



Catalog Ingest Tool v.0.4.0

for the Planetary Data System

Table of Contents

1	Catalog Ingest Tool Guide	
1.1	Overview	1
1.2	Release Notes	2
1.3	Installation	3
1.4	Operation	5
1.5	Appendix A - UNIX Setup Options	16
1.6	Appendix B - Windows Setup Options	17
1.7	Appendix C - Referential Integrity Checking	19
1.8	Appendix D - Report Examples	22
1.8.1	Compare Report	23
1.8.2	Validate Report	25
1.8.3	Ingest Report	28

1.1 Overview

About Catalog Ingest Tool

The Catalog Ingest Tool (CITool) is intended to streamline the ingestion process. The legacy process involves a myriad of tools and manual steps in order to ingest catalog files submitted by the nodes into the PDS central catalog. CITool offers three functions:

- Compare a Catalog Submission

The comparison function is intended for both Discipline Node and Engineering Node Data Engineers to determine the differences between previous submissions and a current submission.

- Validate a Catalog Submission

The validation function is also intended for both Discipline Node and Engineering Node Data Engineers to validate a current submission including checks for referential integrity.

- Ingest a Catalog Submission

The ingestion function is intended for the Engineering Node Operator to ingest the current submission into the PDS central catalog.

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 Catalog Ingest Tool 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 0.4.0

This release of the Catalog Ingest Tool, is intended for beta testing by the PDS Technical Staff. The minor changes in this release include some updates to the command-line options and message severities.

Release 0.3.0

This release of the Catalog Ingest Tool, is intended for beta testing at the Engineering Node. The major changes in this release include the capabilities to find new standard values and perform a complete referential integrity check within a set of catalog files. Other changes include various, minor fixes to reference validation and the compare mode functionality based on the previous beta testing period.

Release 0.2.0

This release of the Catalog Ingest Tool, is intended for beta testing at the Engineering Node. This release includes the capabilities to validate catalog files, perform referential integrity checking for references and ingest the catalog files into the PDS central catalog database.

Release 0.1.0

This is the first release of the Catalog Ingest Tool, intended for beta testing at the Engineering Node. This release includes the capability to compare a catalog file with another catalog file or a directory of catalog files with another directory.

1.3 Installation

Installation

This section describes how to install the Catalog Ingest Tool (CITool) contained in the *citool* package. The following topics can be found in this section:

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

System Requirements

The following sub-sections detail the system requirements for the tool.

Java Requirement

The Catalog Ingest Tool 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.

Data Dictionary Requirement

Release *1r64* or later of the Planetary Science Data Dictionary (PSDD) is required for the tool to function properly. Release *1r66* of the PSDD supports the validation of explicit FILE objects. The latest version of the PDS data dictionary can be retrieved from the [PDS Data Dictionary](#) web page.

Unpacking the CITool Package

Download the *citool* package from the [Tools & Documentation](#) web page. The binary distribution is available in identical zip or tar/gzip packages. Unpack the selected binary distribution file with one of the following commands:

```
% unzip citool-0.4.0-bin.zip
```

```
or
% tar -xzvf citool-0.4.0-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 *citool-0.4.0* 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.

- **doc/**

This document directory contains a local web site with the Catalog Ingest 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 (citool-0.4.0.jar) containing the Catalog Ingest Tool software.

1.4 Operation

Operation

The Catalog Ingest Tool (CITool) has three functions:

- **Compare a Catalog Submission**
Comparisons of catalog submissions, either file to file or directory to directory, can be performed with a report detailing the differences.
- **Validate a Catalog Submission**
Validation specific to catalog files can be performed with a report detailing the results. In addition, referential integrity checks are performed within the set of catalog files. Details on the types of checks performed can be found in the [Referential Integrity Checking](#) section. If viewing this document in PDF form, see the appendix for details.
- **Ingest a Catalog Submission**
Ingestion into the central catalog database can be performed assuming the user has appropriate permissions.

The following topics can be found in this section:

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

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 CITool, 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, *CITool*, 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/citool-0.4.0/bin
```

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

```
% CITool <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, *CITool.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:\citool-0.4.0\bin
```

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

```
C:\> CITool <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

CITool can be executed in various ways. This section describes how to run the tool, as well as its behaviors and caveats.

Command-Line Options

CITool can be run in three modes: compare, validate and ingest. The following tables describe the command-line flag options available when each mode is enabled.

Compare Mode

Setting the *m* flag option with *compare* will enable the tool to run in compare mode. In the compare mode, the following flag options are valid:

Flag	Description
-m, --mode compare	Specifying <i>compare</i> runs the tool in compare mode.
-t, --target <catalogs,URLs,dirs>	Explicitly specify two targets (catalog files, directories, and URLs) to compare. Targets can also be specified implicitly (example: CITool OLDDATASET.CAT, NEWDATASET.CAT).
-r, --report-file <file>	Specify the report file name. Default is standard out.
-v, --verbose <1 2 3>	Specify the severity level and above to include in the report: (1=Info, 2=Warning, 3=Error). Default is warning and above (level 2).
-c, --config <file>	Specify a configuration file to set the default values.
-h, --help	Display CITool usage.
-V, --version	Display CITool version.

Validate Mode

Setting the *m* flag option with *validate* will enable the tool to run in validate mode. In the validate mode, the following flag options are valid:

Flag	Description
-m, --mode validate	Specifying <i>validate</i> runs the tool in validate mode.
-d, --dict <.full file(s)>	Specify the Planetary Science Data Dictionary full file name and any local dictionaries.
-a, --alias;	Enable aliasing.
-A, --allrefs <allrefs file>	Specify the allrefs dictionary support file or URL.
-r, --report-file <file>	Specify the report file name. Default is standard out.
-t, --target <catalogs,URLs,dirs>	Explicitly specify the target (catalog file, directory, or URL). The target can also be specified implicitly (example: CITool DATASET.CAT).

Flag	Description
-v, --verbose <1 2 3>	Specify the severity level and above to include in the report: (1=Info, 2=Warning, 3=Error). Default is warning and above (level 2).
-c, --config <file>	Specify a configuration file to set the default values.
-h, --help	Display CITool usage.
-V, --version	Display CITool version.

Ingest Mode

Setting the *m* flag option with *ingest* will enable the tool to run in ingest mode. In the ingest mode, the following flag options are valid:

Flag	Description
-m, --mode ingest	Specifying <i>ingest</i> runs the tool in ingest mode.
-t, --target <catalog,URLs,dirs>	Explicitly specify target (catalog file, directory, and URLs) to ingest. Targets can also be specified implicitly (example: CITool -m ingest TEST.CAT).
-u, --dbuser <username>	Username of the database.
-p, --dbpass <password>	Password of the database.
-s, --dbserver <servername>	Name of the database server.
-n, --dbname <db name>	Name of the database.
-v, --verbose <1 2 3>	Specify the severity level and above to include in the report: (1=Info, 2=Warning, 3=Error). Default is warning and above (level 2).
-c, --config <file>	Specify a configuration file to set the default values.
-h, --help	Display CITool usage.
-V, --version	Display CITool version.

Running CITool

This section demonstrates some of the ways that the tool can be run using the command-line option flags.

Compare Mode

In compare mode, the tool can run in the following ways:

- Comparing Two Catalog Files

The following command demonstrates how to compare a source catalog file

\$HOME/DIR1/DATASET.CAT with a target catalog file *\$HOME/DIR2/DATASET.CAT*:

```
% CITool $HOME/DIR1/DATASET.CAT, $HOME/DIR2/DATASET.CAT -m compare
```

- Comparing Two Catalog Directories

The following command demonstrates how to compare a source directory, *\$HOME/DIR1*, containing a set of catalog files with a target directory, *\$HOME/DIR2*, containing another set of catalog files.

```
% CITool $HOME/DIR1, $HOME/DIR2 -m compare
```

In this example, the tool will look for files with matching file names between the source and target directory before doing a comparison. So, if a *DATASET.CAT* file is found in directory *\$HOME/DIR1*, a *DATASET.CAT* file needs to be found in directory *\$HOME/DIR2*, and so forth.

- Writing a Compare Report to File

In the first two examples above, the output report is written to standard out. The following command demonstrates how to write the report to a file named *report.txt*:

```
% CITool $HOME/DIR1/DATASET.CAT, $HOME/DIR2/DATASET.CAT -m compare -r report.txt
```

Validate Mode

In validate mode, the tool can run in the following ways:

- Validate a Single Catalog File

The following command demonstrates how to validate a single catalog file *DATASET.CAT* against a data dictionary *pdsdd.full*:

```
% CITool DATASET.CAT -m validate -d pdsdd.full
```

- Validate a Directory of Catalog Files

The following command demonstrates how to validate a directory *\$HOME/DIR* containing a set of catalog files against a data dictionary *pdsdd.full*:

```
% CITool $HOME/DIR -m validate -d pdsdd.full
```



- Perform Referential Integrity Check with the Allrefs File

The allrefs dictionary support file ensures that the reference citations in the *REFS.CAT* file are consistent with what is currently in the PDS database.

The following command demonstrates how to specify an allrefs dictionary support file, *allrefs.out*, to validate and do a complete referential integrity check on references from a set of catalog files in a directory *\$HOME/DIR*.

```
% CITool $HOME/DIR -m validate -d pdsdd.full -a allrefs.out
```

- Writing the Validation Report to File

The following command demonstrates how to write a validation report to a file named *report.txt*:

```
% CITool $HOME/DIR -m validate -d pdsdd.full -a allrefs.out -r report.txt
```

Ingest Mode

In ingest mode, the tool can run in the following ways:

- Ingest a Single Catalog File

The following command demonstrates how to ingest a single catalog file *DATASET.CAT* as a user *tempuser* with a password *temppass* into the database server *starsyb* and the database name *tempdb*.

```
% CITool DATASET.CAT -m ingest -u tempuser -p temppass -s starsyb -n tempdb
```

- Ingest a Directory of Catalog Files

The following command demonstrates how to ingest a directory *\$HOME/DIR* containing a set of catalog files as a user *tempuser* with a password *temppass* into the database server *starsyb* and the database name *tempdb*.

```
% CITool $HOME/DIR -m ingest -u tempuser -p temppass -s starsyb -n tempdb
```

Changing Tool Behaviors Using a Configuration File

A configuration file can be passed to the tool to change its default behaviors. This provides a way to use the tool 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 how to run the tool using a configuration file:

```
% CITool -c config.cfg
```

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 CITool 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 when running in compare mode:

Property Key	Associated Flag	Valid Value(s)
citool.mode	-m	Specify <i>compare</i> to run the tool in compare mode.
citool.targets	-t	Specify two targets (catalog files, directories, and URLs) to compare.
citool.report	-r	Specify the report file name. Do not specify this property key if writing to standard out.

Property Key	Associated Flag	Valid Value(s)
citool.verbose	-v	Specify the severity level and above to include in the report (1=info, 2=warning, 3=error). Default is warnings and above (level 2).

The following table contains valid keywords that can be specified in the configuration file when running in validate mode:

Property Key	Associated Flag	Valid Value(s)
citool.mode	-m	Specify <i>validate</i> to run the tool in validate mode.
citool.targets	-t	Specify a target (catalog file, directory, or URL) to validate.
citool.dictionaries	-d	Specify the Planetary Science Data Dictionary full file name and any local dictionaries.
citool.alias	-a	Set to <i>true</i> to enable aliasing.
citool.allrefs	-A	Specify the ' <i>allrefs</i> ' dictionary file or URL.
citool.report	-r	Specify the report file name. Do not specify this property key if writing to standard out.
citool.verbose	-v	Specify the severity level and above to include in the report (1=info, 2=warning, 3=error). Default is warnings and above (level 2).

The following table contains valid keywords that can be specified in the configuration file when running in ingest mode:

Property Key	Associated Flag	Valid Value(s)
citool.mode	-m	Specify <i>ingest</i> to run the tool in ingest mode.
citool.targets	-t	Specify targets (catalog files, directories, and URLs) to ingest.
citool.dbuser	-u	Specify the username of the database.
citool.dbpass	-p	Specify the password of the database.
citool.dbserver	-s	Specify the name of the database server.
citool.dbname	-n	Specify the name of the database.
citool.report	-r	Specify the report file name. Do not specify this property key if writing to standard out.
citool.verbose	-v	Specify the severity level and above to include in the report (1=info, 2=warning, 3=error). Default is warnings and above (level 2).

The following example demonstrates how to set a configuration file:

```
# This is a CITool configuration file

citool.mode      = validate
citool.targets   = ./TEST_DIR
citool.report    = report.txt
citool.dictionaries = pdsdd.full
```

This is equivalent to running the tool with the following flag options:

```
-t ./TEST_DIR -m validate -r report.txt -d pdsdd.full
```

The following example demonstrates how to set a configuration file with multiple values for a property key:

```
# This is a CITool configuration file with multiple values

citool.mode      = compare
citool.targets   = DIR
citool.dictionaries = pdsdd.full, localdd.full
```

This is equivalent to running the tool with the following flags:

```
-t DIR -m compare -d pdsdd.full, localdd.full
```

The following example demonstrates how to set a configuration file with multiple values that span across multiple lines:

```
# This is a CITool configuration file with multiple values

citool.mode      = compare
citool.targets   = DIR
citool.dictionaries = pdsdd.full, \
                    localdd.full
```

The following example demonstrates how to override a setting in the configuration file.

Suppose the configuration file *config.cfg* is defined as follows:

```
# This is a CITool configuration file

citol.mode      = validate
citol.targets   = ./TEST_DIR
citol.dictionaries = pdsdd.full
```

To use another dictionary instead like *mypdsdd.full*, then the following command demonstrates how to perform this behavior change:

```
% CITool -c config.cfg -d mypdsdd.full
```

Report Formats

This section describes the contents of the CITool report formats. The links below detail the results. If viewing this document in PDF form, see the appendix for the actual examples.

The tool has a different report format depending on the CITool mode.

Compare Report

In a [compare](#) report, the location, severity, and textual representation of the differences between a source and target catalog file are shown. The differences being reported are about the target catalog file with respect to the source. A 'SAME' or 'DIFFERENT' keyword is displayed next to each target to indicate when a target file is identical or different, respectively, against its source.

Validate Report

In a [validate](#) report, the location, severity, and textual description of each detected anomaly is reported. A 'PASS', 'FAIL', or 'SKIP' keyword is displayed next to each file to indicate when a file has passed, failed, or skipped PDS validation, respectively. In addition, anomalies against the referential integrity of the catalog files are reported.

Ingest Report

In a [ingest](#) report, the location, severity, and textual description of each detected anomaly is reported.

Completion or failure of each catalog file ingestion is reported. Detailed information is displayed when the catalog ingestion has failed.

1.5 Appendix A - UNIX Setup Options

UNIX Setup Options

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

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/citool-0.4.0/lib \  
gov.nasa.pds.citool.CITool <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/citool-0.4.0/lib/citool-0.4.0.jar <command-line arguments>
```

1.6 Appendix B - Windows Setup Options

Windows Setup Options

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

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:\citool-0.4.0\lib \
gov.nasa.pds.citool.CITool <command-line arguments>
```

Specify the JAR on the Command-Line

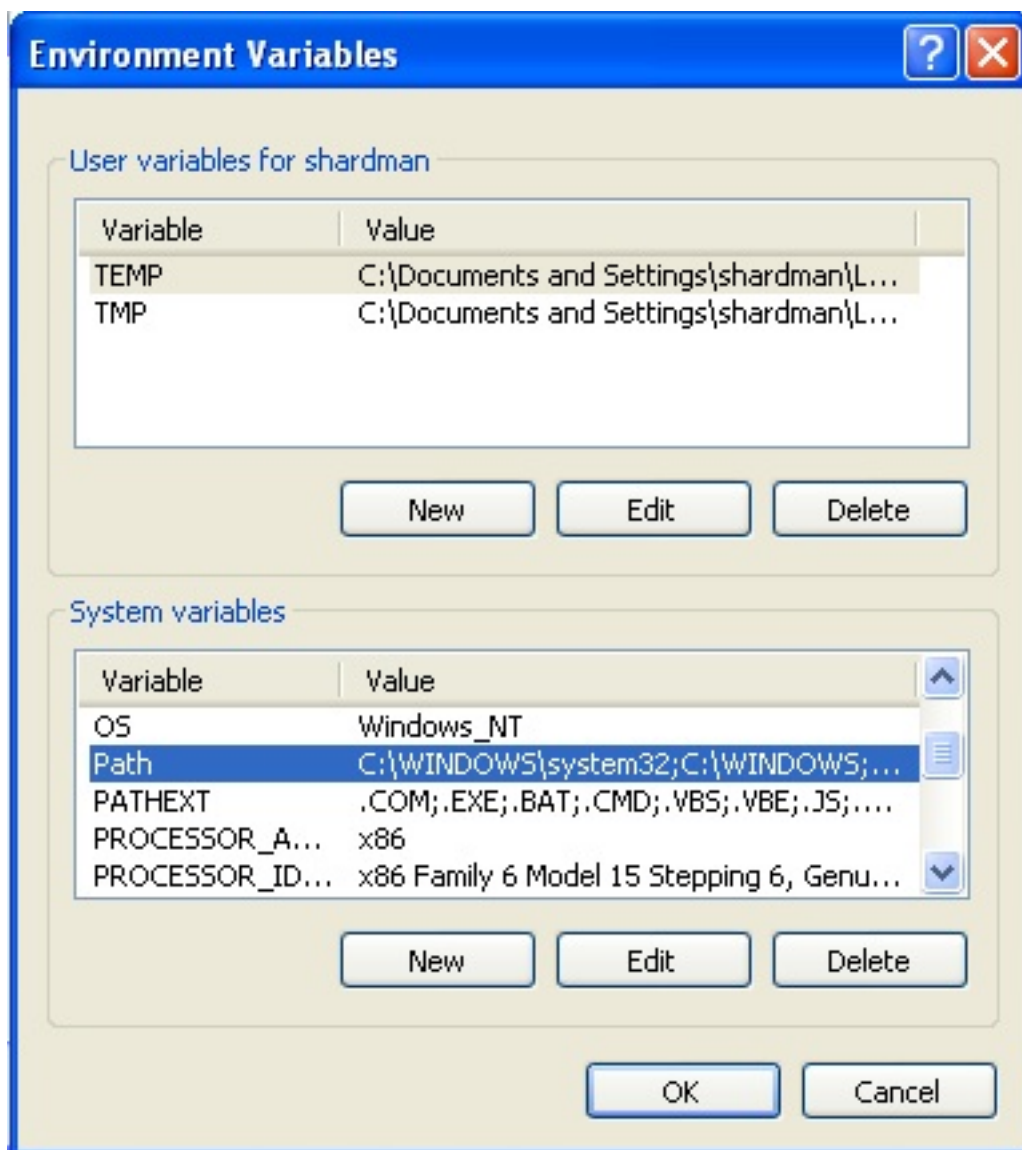
Another alternative method is to specify the executable JAR file on the command-line when running the tool each time. This is done by passing the JAR file specification via the Java "-jar" flag as demonstrated in the following example:

```
C:\> java -jar \
c:\citool-0.4.0\lib\citool-0.4.0.jar <command-line arguments>
```

Specify the Path in the Control Panel

The method for setting the executable path permanently for CITool 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 *citool* 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.

1.7 Appendix C - Referential Integrity Checking

Referential Integrity Checking

This section describes how CITool checks for referential integrity among the catalog files.

Referential integrity is a database concept that ensures that relationships between the tables remain consistent. In CITool, referential integrity is performed during validation mode. The following referential integrity checks are performed:

- [Reference Integrity](#)
- [Personnel Integrity](#)
- [Target Integrity](#)
- [Instrument Integrity](#)
- [Instrument Host Integrity](#)
- [Mission Integrity](#)
- [Volume Integrity](#)
- [Data Set Integrity](#)

Note, however, that not all referential integrity checks can be performed if the set of catalog files that were passed into the tool is incomplete. For example, if a *TARGET.CAT* file is not passed into the tool, then target integrity checks will not be performed.

Reference Integrity

In a reference integrity check, CITool will ensure that referential integrity exists in the use of *REFERENCE_KEY_ID* and *REFERENCE_DESC* as follows:

- Each *REFERENCE_KEY_ID* value specified in a catalog file other than the *REF.CAT* file exists in the *REF.CAT* file.
- Each *REFERENCE_KEY_ID* value specified in the *REF.CAT* file is specified in at least one other catalog file.
- Each *REFERENCE_KEY_ID* value exists in the '*allrefs*' dictionary support file.
- Each *REFERENCE_DESC* value specified in the *REF.CAT* file matches the corresponding value in the '*allrefs*' dictionary support file.

Personnel Integrity

In a personnel integrity check, CITool will ensure that referential integrity exists in the use of *PDS_USER_ID* and *NODE_ID* as follows:

- Each *PDS_USER_ID* and *NODE_ID* value specified in a catalog file other than the *PERSON.CAT* file exists in the *PERSON.CAT* file.

Target Integrity

In a target integrity check, CITool will ensure that referential integrity exists in the use of *TARGET_NAME* as follows:

- Each *TARGET_NAME* value specified in a catalog file other than the *TARGET.CAT* file exists in the *TARGET.CAT* file.
- Each *TARGET_NAME* value specified in the *TARGET.CAT* file is specified in at least one other catalog file.

Instrument Integrity

In an instrument integrity check, CITool will ensure that referential integrity exists in the use of *INSTRUMENT_ID* as follows:

- Each *INSTRUMENT_ID* value specified in a catalog file other than the *INSTRUMENT.CAT* file exists in the *INSTRUMENT.CAT* file.
- Each *INSTRUMENT_ID* value specified in the *INSTRUMENT.CAT* file is specified in at least one other catalog file.

Instrument Host Integrity

In an instrument host integrity check, CITool will ensure that referential integrity exists in the use of *INSTRUMENT_HOST_ID* as follows:

- Each *INSTRUMENT_HOST_ID* value specified in a catalog file other than the *INSTRUMENT.CAT* file, exists in the *INSTRUMENT.CAT* file.
- Each *INSTRUMENT_HOST_ID* value specified in the *INSTRUMENT.CAT* file is specified in at least one other catalog file.

Mission Integrity

In a mission integrity check, CITool will ensure that referential integrity exists in the use of *MISSION_NAME* as follows:

- Each *MISSION_NAME* value specified in a catalog file other than the *MISSION.CAT* file exists in the *MISSION.CAT* file.

- Each *MISSION_NAME* value specified in the *MISSION.CAT* file is specified in at least one other catalog file.

Volume Integrity

In a volume integrity check, CITool will ensure that referential integrity exists in the use of *MEDIUM_TYPE* as follows:

- Each *MEDIUM_TYPE* value specified in a catalog file other than the *VOLUME.CAT* file exists in the *VOLUME.CAT* file.
- Each *MEDIUM_TYPE* value specified in the *VOLUME.CAT* file is specified in at least one other catalog file.

Data Set Integrity

In a data set integrity check, CITool will ensure that referential integrity exists in the use of *DATA_SET_ID* and *DATA_SET_ID_COLLECTION_ID* as follows:

- Each *DATA_SET_ID* value specified in a catalog file other than the *DATASET.CAT* file exists in the *DATASET.CAT* file.
- Each *DATA_SET_ID* value specified in the *DATASET.CAT* file is specified in at least one other catalog file.
- The *DATA_SET_COLL_OR_DATA_SET_ID* value specified in the *VOLUME.CAT* and *NSSDC.CAT* files is consistent with the *DATA_SET_ID* or *DATA_SET_COLLECTION_ID* value specified in the *DATASET.CAT* file.
- The *PRODUCT_DATA_SET_ID* value specified in the *INVENTORY.CAT* file is consistent with the *DATA_SET_ID* or *DATA_SET_COLLECTION_ID* value specified in the *DATASET.CAT* file.

1.8 **Appendix D - Report Examples**

Report Examples

This section details the various report formats available from the Catalog Ingest Tool.

1.8.1 Compare Report

Compare Report

The following is an example of a compare report:

```
PDS Catalog Ingestion Tool Report

Version           0.2.0
Date              Fri, May 29 2009 at 01:23:36 PM

Parameter Settings:

Mode              Compare
Target(s)
  Source = file:/c:/catalog-files/diff/dir1/
  Target = file:/c:/catalog-files/diff/dir2/
Severity Level    WARNING

Summary:

[X] Differences Found
[ ] No Differences Found

Compare Details:

DIFFERENT: file:/c:/catalog-files/diff/dir2/DATASET.CAT
  WARNING line 2: Element RECORD_TYPE has different value than source
    Source: line 13: file:/c:/catalog-files/diff/dir1/DATASET.CAT
    13c2
    < FIXED_LENGTH
    > STREAM

  WARNING line 3: Element LABEL_REVISION_NOTE has different value than source
    Source: line 2: file:/c:/catalog-files/diff/dir1/DATASET.CAT
    12d13
    < 2005-07-29 RS:simpson incorporated aerobraking data

  WARNING line 31: Element ARCHIVE_STATUS not present in source
  WARNING line 28: Element DATA_SET_TERSE_DESC not present in source
  WARNING line 32: Element CITATION_DESC not present in source
  ERROR Object DATA_SET_TARGET not present in target
    Source: line 774: file:/c:/catalog-files/diff/dir1/DATASET.CAT
  WARNING line 782: Object DATA_SET_TARGET not present in source

SAME: file:/c:/catalog-files/diff/dir2/MISSION.CAT
```

End of Report

1.8.2 Validate Report

Validate Report

The following is an example of a validate report:

```
PDS Catalog Ingestion Tool Report

Version           0.2.1
Date              Mon, Aug 17 2009 at 02:46:30 PM
Dictionary version
/* Planetary Science Data Dictionary database dump */
/* Start of alias definitions */
/* Version: OPS */
/* Online Database: pdscatlr75 */
/* Generated: Mon Jul 20 11:09:38 2009 */

Parameter Settings:

Mode              Validate
Target            file:/C:/catalog-files/goodcatfiles/
Dictionary File(s) [c:\pdsdd\pdsdd.full]
Allrefs File      file:/C:/catalog-files/goodcatfiles/allrefs_lr74.out
Severity Level    WARNING
Report File       report-validate.txt

Summary:

6 of 7 validated, 1 skipped
5 of 6 passed

4 referential integrity check(s) made
1 of 4 passed

0 new standard value(s) found

Validation Details:

SKIP: file:/C:/catalog-files/goodcatfiles/allrefs_lr74.out
      WARNING Not a label. Could not find the PDS_VERSION_ID in the first line.

PASS: file:/C:/catalog-files/goodcatfiles/MK88-120DS.CAT

PASS: file:/C:/catalog-files/goodcatfiles/MK88-120DS_REF.CAT
```

```

WARNING line 17: Could not find year at the end of the citation. Found 557-578.
WARNING line 24: Could not find year at the end of the citation. Found
http://www.ifa.hawaii.edu/88inch/manuals/user.pdf.
ERROR line 24: REFERENCE_DESC value does not start with UHGUIDE
WARNING line 31: Could not find year at the end of the citation. Found #2059.
ERROR line 31: REFERENCE_DESC value does not start with PIETERSANDPRATT

```

```
PASS: file:/C:/catalog-files/goodcatfiles/MK88-120_INM.CAT
```

```

FAIL: file:/C:/catalog-files/goodcatfiles/MK88-120_NSSDC.CAT
ERROR line 5: Undefined Object: NSSDCDSID

```

```
PASS: file:/C:/catalog-files/goodcatfiles/MK88_INST.CAT
```

```
PASS: file:/C:/catalog-files/goodcatfiles/MK88_INSTHOST.CAT
```

Referential Integrity Details:

```

FAIL: Data Set Integrity
  Parent File(s): [MK88-120DS.CAT]
  Begin checking children
  End checking children
    line 7: MK88-120DS.CAT: MK88-L-120CVF-3-RDR-120COLOR-V1.0 is not specified in
another catalog file.

FAIL: Reference Integrity
  Parent File(s): MK88-120DS_REF.CAT
    line 8: MCCORDETAL not found in allrefs dictionary.
    line 17: Description for PIETERS1986 does not match description in allrefs
dictionary.
      Source: allrefs_lr74.out
      < Pieters, C. M. (1986), Composition of the Lunar Highland Crust from
Near-Infrared Spectroscopy, Reviews of Geophysics, Vol. 24, No. 3, 557-578.
      > Pieters, C. M. (1986), Composition of the Lunar Highland Crust from
Near-Infrared Spectroscopy Reviews of Geophysics, Vol. 24, No. 3, 557-578.

  Begin checking children
    line 102: MK88-120DS.CAT: MCCORDETAL1981 is not found in the parent.
    line 33: MK88_INST.CAT: MCCORDETAL1981 is not found in the parent.
  End checking children
    line 8: MK88-120DS_REF.CAT: MCCORDETAL is not specified in another catalog file.

PASS: Instrument Integrity
  Parent File(s): [MK88_INST.CAT]
  Begin checking children
  End checking children

FAIL: Instrument Host Integrity
  Parent File(s): [MK88_INSTHOST.CAT]
  Begin checking children
  End checking children
    line 6: MK88_INSTHOST.CAT: MK88 is not specified in another catalog file.

End of Report

```



1.8.3 Ingest Report

Ingest Report

The following is an example of ingest report:

```
PDS Catalog Ingestion Tool Report

Version           0.2.0
Date              Mon, Jun 08 2009 at 01:48:42 PM

Parameter Settings:

Mode              Ingest
Target            file:/Users/hyunlee/dev/pds_en/citool/catfiles/test1/
Severity Level    WARNING

Summary:

Catalog Ingestion is completed.
  Number of processed files: 7
  Number of successful ingestion to the table: 27
  Number of failed ingestion to the table: 1

Catalog Ingestion Details:

Failed to parse the file. Run tool in validation mode for more details
  file:/Users/hyunlee/dev/pds_en/citool/catfiles/test1/DSColl.CAT

Completed ingestion of 'MISSION' catalog object.

ERROR: PERSINFO - NODE-USGS is not a valid PDS_AFFILIATION (not found in nodeinfo
table).

ABORT: PERSINFO - one or more of the required keywords was not found.

      - PDS_USER_ID: MGARCIA

      - LAST_NAME: GARCIA

      - FULL_NAME: MARIO GARCIA

      - TELEPHONE_NUMBER: 9285567246

      - ALTERNATE_TELEPHONE_NUMBER: 9285567012

      - FAX_NUMBER: 9285567090
```


- INSTITUTION_NAME: UNITED STATES GEOLOGICAL SURVEY
- NODE_ID: IMAGING
- REGISTRATION_DATE: 1996-09-26
- PDS_ADDRESS_BOOK_FLAG: null
- PDS_AFFILIATION: NODE-USGS
- LABEL_REVISION_NOTE: 2005-06-07 JHO, TEST PERSONNEL TEMPLATE

Completed ingestion of 'PERSONNEL' catalog object.

Completed ingestion of 'INSTRUMENT' catalog object.

Completed ingestion of 'DATA_SET' catalog object.

Completed ingestion of 'INSTRUMENT_HOST' catalog object.

Completed ingestion of 'REFERENCE' catalog object.

End of Report