

# Xman and xcheck Operations Guide (PDS Version)

*J. Kodis, P. McCaslin*

DRAFT

September 29, 2008

## OVERVIEW

The xman program generates XML-formatted manifests of Submission Information Packages (SIPs) to be delivered to NSSDC. Xman generated manifests convey:

- Global SIP identifying and descriptive information
  - Information about Transfer Objects within the SIP. (PDS SIPs will have only one Transfer Object containing the contents of a single PDS volume)
  - Information about individual files with the Transfer Object, including checksums
- See Appendix A for the XML schema of the manifest.

The xcheck program validates and extracts information from an xman-generated manifest files.

## ***XMAN* Installation**

The xman package is distributed in two forms: a setup.exe file for use on Microsoft platforms, and a source distribution kit for use on all other platforms.

On Microsoft platforms, the *xman* program is distributed in a standard *setup.exe* file which when run will offer the user a choice of the directory into which the *xman* program should be installed. After installation, the *xman* manifest generator program can be run from the XP command line shell, *cmd.exe*.

On all other platforms, the source code is extracted from the xman tar.gz file and built using the usual `./configure; make; make check` commands to collect system information, build the xman and xcheck programs, and run the regression tests. Building the programs will require a C compiler, its support libraries, and the Expat XML parsing library. On Apple OS-X platforms, these are supplied by the Xcode tools from the operating system installation disk; On Red Hat, Fedora, Centos, and other Red Hat derived operating systems, these can be installed using `"yum groupinstall 'Development Tools'; yum install expat-devel"`; On Solaris, these tools are available from the Sun Freeware collection at [www.sunfreeware.com](http://www.sunfreeware.com).

The manifest generation program, *xman.exe*, is a command-line driven program. To simplify the installation and operation of the program, all files in the *xman* distribution should be unpacked into a single directory, the program should be run from this directory, and the program should have permission to write the manifest and log file that will be generated into this directory. This imposes a requirement that the current working directory have enough space to hold the entire manifest,

which will accrue at a rate of about a kilobyte per file being recorded.

## ***XMAN* CONFIGURATION**

Two configuration files are required for the operation of the *xman* program: *producer-id.tsv*, which identifies the producer running the *xman* program, and *id-map.tsv*, which maps producer and volume identifiers to a producer archive product identifier. These configuration files should be placed in the same directory from which the *xman* program will be run.

The *producer-id.tsv* files are node-specific and need only be installed once.

The *id-map.tsv* file is shared by the data nodes. This file is updated every time a new PDS Producer Archive Project is registered at NSSDC. A data node must re-install the *id-map.tsv* when notified by NSSDC that a Producer Archive Project submitted by that node has been registered. Additional information concerning registration of Producer Archive Project is in the document [Submission of Large Data Sets to NSSDC](#).

The *producer-id.tsv* configuration files and the up-to-date *id-map.tsv* file available are accessible on-line at <http://nssdcftp.gsfc.nasa.gov/PDS>.

## ***XMAN* OPERATION**

After installation, the *xman* manifest generator program can be run from the XP command line shell, *cmd.exe*, on Microsoft platforms, or from the bash, tcsh, or any other standard command shell on any other platform. From the shell prompt, change the current working directory to the directory containing the *xman* program and its support files. The *xman* program is then run from this directory and supplied with a single argument: the top level directory of the volume to be recorded in the manifest. Once started, the *xman* program will run to completion with no further interaction. While the run time depends on a wide range of factors, several hours of processing time should be allocated for each terabyte of data to be processed. See the *xman* and *xcheck* MAN pages for additional information.

## **INPUTS**

The main input to the *xman* program is the path to the top-level directory of the volume to be recorded in the generated manifest file. The only other input normally supplied to the *xman* program is the `--comment=<message>` command line option. This option provides a way to record a small item of manually generated information about a data volume in the manifest. The message string supplied to this option must be a short string of printable UTF-8 characters.

**The Data Volume:** Because the manifest schema specifies a UTF-8 character set encoding, all directory names and file names must be valid UTF-8 character sequences. Due to limitations on the size of the magnetic tapes currently used for archival storage of the data, no data file may exceed 300 GB in size.

**Volume Description:** The top level of the volume for which a manifest is to be generated must contain a properly formatted ODL *VOLDESC.CAT* file. This file must contain an "OBJECT = VOLUME" block containing VOLUME\_ID parameter/value pair. Omission of this parameter will result in a fatal error condition.

## OUTPUTS

If no errors are encountered during the execution of the *xman* program, the only output displayed will be a summary listing the number of files processed, their volume, and the time required. A typical program run might look like:

```
C:\Documents and Settings\jmkodis\xman-xp>xman.exe g:  
xman: 462 files, 654153487 bytes in 417.671 seconds at 1.566 MB/sec
```

Along with this will be the primary output of the *xman* program, an XML-formatted manifest of the data volume and a log file documenting the execution of the *xman* program. Any warning or error messages will be recorded in this file, along with the version information for the *xman* program, the start and stop times of the manifest generation run, the total file count, byte count, run time, and data rate achieved during the program's execution. Unless specified otherwise, these files will be names Sip-manifest-<volume-id>.xml and Sip-manifest-<volume-id>.log.

The manifest.xml file will be as described in version 0.13 of the AII-SIPManifest.xsd, included as Attachment A. The *xman* program only records information on directories and regular data files. Any other file system objects encountered are silently ignored.

## TRANSFER MEDIA PREPARATION

Data are copied to removeable disk for delivery to NSSDC. USB disks up to 2 TB in size are acceptable. Increases in the 2 TB size limit will be negotiated as larger disks become available. Disk must be formatted using the Linux ext2 or ext3 file system, the Microsoft FAT file system, or the Macintosh Extended or Extended Journaled file system.

The preparation of a data volume for transfer to the NSSDC on removable disk involves generating a manifest, manually deciding how to partition the volume among the number of drives required to hold all the data, and then copying the data and a copy of the *VOLDESC.CAT* file for the volume to each removable drive. In making this copy, the contents of the volume to be transferred should be

held constant so that the contents of the removable media will stay in sync with the manifest.

If the contents of the volume requires more than a single removable disk, the volume can be split across removable disks in whatever organization is most convenient for the originating node, with a few restrictions: the volume directory structure must begin at the top-most directory of each removable disk, and the directory structure of the original volume must be maintained as it is copied to the removable disks, with the exception that the top-level directory on each removable disk must have a copy of the *VOLDESC.CAT* file associated with the volume being delivered.

As an example, imagine that a scientific mission known as the Bongo Fury experiment has organized their results into a *level-0* directory to hold the raw data, with the higher order products placed in a *derived* directory holding *level-2* and *level-3* subdirectories, and stored the entire volume on the system's *G:\* drive. The top few levels of the volume might be:

```
G:\
  bongo-fury\
    VOLDESC.CAT
    CATALOG\
    level-0\
    derived\
      level-2\
      level-3\
```

One reasonable way to distribute this data volume across two removable disks mounted as drives *J:\* and *K:\* would be:

```
J:\
  VOLDESC.CAT
  level-0\

K:\
  VOLDESC.CAT
  CATALOG\
  derived\
    level-2\
    level-3\
```

These drives along with the manifest describing the data that they hold can then be shipped to the NSSDC. The contents of the removable disk drives will be compared to the manifest prior to storage in the permanent archive.

## Appendix A: SIP Manifest Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSpy v2005 rel. 3 U (http://www.altova.com) by John Garrett (NASA Goddard
Space Flight Center) -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
```

```

xmlns="urn:us:gov:nasa:nssdc:schema:sipmanifest:v0.13"
targetNamespace="urn:us:gov:nasa:nssdc:schema:sipmanifest:v0.13"
elementFormDefault="unqualified" attributeFormDefault="unqualified">
  <xs:complexType name="ParameterRangeType">
    <xs:annotation>
      <xs:documentation>This type is used to express the range of parameter
values covered for this object. This may be used for access to the data.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="TimeRange" minOccurs="0">
        <xs:annotation>
          <xs:documentation>Range of the event time (in UTC) in
this object. Since these values are used for searching and since we want a common format for
that searching, we will use the full XML dateTime format for the type. For times that are
not accurate to the second, beginTime and EndTime would be set to included the entire range
that could be covered by the less precise time range. For example, if range is from
2007-07-04 UTC to 2007-11-27 UTC, beginTime = "2007-07-04T00:00:00 UTC" and endTime =
"2007-11-27T23:59:59 UTC" </xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:sequence>
            <xs:element name="BeginTime" type="xs:dateTime"/>
            <xs:element name="EndTime" type="xs:dateTime"
minOccurs="0"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="OrbitRange" minOccurs="0">
        <xs:annotation>
          <xs:documentation>Span of orbits in this object</
xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:sequence>
            <xs:element name="BeginOrbit"
type="xs:nonNegativeInteger">
              <xs:annotation>
                <xs:documentation>First orbit
included in this object. (Note: Number of the very first orbit for this spacecraft (0 or 1)
follows the converntion used for the spacecraft.</xs:documentation>
              </xs:annotation>
            </xs:element>
            <xs:element name="EndOrbit"
type="xs:positiveInteger" minOccurs="0"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ChecksumType">
    <xs:annotation>
      <xs:documentation>Overall checksum for all content at this level for
this SIP instance. (Note: Producer and Archive need to coordinate on the meaning of (i.e.,
how to calculate) this checksum if directories or more than one file are involved. </
xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="ChecksumMethod">
        <xs:annotation>
          <xs:documentation>Name or ID of the checksum algorithm
used (i.e., not the actual algorithm). </xs:documentation>
        </xs:annotation>
        <xs:simpleType>

```

```

        <xs:restriction base="xs:string">
            <xs:enumeration value="CRC32"/>
            <xs:enumeration value="MD5"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element name="ChecksumValue" type="xs:string"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="GroupType">
    <xs:sequence>
        <xs:element name="GroupTypeID" type="xs:string">
            <xs:annotation>
                <xs:documentation>An ID unique within the Producer-
Archive Project which identifies the type of this Group instance. This ID was defined during
data modeling that is part of the Submission Agreement. [PAIS WB0.8 ASM:
transferObjectGroupID]</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="GroupID" type="xs:string">
            <xs:annotation>
                <xs:documentation>An ID unique within the combined
context of the parent objects (any Transfer Object Group Types and the Transfer Object Type)
containing this group object, the Producer Site, and the Producer-Archive Project. This
identifies this Group instance. (Could be Directory name.) [PAIS WB0.8 ASM:
transferObjectGroupName TBD?]</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="ParameterRange" type="ParameterRangeType"
minOccurs="0">
            <xs:annotation>
                <xs:documentation>Values for attributes that could be
used to discover the data content included in this group (directory) instance. [PAIS WB0.8
ASM: ANY]</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="NumberOfFilesIncluded" type="xs:positiveInteger"
minOccurs="0">
            <xs:annotation>
                <xs:documentation>Total of all actual files (i.e.,
directories are not counted) in the payload of this Group instance. [PAIS WB0.8 ASM: ANY]</
xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="Group" type="GroupType" minOccurs="0"
maxOccurs="unbounded">
            <xs:annotation>
                <xs:documentation>Instances of individual groups
(directories or directory trees) that are part of this Group instance. If 0 groups are
included, the current Group instance is the bottom group in the current group tree. [PAIS
WB0.8 ASM: transferObjectGroup (0..N)]</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="File" minOccurs="0" maxOccurs="unbounded">
            <xs:annotation>
                <xs:documentation>Instances of individual files that are
part of this Group instance. If 0 files (and 0 groups) are included, this indicates an
existing but empty group (directory). [PAIS WB0.8 ASM: dataObject (0..N)]</xs:documentation>
            </xs:annotation>
        </xs:complexType>
        <xs:sequence>
            <xs:element name="DataObjectTypeID"
type="xs:string">
                <xs:annotation>

```

```

                                <xs:documentation>An ID unique
within the combined context of the Transfer Object Group Type, the Transfer Object Type, the
Producer Site, and the Producer-Archive Project which identifies the type of the Data Object
(File). [PAIS WBO.8 ASM: dataObjectTypeID (1..1)] [NOTE: JGG added this new
element since it was a required PAIS WBO.8 ASM element and was not included in this schema.
TBD - Do we want to keep this?]</xs:documentation>
                                </xs:annotation>
                                </xs:element>
                                <xs:element name="FileLocation" type="xs:anyURI">
                                <xs:annotation>
                                <xs:documentation>URI where this
file can be found. NOTE: the URI serves as a unique Data Object (File) Instance Identifier.
[PAIS WBO.8 ASM: dataObjectLocation 1.1]</xs:documentation>
                                </xs:annotation>
                                </xs:element>
                                <xs:element name="ParameterRange"
type="ParameterRangeType" minOccurs="0">
                                <xs:annotation>
                                <xs:documentation>Values for
attributes that could be used to discover the data content included in this file instance.
[PAIS WBO.8 ASM: ANY]</xs:documentation>
                                </xs:annotation>
                                </xs:element>
                                <xs:element name="Checksum" type="ChecksumType"
maxOccurs="unbounded">
                                <xs:annotation>
                                <xs:documentation>Checksum Value(s)
and associated Checksum Method (i.e., type of checksum) for this file [PAIS WBO.8 ASM: ANY]</
xs:documentation>
                                </xs:annotation>
                                </xs:element>
                                <xs:element name="Format" minOccurs="0"
maxOccurs="unbounded">
                                <xs:annotation>
                                <xs:documentation>Identification of
the format of this file. Multiple independently useable format identifications are allowed.
Each identification is made by supplying the Authority or system in which the format
registered and the ID for the format within that system. [PAIS WBO.8 ASM: ANY]</
xs:documentation>
                                </xs:annotation>
                                <xs:complexType>
                                <xs:sequence>
                                <xs:element name="Authority"
type="xs:string">
                                <xs:annotation>
                                <xs:documentation>An identifier of a system of format identifiers. Examples: IETF MIME types
or CCSDS Control Authority (i.e. ADID assignments) [PAIS TransferObjectDescriptor (TOD) uses
registrationAuthority]</xs:documentation>
                                </xs:annotation>
                                </xs:element>
                                <xs:element name="FormatID"
type="xs:string">
                                <xs:annotation>
                                <xs:documentation>The ID assigned to the data object format with the system of format
identifiers identified by Authority [PAIS TOD uses registeredID]</xs:documentation>
                                </xs:annotation>
                                </xs:element>
                                </xs:sequence>
                                </xs:complexType>
                                </xs:element>
                                <xs:element name="FileSize"
type="xs:nonNegativeInteger" minOccurs="0">

```

```

                <xs:annotation>
                    <xs:documentation>Size of the file
in bytes. (Zero size file allowed.) [PAIS WB0.8 ASM: ANY]</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:any namespace="##other" minOccurs="0"
maxOccurs="unbounded" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded" />
</xs:sequence>
</xs:complexType>
<xs:complexType name="TransferObjectType">
    <xs:sequence>
        <xs:element name="TransferObjectReplacedID" type="xs:string"
minOccurs="0">
            <xs:annotation>
                <xs:documentation>An ID that matches a previously
submitted Transfer Object ID. The information contained within this current Transfer Object
instance will replace within the Archive the Transfer Object instance information identified
by this ID. [PAIS WB0.8 ASM: replacementTransferObjectID (0..1)]</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="TransferObjectTypeID" type="xs:string">
            <xs:annotation>
                <xs:documentation>An ID unique within the Producer-
Archive Project which identifies the type of this Transfer Object instance. [PAIS WB0.8 ASM:
descriptorID (1..1)] (TBD) Any need to link to a particular version of descriptor?</
xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="TransferObjectID" type="xs:string">
            <xs:annotation>
                <xs:documentation>An ID unique within the Producer-
Archive Project which identifies this Transfer Object instance. [PAIS WB0.8 ASM:
transferObjectID (1..1) (For PAIS, it is only unique within context of Transfer Object Type,
the Producer Site, and Producer-Archive Project,)]</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="ParameterRange" type="ParameterRangeType"
minOccurs="0">
            <xs:annotation>
                <xs:documentation>Values for attributes that could be
used to discover the data content included in this Transfer Object instance. [PAIS WB0.8 ASM:
ANY]</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="NumberOfFilesIncluded" type="xs:positiveInteger"
minOccurs="0">
            <xs:annotation>
                <xs:documentation>Total of all actual files (i.e.,
directories are not counted) in the payload of this SIP instance. [PAIS WB0.8 ASM: ANY]</
xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="Group" type="GroupType" maxOccurs="unbounded">
            <xs:annotation>
                <xs:documentation>A set of values applicable to this
entire Group instance. Directories are one type of Group. (Can tar, (g)zip, arc, etc. be
others?) Restrictions: 0 groups may occur within a TransferObject only when
TransferObjectReplacedId is populated and then the absence of any group indicates that the
TransferObject identified by TransferObjectReplacedID is to be deleted. [PAIS WB0.8 ASM:
transferObjectGroup (1..N)]</xs:documentation>
            </xs:annotation>
        </xs:element>
    </xs:sequence>
</xs:complexType>

```



```

        </xs:annotation>
    </xs:element>
    <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded" />
</xs:sequence>
</xs:complexType>
<xs:complexType name="TransferObjectDeleteType">
    <xs:sequence>
        <xs:element name="TransferObjectIDToDelete" type="xs:string">
            <xs:annotation>
                <xs:documentation>An ID that matches a previously
submitted Transfer Object ID. The information contained within this Transfer Object instance
within the Archive should be deleted.r[PAIS WB0.8 ASM: proposed update]</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
</xs:complexType>
<xs:element name="SIPManifest">
    <xs:annotation>
        <xs:documentation>Submission Information Package Manifest [PAIS White
Book (WB) 0.8 Abstract SIP Model (ASM)]</xs:documentation>
    </xs:annotation>
    <xs:complexType>
        <xs:sequence>
            <xs:element name="SIPGlobal">
                <xs:annotation>
                    <xs:documentation>A set of attributes providing
values applicable to this entire SIP instance. [PAIS WB0.8 ASM: SIPGlobalInformation (1..1);
also has SIPSequenceNumber (0..1)]</xs:documentation>
                </xs:annotation>
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="ProducerArchiveProjectID"
type="xs:string">
                            <xs:annotation>
                                <xs:documentation>An ID
unique within the Archive for the Producer-Archive Project for which this SIP instance was
generated. [PAIS WB0.8 ASM: projectID (1..1)]</xs:documentation>
                            </xs:annotation>
                        </xs:element>
                        <xs:element name="ProducerID"
type="xs:string">
                            <xs:annotation>
                                <xs:documentation>An ID
unique within the Producer-Archive Project that indicates which Producer generated and sent
this SIP instance. This can be used to respond to the Producer. [PAIS WB0.8 ASM: producerID
(1..1)]</xs:documentation>
                            </xs:annotation>
                        </xs:element>
                        <xs:element name="SIPContentTypeID"
type="xs:string">
                            <xs:annotation>
                                <xs:documentation>An ID
unique within the Producer-Archive Project for the SIP Content Type being used for this SIP
instance. [PAIS WB0.8 ASM: sipContentTypeID (1..1)]</xs:documentation>
                            </xs:annotation>
                        </xs:element>
                        <xs:element name="SIPFormID">
                            <xs:annotation>
                                <xs:documentation>An ID
unique within the (TBD) ProducerArchiveProject | Producer | ProducerSite? for the SIP Form
being used for this SIP instance. [PAIS WB0.8 ASM: ANY]</xs:documentation>
                            </xs:annotation>
                        </xs:complexType>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>

```

```

<xs:sequence>
  <xs:element
name="SIPForm" type="xs:string">
  <xs:annotation>
<xs:documentation>A, B, C, or D from Don's presentation.</xs:documentation>
  </
xs:annotation>
  </xs:element>
  <xs:element
name="SIPFormVersion" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="SIPID" type="xs:string">
  <xs:annotation>
    <xs:documentation>An ID
unique within the Producer-Archive Project Archive for this SIP instance. [PAIS WBO.8 ASM:
SIPID(1..1)]</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="ParameterRange"
type="ParameterRangeType" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Values for
attributes that could be used to discover the data content included in this SIP instance.
[PAIS WBO.8 ASM: ANY]</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="NumberOfFilesIncluded"
type="xs:positiveInteger" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Total of
all actual files (i.e., directories are not counted) in the payload of this SIP instance.
[PAIS WBO.8 ASM: ANY]</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="ProducerComment"
type="xs:string" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Any type
of comment from the producer regarding this SIP instance. [PAIS WBO.8 ASM: ANY]</
xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="CreationTime"
type="xs:dateTime" minOccurs="0">
  <xs:annotation>
    <xs:documentation>The time
(in UTC) the producer created this SIP instance. [PAIS WBO.8 ASM: ANY]</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:any namespace="##other" minOccurs="0"
maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
</xs:element>
<xs:choice maxOccurs="unbounded">
  <xs:element name="TransferObject"
type="TransferObjectType">
  <xs:annotation>
    <xs:documentation>A set of values
applicable to this entire Transfer Object instance. [PAIS WBO.8 ASM:

```

```
transferObjectIdentificationAndStatusInformation (1..N)]</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="DeleteTransferObject"
type="TransferObjectDeleteType">
    <xs:annotation>
        <xs:documentation>A set of values
applicable to this entire Transfer Object instance. [PAIS WB0.8 ASM: ]</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:any namespace="##other" minOccurs="0"
maxOccurs="unbounded" />
    </xs:choice>
    </xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```