



NSSDC Manifest Builder v.0.1.0

for the Planetary Data System

Table of Contents

1 NSSDC Manifest Builder Guide	
1.1 Overview	1
1.2 Release Notes	2
1.3 Installation	3
1.4 Operation	5
1.5 XML Schema	10
1.6 Appendix A - UNIX Setup Options	11
1.7 Appendix B - Windows Setup Options	12

1.1 Overview

About NSSDC Manifest Builder

The NSSDC Manifest Builder is a command-line tool for generating a new NSSDC Delivery Manifest document based on provided input and an accessible NSSDC delivery directory. Generation of the Delivery Manifest is just one step in the process for submission of PDS Data Sets to the NSSDC. For additional information on the submission process see the process description document and Xman Operations Guide published by the NSSDC.

Please send comments, change requests and bug reports to the [PDS Operator](mailto:pds_operator@jpl.nasa.gov) at pds_operator@jpl.nasa.gov.

1.2 Release Notes

Release Notes

The purpose of this section is to provide a description of a NSSDC Manifest Builder release including any impact that the new or modified capabilities will have on the Discipline Nodes or the PDS user community. A somewhat itemized list of changes for each release can be found on the [Release Changes](#) page. If viewing this document in PDF form, see the appendix for details.

Release 0.1.0

This is the first release of the application as a prototype for proof-of-concept testing. It includes support for creating Delivery Manifest documents based on the corresponding XML Schema.

1.3 Installation

Installation

This section describes how to install the NSSDC Manifest Builder contained in the *manifest-builder* package. The following topics can be found in this section:

- [System Requirements](#)
- [Unpacking the Package](#)

System Requirements

The NSSDC Manifest Builder was developed using Java and will run on any platform with a supported Java Runtime Environment (JRE). The tool was specifically developed under Sun Java version 1.5, so the tool will execute correctly under versions 1.5 or 1.6.

Since the tool was developed using Sun's Java, this is the preferred Java environment for operation. The Sun Java package can be obtained from the [Sun Java](#) web site. Other Java environments are relatively compatible with Sun's Java.

Unpacking the Package

The binary distribution is available in identical zip or tar/gzip packages. Unpack the selected binary distribution file with one of the following commands:

```
[node: ~] unzip manifest-builder-0.1.0-bin.zip  
or  
[node: ~] tar -xzf manifest-builder-0.1.0-bin.tar.gz
```

Note: Depending on the platform, the native version of *tar* may produce an error when attempting to unpack the distribution file because many of the file paths are greater than 100 characters. If available, the GNU version of *tar* will resolve this problem. If that is not available or cannot be installed, the zipped package will work just fine in a UNIX environment.

The commands above result in the creation of the *manifest-builder-0.1.0* directory with the following directory structure:

- **README.txt**

A README file directing the user to the available documentation for the project.

- **LICENSE.txt**

The copyright notice from the [California Institute of Technology](#) detailing the restrictions regarding the use and distribution of this software. Although the license is strictly worded, the software has been classified as Technology and Software Publicly Available (TSPA) and is available for *anyone* to download and use.

- **bin/**

This directory contains batch and shell scripts for executing the tool along with the XML Schema for the Delivery Manifest.

- **doc/**

This document directory contains a local web site with the NSSDC Delivery Manifest Guide, javadoc, unit test results and other configuration management related information. Just point your favorite browser to the *index.html* file in this directory.

- **lib/**

This directory contains the dependent jar files for the tool along with the executable jar file (manifest-builder-0.1.0.jar) containing the NSSDC Delivery Manifest software.

- **test/**

This directory contains a test Submission Information Package (SIP) for NSSDC.

1.4 Operation

Operation

The NSSDC Manifest Builder generates a new NSSDC Delivery Manifest document based on provided input and an accessible NSSDC delivery directory. The following topics can be found in this section:

- [Tool Setup](#)
- [Tool Execution](#)

Note: The command-line examples in this section have been broken into multiple lines for readability. The commands should be reassembled into a single line prior to execution.

Tool Setup

In order to execute the NSSDC Manifest Builder, the user's environment must first be configured appropriately. This section describes how to setup the user environment on UNIX-based and Windows machines.

UNIX-Based Setup

This section details the environment setup for UNIX-based machines. The preferred method is to specify the shell script, *NSSDCManifest*, on the command-line. Setting the *PATH* environment variable to the location of the script, enables the shell script to be executed from any location on the user's machine.

The following command demonstrates how to set the *PATH* environment variable, by appending to its current setting:

```
% setenv PATH ${PATH}:%HOME/manifest-builder-0.1.0/bin
```

The tool can now be executed via the shell script as demonstrated in the following example:

```
% NSSDCManifest <command-line arguments>
```

Additional methods for setting up a UNIX-based environment can be found in the [UNIX Setup Options](#) section. If viewing this document in PDF form, see the appendix for details.

Windows Setup

This section details the environment setup for Windows machines. The preferred method is to specify the batch file, *NSSDCManifest.bat*, on the command-line. Setting the *PATH* environment variable to the location of the file, enables the batch file to be executed from any location on the user's machine.

The following command demonstrates how to set the *PATH* environment variable, by appending to its current setting:

```
C:\> set PATH = %PATH%;C:\manifest-builder-0.1.0\bin
```

The tool can now be executed via the batch file as demonstrated in the following example:

```
C:\> NSSDCManifest <command-line arguments>
```

Additional methods for setting up a Windows environment can be found in the [Windows Setup Options](#) section. If viewing this document in PDF form, see the appendix for details.

Tool Execution

NSSDC Manifest Builder can be executed in various ways. This section describes how to run the tool, as well as the behaviors and caveats of the tool.

Command-Line Options

The following table contains command-line options available to the tool:

Flag	Description
-R, --root-dir	Specify the root local directory path where the volumes and SIP manifests are located. The default directory is ".".
-O, --output-filename	Specify an output filename for the resulting Delivery Manifest file. Defaults to the value of "<root-dir>/<Delivery_Identifier>.xml". The <Delivery_Identifier> defaults to "<pds-node-id>_YYYY-MM-DDTHHMMSS".

Flag	Description
-H, --manifest-builder-home	Specify the path to find the XML Schema for the Delivery Manifest. The default directory is ".".
-D, --delivery-description	Specify a general description for the delivery manifest. Defaults to "An Electronic Delivery of a SIP, completed on [CURRENT-DATE]".
-n, --pds-node-id	Specify the 3 Letter ID for the PDS Node creating the delivery manifest with the PDS_ prefix. Valid values are PDS_ATM, PDS_CDN, PDS_ENG, PDS_GEO, PDS_GRE, PDS_HRD, PDS_IMG, PDS_JPL, PDS_LDN, PDS_LOL, PDS_LRO, PDS_NAI, PDS_PPI, PDS_PSI, PDS_RNG, PDS_RSF, PDS_SBN, PDS_SWR, PDS_TES, PDS_THE and PDS_UOI.
-a, --archive-volume-uri	Specify one or more archive volume URIs for the current delivery.
-i, --delivery-filename	Specify the name of the delivery file that will contain the SIP.
-u, --delivery-url	Specify the base URL for the delivery manifest.
-f, --package-format	Specify the electronic delivery package format for the SIP information to be stored. The valid values are ZIP, TAR and GZIP_TAR. The default value is "ZIP".
-E, --electronic-delivery-description	Specify a description for the electronic delivery. Defaults to "An Electronic Delivery of a SIP, completed on [CURRENT-DATE]".
-N, --electronic-delivery-number	Specify a delivery number for the electronic delivery. The default value is "1".
-p, --this-package-id	Specify an identifier for this specific package. Defaults to the value of "<Electronic_Delivery_Identifier>".
-P, --data-path	Specify the data path for the manifest. The default directory is "/".
-s, --sip-directory	Specify the name of the directory where SIP Manifest XML files are located. The default directory is "SIP_Manifest".
-S, --schema-url	Specify the local file URL where the NSSDC Delivery Manifest XML Schema document is located.
-c, --config	Specify a configuration file to set the default values.
-h, --help	Display NSSDC Manifest Builder usage.
-V, --version	Display NSSDC Manifest Builder version.

Execute NSSDC Manifest Builder

This section demonstrates execution of the tool using the command-line option flags. The examples below execute the tool via the batch/shell script. Alternate methods for executing the tool can be found in the [Tool Setup](#) section.

The following command demonstrates generation of the Delivery Manifest from the test data provided in the deployment package replicating the values that are found in the example Delivery Manifest (PDS_SBN_2010-03-25T163612.xml):

```
% NSSDCManifest -R $HOME/manifest-builder-0.1.0/test/sip-1 -D "test" -n PDS_SBN \
-a N/A -i test.xml -u http://pdssbn.astro.umd.edu/delivery/NSSDC_test/test.xml -f ZIP -E
"test"
```

A configuration file can be passed into the command-line to change the default behaviors of the tool and to also provide users a way to perform validation with a single flag. For more details on how to setup the configuration file, see the [Using a Configuration File](#) section.

The following command demonstrates performing validation using a configuration file:

```
% NSSDCManifest -c config.txt
```

Using a Configuration File

A configuration file is used to set the default behaviors of the tool. It consists of a text file made up of keyword/value pairs. The configuration file follows the syntax of the stream parsed by the Java `Properties.load(java.io.InputStream)` method.

Blank lines and lines which begin with the hash character "#" are ignored.

Values may be separated on different lines if a backslash is placed at the end of the line that continues below.

Escape sequences for special characters like a line feed, a tabulation or a unicode character, are allowed in the values and are specified in the same notation as those used in Java strings (e.g. `\n`, `\t`, `\r`).

Since backslashes (`\`) have special meanings in a configuration file, keyword values that contain this character will not be interpreted properly by the tool even if it is surrounded by quotes. A common example would be a Windows path name (e.g. `c:\VTT_EN_1-1\target`). Use the forward slash character instead (`c:/VTT_EN_1-1/target`) or escape the backslash character (`c:\\VTT_EN_1-1\\target`).

Note: Any flag specified on the command-line takes precedence over any equivalent settings placed in the configuration file.

The following table contains valid keywords that can be specified in the configuration file:

Keyword	Associated Command-Line Option
nssdc.root-dir	-R, --root-dir
nssdc.output-filename	-o, --output-filename

Keyword	Associated Command-Line Option
nssdc.delivery-description	-D, --delivery-description
nssdc.pds-node-id	-n, --pds-node-id
nssdc.archive-volume-uri	-a, --archive-volume-uri
nssdc.delivery-filename	-i, --delivery-filename
nssdc.delivery-url	-u, --delivery-url
nssdc.package-format	-f, --package-format
nssdc.electronic-delivery-description	-E, --electronic-delivery-description
nssdc.electronic-delivery-number	-N, --electronic-delivery-number
nssdc.this-package-id	-p, --this-package-id
nssdc.data-path	-P, --data-path
nssdc.sip-directory	-s, --sip-directory
nssdc.schema-url	-S, --schema-url

The following example demonstrates how to populate a configuration file:

```
# NSSDC Manifest Builder Config File

nssdc.pds-node-id = PDS_SBN
nssdc.archive-volume-uri = N/A
nssdc.package-format = ZIP
```

Using the above configuration file, the following command is equivalent to the first example execution in the [Execute NSSDC Manifest Builder](#) section:

```
% NSSDCManifest -c config.txt -R $HOME/manifest-builder-0.1.0/test/sip-1 -D "test" \
-i test.xml -u http://pdssbn.astro.umd.edu/delivery/NSSDC_test/test.xml -E "test"
```

1.5 XML Schema

XML Schema

This document lists the documentation available for the NSSDC Delivery Manifest XML Schema:

- [XML Schema Document](#)
- [Schema Documentation](#)

If viewing this document in PDF form, the Schema Documentation is not available and the link to the XML Schema Document is:

http://pds.nasa.gov/schema/pds3/ops/NSSDCPDSDeliveryManifest_V1_0_0.xsd

1.6 Appendix A - UNIX Setup Options

UNIX Setup Options

This section details a couple of options for setting up a UNIX environment for launching VTool.

Specify the CLASSPATH on the Command-Line

An alternative method to setting the *CLASSPATH* variable with all of the tool's dependent JAR files is to specify the *java.ext.dirs* Java property on the command-line when running the tool each time. This is done by passing the property via the Java "-D" flag as demonstrated in the following example:

```
% java -Djava.ext.dirs=$HOME/manifest-builder-0.1.0/lib \
gov.nasa.pds.nssdc.manifest.builder.MFBuilder <command-line arguments>
```

Specify the JAR on the Command-Line

Another alternative method is to specify the executable JAR file on the command-line when running the tool each time. This is done by passing the JAR file specification via the Java "-jar" flag as demonstrated in the following example:

```
% java -jar $HOME/manifest-builder-0.1.0/lib/manifest-builder-0.1.0.jar <command-line
arguments>
```

1.7 Appendix B - Windows Setup Options

Windows Setup Options

This section details a couple of options for setting up a Windows environment for launching VTool.

Specify the CLASSPATH on the Command-Line

An alternative method to setting the *CLASSPATH* variable with all of the tool's dependent JAR files is to specify the *java.ext.dirs* Java property on the command-line when running the tool each time. This is done by passing the property via the Java "-D" flag as demonstrated in the following example:

```
C:\> java -Djava.ext.dirs=c:\manifest-builder-0.1.0\lib \
gov.nasa.pds.nssdc.manifest.builder.MFBuilder <command-line arguments>
```

Specify the JAR on the Command-Line

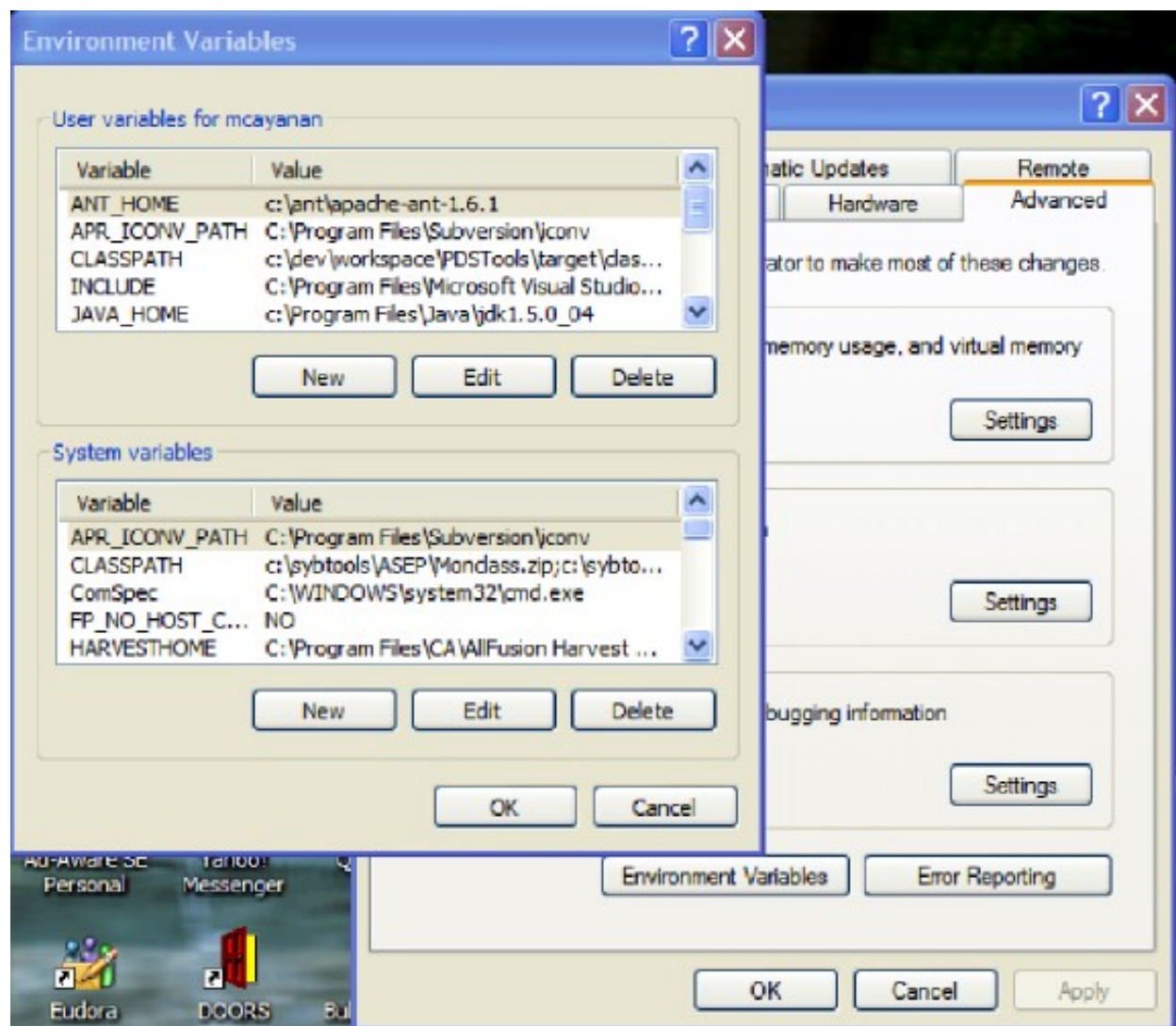
Another alternative method is to specify the executable JAR file on the command-line when running the tool each time. This is done by passing the JAR file specification via the Java "-jar" flag as demonstrated in the following example:

```
C:\> java -jar c:\manifest-builder-0.1.0\lib\manifest-builder-0.1.0.jar <command-line
arguments>
```

Specify the Path in the Control Panel

The method for setting the executable path permanently for VTool is to set the *Path* environment variable via the control panel as follows:

- Right-click on *My Computer* icon on your desktop and select the *Properties* menu item.
- Navigate to the *Advanced* tab and select the *Environment Variables* button. At this point, you should now see a window like the one below:



- Highlight the *Path* variable in the System Variables list and select the Edit button.
- Append to the current contents of the variable, the path to the *bin* directory within *manifest-builder* package. Separate the package path from the current contents of the variable with a semicolon.
- Select the OK button when you are finished editing the *Path* variable, then select the OK button at the Environment Variables window to apply the changes.

Note: If you already have a DOS window open, you will need to close and re-open the window for the *Path* changes to take effect.