISION_ID        = 0001
REVISION_DATE      = 2005-01-15
REVISION_MEDIA     = "ONLINE"
DESCRIPTION        = "
  First delivery of Mars Express SPICAM data:
  RELEASE 0001, REVISION: 0001.
  Due to corrections of the time correlation packages and
  corresponding updates of the SPICE kernels, all files later
  than 12-10-2004T14:00:00 were updated.
"
END_OBJECT         = REVISION

END_OBJECT         = DATA_SET_RELEASE

END
```

Keep revision=0000 sub-object !

Description of this release!

Figure 9: DATA\_SET\_RELEASE object with two revision sub-objects.





The next example shows a release object of a second release.

```
PDS_VERSION_ID          = PDS3
LABEL_REVISION_NOTE     = "TEST RELEASE CATALOG OBJECT"
RECORD_TYPE             = STREAM
RELEASE_ID              = 0002
REVISION_ID            = 0000

OBJECT                  = DATA_SET_RELEASE
DATA_SET_ID            = "MEX-M-SPI-2-RAW-DATA-ARCHIVE-V1.0"
RELEASE_ID             = 0002

DESCRIPTION             = "
    This release contains the digital numbers (DN) contained in the telemetry
    (TM) packages of the SPICAM instrument on board of spacecraft Mars Express.
    Data not been further converted or calibrated. This release contains data
    from orbit 11 up to orbit 234."

OBJECT                  = REVISION
REVISION_ID            = 0000
REVISION_DATE          = 2004-09-01
REVISION_MEDIA         = "ONLINE"
DESCRIPTION            = "Original revision."
END_OBJECT             = REVISION

END_OBJECT              = DATA_SET_RELEASE

OBJECT                  = DATA_SET_RELEASE
DATA_SET_ID            = "MEX-M-SPI-2-RAW-DATA-ARCHIVE-V1.0"
RELEASE_ID             = 0001
ARCHIVE_STATUS         = "ACCUMULATING"

DESCRIPTION             = "
    This release contains the digital numbers (DN) contained in the telemetry
    (TM) packages of the SPICAM instrument on board of spacecraft Mars Express.
    Data not been further converted or calibrated. This release contains data
    from the cruise phase of the s/c including all data from the Near Earth
    Verification (NEV) phase from July 2003 until December 2003. Data from
    FIRST LIGHT (Earth Moon Scenario) and Interference Scenario are contained."

OBJECT                  = REVISION
REVISION_ID            = 0000
REVISION_DATE          = 2003-01-01
REVISION_MEDIA         = "ONLINE"
DESCRIPTION            = "Original revision"
END_OBJECT             = REVISION

END_OBJECT              = DATA_SET_RELEASE

END
```

Figure 10: DATA\_SET\_RELEASE and revision object within a release catalogue file.  
Release number 2.



## 2.3 The INDEX Directory and INDEX Files

All index files below the INDEX directory within a data set release are unique and are valid for the actual data set and all previous ones. This means that the content of an index file covers all previous data set releases. A default set of index files are:

- INDEX.TAB and INDEX.LBL; the main index file
- BROWSE\_INDEX.TAB and BROWSE\_INDEX.LBL; the browse/quicklook index file
- GEO\_XXX.TAB and GEO\_XXX.LBL; the geometry index file (where XXX is the reference target body)

The PVV tool can be used to produce the main or browse index files.

The index label files shall contain the keywords RELEASE\_ID and REVISION\_ID, set to the latest RELEASE\_ID and within this release to the latest REVISION\_ID.

The index table file shall contain two columns named RELEASE\_ID and REVISION\_ID. Each entry in the index table - each data product – shall contain the correct RELEASE\_ID and REVISION\_ID of the data product.

The entire contents of the INDEX directory will be sent with each release or revision delivered to the PSA. The 'tar.gz' file created using the 'pvv build' command will always include these files by default.

If you use the 'pvv freeze' command and manually generate the delta archive for delivery to the PSA you must ALWAYS include the entire INDEX directory, even if nothing except the main index table has changed (Section 4.4).

## 2.4 Content of the EXTRAS Directory

The files below the EXTRAS directory do usually not follow strict PDS Standard rules. They might e.g. not have proper label files with each data product. Especially, there might be no mentioning of the RELEASE\_ID and REVISION\_ID keyword. Therefore all files below the EXTRAS directory need special treatment.

With each release, all files within the EXTRAS directory of this release will supersede any previous content of the EXTRAS directory that has been ingested into the PSA before. With each new release, all files below the EXTRAS directory need to be transmitted again to the PSA archive team and need to be (re)ingested again into the PSA database.

The 'tar.gz' file created using the 'pvv build' command will always include the files below the EXTRAS directory by default.

If you use the 'pvv freeze' command and manually generate the delta archive for delivery to the PSA you must ALWAYS include the entire EXTRAS directory, even if nothing has changed (Section 4.4).

## 2.5 The \*INFO.TXT Files

The \*INFO.TXT files throughout the data set do not normally follow PDS rules. Especially, they do not necessarily need a label file, and therefore they need special handling.

The \*INFO.TXT files contained within each release will supersede any existing \*INFO.TXT files that are already ingested in the PSA database. All the \*INFO.TXT files will therefore be retransmitted and (re-) ingested into the PSA database (the 'pvv build' command will do this by



default).

It is the responsibility of the data producer to ensure the correctness of the \*INFO.TXT files. The PSA user interface will contain end user information to ensure that end users of the PSA understand properly how these files are handled.

The 'tar.gz' file created using the 'pvv build' command will always include the \*INFO.TXT files below their respective directory by default.

If you use the 'pvv freeze' command and manually generate the delta archive for delivery to the PSA you must ALWAYS include every \*INFO.TXT file underneath its respective directory, even if nothing has changed (Section 4.4).

### 3. General Rules

- release and revision id numbers shall have a length of 4
- release\_id numbers range from [0001:9999]
- revision\_id numbers range from [0000:9999]
- index.lbl/tab are created by the PVV tool
- browse.lbl/tab are created by the PVV tool
  
- the label part of **all** index files must contain the RELEASE\_ID and REVISION\_ID keywords
- **all** index files shall contain two columns, one for RELEASE\_ID and one for REVISION\_ID
  - these two columns are not used by the PVV but might be important for the end user ;
  - this is also applicable for user defined index files
- the data producer must keep her/his whole data set on a single file-system
- **all** label files must contain the RELEASE\_ID and REVISION\_ID keyword
  - wherever they are located
  - whatever type they are (attached, detached, combined detached)
- there shall be one release.cat file on a dataset; this release.cat file contains the actual release information and the history of previous release information;
- the release catalogue file shall contain only one release object for each release
- each release object can contain one or more revision sub-objects
- all catalogues except the release.cat will always be RELEASE\_ID=0001, and any updates must be made via a revision by special request to your Mission Archive Manager.

### 4. The PVV tool and the release object: a quick guide

In this section a few user case scenarios are presented showing in parallel the usage of the PVV verification tool. For details of the PVV, please refer to the PVV User Manual. There are some basic rules you must follow in order for the PVV to successfully run on a release dataset:

- You must construct your dataset *incrementally*. In other words, if you already start testing using data from several Releases, you **must** first construct the dataset for Release=0001 and PVV it, then add Release=002 and PVV it etc., until the complete dataset is tested, up to the current Release.
- All Revisions made to a single Release are cumulative. In other words:
  - i. In Release0001, Revision0000 of a dataset: Product X is updated and



delivered as Release0001 Revision0001

- ii. In Release0001, Revision0001 of the dataset: later updates to different data (e.g Product\_Y or Document\_Z). Although this is the first revision of these specific products, it is the second revision to this Release, and so these should be delivered as Release0001, Revision0002.
- If a new document or 'non-DATA' file is to be delivered (i.e. a file not in the DATA, CALIB or BROWSE directory), it should be delivered as part of the latest Release. For example, if you are ready to deliver Release 0004 of a dataset, and you have also decided to include a new document to the DOCUMENT directory, this document should be delivered as Release 0004 as well, even though it is the first delivery of this document.
- Similarly, if a new 'non-DATA' file is to be added between release deliveries, it should form the next Revision of the latest Release. For example, if you have delivered Release 0004 Revision 0000, and you now want to add a new file to the dataset, you can do so by making this product Release 0004 Revision 0001.

[N.B. The DOCUMENTS directory on the online database will not take account of release or revision IDs in the labels. The latest copies of ALL documents will always be delivered to the end user from the online PSA archive. Using the Release ID in the above example simply allows the PVV to check the file as part of the overall delivery].



#### 4.1 Preparation and Ingestion of Release 0001/Revision 0000

PVV command line	Remarks
(1)> cd DATA_SET_ROOT_DIRECTORY	to 1:
(2)> vi VOLDESC.CAT	to 2: verify that the keywords RELEASE_ID=0001 and REVISION_ID=0000 are in the VOLDESC.CAT file; these settings will never change (unless you modify the VOLDESC.CAT itself)
(3)> pvv scan -Drelease=1	to 3: build an XML image of the entire data set. This means that if you have already data products from a higher release in this dataset, they need to be correct, because they are taken into account during the scan;
(4)> pvv index	to 4: creates the files ./INDEX/INDEX.TAB and ./INDEX/INDEX.LBL containing information of ALL available products below the data set root directory, regardless of the release.
(5)> pvv verify -Drelease=1	to 5: verification of doc, index, calib, extra, data, etc directories and all data products contained there-in. A cross-check between the data products and the catalog files is also completed;
(6a)> pvv build -Drelease=1	to 6a: builds a tar.gz file with only those files relevant to release1 (so in this case, the entire data set). This command also freezes release 1 of the xml image file. This means that all information relating to release 1 in the xml image file are protected from overwriting. Be aware, that if you change anything in a data product belonging to the release 1 after you have frozen the XML file, this will not be taken into account and you will receive errors the next time you run the pvv verify command.
(6b)> pvv freeze -Drelease=1	To 6b: Simply freezes you XML file for release1, after which you will have to MANUALLY construct the delta archive for delivery to the PSA (Section 4.4)
(7a)> pvv upload FILENAME	to 7a: the pvv upload can be used to transfer the file created in step (6) to the PSA.
(7b)> ftp psa.sciops.esa.int ...	to 7b: alternatively, you can use your preferred ftp client to transfer the data to the PSA



#### 4.2 Preparation and Ingestion of Release 0002/Revision 0000

PVV command line	Remarks
(1)> cd DATA_SET_ROOT_DIRECTORY	to 1:
(2)> vi CATALOG/RELEASE.CAT	to 2: verify that the object DATA_SET_RELEASE object for release nr 0002 is available ; also update the keywords RELEASE_ID and REVISION_ID in the label of this catalog file to match the highest release / revision delivered to the PSA.
(3)> pvv scan -Drelease=2	to 3: append to the XML image of the data set the information of release nr 2.
(4a) rm INDEX/INDEX.LBL (4b) rm INDEX/INDEX.TAB (4c)> pvv index	to 4: a, b) delete the already existing INDEX files c) creates the files ./INDEX/INDEX.TAB and ./INDEX/INDEX.LBL containing information of ALL available products below the data set root directory, regardless of the release.. For any other index files in the INDEX directory, you have to ensure that the keyword RELEASE_ID is set to the highest release number and that the REVISION_ID is set to 0000.
(5)> pvv verify -Drelease=2	to 5: verification of doc, index, calib, extra, data, etc directories and all data products of release nr 2; at the end a cross-check between the data products and the catalog files is completed; N.B. Any change you have made to any product contained in release 1 (already delivered to the PSA) will cause an error in this step (please see user case 4.3)
(6a)> pvv build -Drelease=2	to 6a: constructs a 'tar.gz' file containing only those files from release2, and freezes the release 2 of the xml image file;
(6b)> pvv freeze -Drelease=2	to 6b: freezes the XML file for release2. You will then have to manually construct the delta archive for release2 before delivering to the PSA (Section 4.4).
(7a)> pvv upload FILENAME	to 7a: you can use the 'pvv upload' command to transfer the release2 tar.gz file created in step 6 to the PSA.



<p>(7b)&gt; ftp psa.sciops.esa.int ...</p>	<p>to 7b: alternatively, you can use your preferred ftp client to transfer the data to the PSA</p>
--	--

### 4.3 Preparation and Ingestion of Release 0001/Revision 0001

Let's assume one of the data products in release one is updated after the delivery of release 2.

PVV command line	Remarks
(1)> vi your_product	to 1: check that the label of the altered or new data product contains the keywords RELEASE_ID=0001 and REVISION_ID=0001
(2)> pvv label your_product	to 2: verify the label of your data product file
(3)> cd DATASET_ROOT_DIRECTORY	to 3: go to your dataset root directory
(4)> vi CATALOG/RELEASE.CAT	to 4: update your RELEASE.CAT file such that it contains the new sub-object REVISION = 0001 within the data_set_release object 1. Also, update the keywords RELEASE_ID and REVISION_ID to match the highest release / revision delivered to the PSA.
(5)> pvv labels -Drelease=1 -Drevision=1	to 5: verify all the data products from your revision 1 of release 1
(6a)> rm INDEX/INDEX.LBL (6b)> rm INDEX/INDEX.TAB (6c)> pvv index	to 6: clean up the existing index files and re-run the pvv index command. Do this also for any other index files, if required.
(7)> pvv verify -Drelease=1 -Drevision=1	to 7: verify your release 1 and revision 1
(8a)> pvv build -Drelease=1 -Drevision=1	to 8a: construct a 'tar.gz' file containing only those files relevant to the revision, and freeze the XML file.
(8b)> pvv freeze -Drelease=1 -Drevision=1	to 8b: Freeze your XML file and MANUALLY construct the delta archive containing the release1 revision1 information for upload (Section 4.4).
(9)> pvv status	to 9: check what the status of your data set is
(10)> pvv status -Drelease=1 -Drevision=1	to 10: check which data products have been transferred (using the 'pvv build' or 'pvv freeze' command) in release 1, revision 1. Note that a release or revision status of 'transferred' only means that the files were readied for transfer using the 'pvv build' or 'pvv freeze' command. It does not mean they have been uploaded onto the PSA. N.B. you can use the pvv status command to check the files that have been scanned, verified or transferred (using 'pvv build') in





<pre>(11a)&gt; pvv upload RELEASE1_REVISION1_FILENAME</pre> <pre>(11b)&gt; ftp psa.sciops.esa.int</pre>	<p>each release and revision. This can help you keep track of your progress.</p> <p>to 11a: You can use the 'pvv upload' command to transfer your release1 revision1 file generated in step 8 to the PSA.</p> <p>to 11b: Alternatively, you can use your preferred ftp client to transfer your data to the PSA.</p>
--	---

**IMPORTANT NOTE:**

The XML data set image file contains the creation time of the individual files of your dataset as time stamps after the 'pvv build' or 'pvv freeze' command. In case of moving data products in and out of a file system, the user therefore has to maintain the creation times of all the files involved.

On Solaris e.g. the 'mv' command preserves the creation time of a file and should be used instead of making a copy using the 'cp' command.

**4.4 Preparation and Ingestion of Release 0001/Revision 0001**

If you are using the release concept [AD6] and have decided not to use the 'pvv build' command to automatically generate a delta archive, then you will have to freeze your archive using the 'pvv freeze' command and then manually construct your own delta archive to transfer to the PSA. In order for the PSA to successfully ingest this into your existing data set structure, you will need to ensure that the correct files are in the delta archive you generate. The following sections describe the requirements for this.

**4.4.1 Manually constructing your first Release [Release0001 Revision0000]**

For your first release, you will need to pack the entire data set together into a single file (e.g. tar.gz or tar.bz2). This will then have to be manually transferred to the PSA using your preferred ftp client.

**4.4.2 Manually constructing your second Release [Release0002 Revision0000]**

For your second release, you will need to pack the following files together into a single file (e.g. tar.gz or tar.bz2):

- All products and product labels from Release2
- **All** \*INFO.TXT files (e.g. CALINFO.TXT, DOCINFO.TXT etc)
- The **entire** INDEX directory (even if nothing has changed)
- The **entire** EXTRAS directory (even if nothing but the main index has changed)
- The RELEASE.CAT file updated with information for the new release / revision

You should maintain the directory structure for all files packed together. This delta archive will then have to be manually transferred to the PSA using your preferred ftp client.



#### ***4.4.3 Manually constructing a Revision [e.g. Release0001 Revision0001]***

For any revision to a release, you will need to pack the following files together into a single file (e.g. tar.gz or tar.bz2):

- All products and product labels from Release1 Revision1
- **All** \*INFO.TXT files (e.g. CALINFO.TXT, DOCINFO.TXT etc)
- The **entire** INDEX directory (even if nothing but the main index has changed)
- The **entire** EXTRAS directory (even if nothing has changed)
- The RELEASE.CAT file updated with information for the new release / revision

You should maintain the directory structure for all files packed together. This delta archive will then have to be manually transferred to the PSA using your preferred ftp client.

## 5. Frequently Asked Questions

- Q1: I have updated a file and want to submit a new revision (existing release, revision+1). What INDEX files do I need to prepare.
- A1: You have to prepare all index files that are affected by your revision. This is at least the INDEX.TAB/LBL file. If you are using a BROWSE directory, you might need to update the BROWSE\_INDEX.TAB/LBL file. Your INDEX.LBL shall contain the two keywords RELEASE\_ID and REVISION\_ID. The INDEX.TAB shall contain two columns (RELEASE\_ID, REVISION\_ID) specifying the release and revision of each product. All index files are transferred with each release or revision delivery.
- Q2: I want to delete a file from a previous release. How do I do this?
- A2: This mechanism is not yet defined. Needs to be done. Contact your PSA mission contact point
- Q3: How and where can I define the end of the proprietary period for a data set release that I deliver to the PSA before the end of the proprietary period?
- A3: There is no automatic definition (speak within the release object e.g.) of the proprietary period. The Planetary Science Archive staff will maintain this information via an administration tool.
- Q4: Can I point to a text file from within my revision object?
- A4: Indeed, you can. E.g. the following line identifies an ASCII description of release=0001 and revision=0000 within the catalogue directory:  
" ^DESCRIPTION = REVISION\_0001\_0000.TXT"
- Q5: Version 1.5 of this document did contain a whole chapter on the index file creation. Why is all the information gone?
- A5: The pre-1.6 version was based on the assumption that a revision is defined by the content of an index file. This means that an update of a file A had to be defined by a line in the main index table that described the update. This mechanism was found to be too complicated. From version 1.7 of this document the assumption is that each label file contain the two keywords RELEASE\_ID and REVISION\_ID and that herewith the PVV tool can identify all releases and revisions. The PVV cannot only identify the latest revision, but also all previous revisions of all releases. If a complete crash should happen at the data producers institute, the PSA can reproduce all previous releases and revisions and help the data producer to re-install a valid data set at the producers institute.
- Q6: There are no label files in my EXTRAS directory. How can I deliver and update data in the EXTRAS directory?
- A6: All files below the EXTRAS directory needs to be updated for each release. Please see section 2.4 or contact your PSA archive scientist.
- Q7: Can I update the software within the SOFTWARE directory using the release-revision concept?
- A7: Yes, via the release=0001 and the next revision.
- Q8: This document says that the PVV creates the INDEX.TAB/LBL index file. Why can't I create the index files on my own?
- A8: You can create the index files by your own. Running the pvv index command will simply



throw an error saying that the index file already exists. The PVV created index file is the one required by the PDS standard plus

- the two keywords RELEASE\_ID and REVISION\_ID in the index label
- the two columns release\_id and revision\_id that identify for each entry the release and the revision

Q9: How do I update a catalogue file?

A9: Prepare your catalogue file, email it to your mission PSA contact at ESTEC and get in direct contact with her/him. The PSA administrator will ingest your catalogue file via a special administration tool and you should check the results by querying the PSA and comparing the catalogue files received with the ones you intended to update.

Q10: What from all of this will the end user see when accessing the data via the Planetary Science Archive?

A10: The end user – after having located data of interest – will be offered the following options:

- o Get the full data set (collection of all releases)(the content of all directories)
- o Get only the data for a time period of interest
- o Get the data for a time period of interest plus all non-data directories. This will be a fully PDS compatible data set.

If a user wants to retrieve the data ingested into the PSA from a special release, he can build up his search query using the RELEASE\_ID and REVISION\_ID keywords and retrieve the resulting data products.