Label Template Design Tool v.1.0.0

for the Planetary Data System

NASA Jet Propulsion Laboratory,
California Institute of Technology

24 March 2008

Table of Contents

1.1 Overview

About Label Template Design Tool

The Label Template Design Tool (LTDTool) is intended for designing label templates and is not intended to generate multiple labels in a production mode. The tool will generate a label template that is compliant with the latest PDS standards and data dictionary using an interactive label editor that gets input from the user, the PDS data dictionary, and the PDS standards, and outputs a PDS label template file that can be used in the creation of PDS data product labels. The user does not need to be a PDS expert because the tool guides the user through the label creation process.

1.2 Release Notes

Release Notes

The purpose of this section is to provide a description of a Label Template Design 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.

Release 1.0.0

This release of the Label Template Design Tool represents the operational release for the Phase I targeted capabilities.

The major changes for this release are as follows:

- Line and Column Indication

  The Template Editor pane now indicates the position of the cursor within the template by displaying the line number and column number at the bottom of the pane.

- Working Data Dictionary (WDD) Upgrades

  The objects and elements contained in the WDD will now have a status type of PROPOSED. The object type for object definitions can now be specified and defaults to SPECIFIC. Included a check for new object and element names to ensure that duplicate definitions are not created.

The liens for this release are as follows:

- Export Specific Label Object Definitions (SLODs)

  This capability has yet to be implemented.

- Support for Label Fragments

  This capability has yet to be implemented.

- Miscellaneous Issues

  The following miscellaneous issues have yet to be resolved:

  - When dragging from the Data Dictionary Listing pane the user must click the object or element twice in order to drag. Should be able to click once and drag.
  - When resizing the application window, only the Template Editor should increase in size. In this release, the dictionary panes increase in size.

- A project must be uniquely named even when saving in a different directory.
- Print function has not been implemented yet
- When using the quit function from the Mac OS X menu bar, the application exits without saving changes.
- The inclusion of <CR><LF> characters at the end of each line in a template causes quirky behavior on the Mac OS X platform. If content is appended after these characters on a line and the template is reformatted, the appended content will be moved to the next line.
- The tool does not support adding or modifying standard value lists in the WDD where the values contain commas. The only known example of this is the BAND_SEQUENCE element.

Release 0.4.0

This release of the Label Template Design Tool represents the binary executable release for the acceptance test phase with the PDS Technical Staff.

The major changes for this release are as follows:

- Copy/Paste Objects/Elements from Dictionary to Template

  Modified the drag and drop capability from the Data Dictionary listing to recursively include required sub-objects and elements in an Object block. In both cases where objects and elements are dragged to the template, substitution variables are included for the elements.

- Template Validation Report

  A new report was created in VTool to be used specifically by LTDTool for template validation. This report does not have the header that is found in the other validation reports.

- Template Reformatting

  The reformatting function was refined to position the "=" two characters past the longest element name up to a maximum predefined position. In addition, lines will be wrapped at position 78 with subsequent lines of a wrapped value being indented 2 positions. Also, all changes performed during reformatting are represented as one change as it pertains to the *Undo* function. The cursor position in the template prior to reformatting is retained approximately when a template is reformatted.

- Template Skeletons

  New templates were added for IMAGE and TABLE objects. All templates now validate successfully without modification.

- Menu Restructure

  A new project-related dialog window was added for creating new projects or opening existing projects. The menu options in the *File* menu were modified accordingly to dislay this new dialog. Also removed the cascading menus and provided more explicit menu options. Removed the *Export Label* and *From URL* menu options. Added more keyboard shortcuts.

- Data Dictionary Listing

  The pane is now organized by tabs with the inclusion of a new tab titled *Both* which will display Objects and Elements together. The selected object/element now continues to be selected when the active query is cancelled.

- Miscellaneous Issues
  The following miscellaneous issues were resolved:
  - Default pane sizing was modified so that the dictionary related panes are 1/3 of the total width and the Validation Status pane is 1/3 of the total height.
  - Lines are now wrapped in both the Validation Status and Definition Details panes to alleviate the need to horizontally scroll.
  - Incorporated wait features when performing functions that take a few seconds to complete.
  - Updated the menu bar icons.
  - Added the Help function to display the version, build date and copyright statement for the application.
  - Modified the startup scripts to auto-detect the JAR file alleviating the need for the user to modify the scripts.
  - Fixed typos and grammatical errors in a number of messages.

The liens for this release are as follows:

- Export Specific Label Object Definitions (SLODs)

  This capability has yet to be implemented.

- Support for Label Fragments

  This capability has yet to be implemented.

- Miscellaneous Issues
  The following miscellaneous issues have yet to be resolved:
  - When dragging from the Data Dictionary Listing pane the user must click the object or element twice in order to drag. Should be able to click once and drag.
  - When resizing the application window, only the Template Editor should increase in size. In this release, the dictionary panes increase in size.
  - A project must be uniquely named even when saving in a different directory.
  - Print function has not been implemented yet
  - When using the quit function from the Mac OS X menu bar, the application exits without saving changes.
  - The inclusion of <CR><LF> characters at the end of each line in a template causes quirky behavior on the Mac OS X platform. If content is appended after these characters on a line and the template is reformatted, the appended content will be moved to the next line.

- The tool does not support adding or modifying standard value lists in the WDD where the values contain commas. The only known example of this is the BAND_SEQUENCE element.
- The STATUS_TYPE for elements and objects in the WDD should be set to PROPOSED.

Release 0.3.0

This release of the Label Template Design Tool represents the binary executable release for the beta test phase with the PDS Technical Staff.

The major changes for this release are as follows:

- Undo/Redo Capabilities

  Added the capabilities to undo and redo changes made to the label template in the Template Editor pane.

- Copy/Paste Objects/Elements from Dictionary to Template

  Modified the drag and drop capability from the Data Dictionary listing by including the Object block structure and required elements for Objects as well as the addition of an "=" character for Elements.

- Attributes for Element Definitions

  Modified the Element definition listing and the Data Dictionary add/modify dialog by adding the Standard Value Type and Standard Value Set attributes.

- Extra white space on template lines

  Corrected a problem where extra white space is showing up at the end of lines of an imported label template causing the line numbers in the validation report to not be in sync with the template. Also corrected a problem where lines were double-spaced in the Template Editor pane.

- Initial window pane sizes need work

  Corrected a problem where the Validation Status pane is maximized where the Template Editor pane should be.

- Miscellaneous Issues
  The following miscellaneous issues were resolved:
  - Added menu option quick keys (Alt+key) for the Windows platform.
  - Problem handling relative paths references on the command-line.
  - Problem with attribute values not being forced to upper case.
  - Problem with index out of bounds error occurring during reformatting.

The liens for this release are as follows:

- Copy/Paste Objects/Elements from Dictionary to Template

  Although the capability to copy and paste objects and elements from the data dictionary listing has been added in this release, it still needs a little work. For Objects, only required elements are included but not

required sub-objects and their required elements. For Elements, we included the "=" character but it has been suggested that we include an appropriate substitution variable.

- Export Data Dictionary Submissions

  Although the capability to add and modify object and element definitions for the Working Data Dictionary (WDD) has been added in this release, the capability to export this information has yet to be implemented.

- Export Specific Label Object Definitions (SLODs)

  This capability has yet to be implemented.

- Miscellaneous Issues
  The following miscellaneous issues have yet to be resolved:
  - Print function has not been implemented yet
  - Help function has not been implemented yet
  - Quit function on the Mac OSX menu bar not supported yet

    When using the quit function from the Mac OSX menu bar the tool exits without saving changes.

  - Menu Management has not been implemented yet

    There are scenarios where certain menu options are not valid given the current state of the tool. In these scenarios those menu options should be grayed out.

  - Documentation

    The tool still needs documentation regarding how the tool should be used to generate a label template.

Release 0.2.0

This release of the Label Template Design Tool represents the binary executable release for the alpha test phase with the PDS Technical Staff.

The major changes for this release are as follows:

- Initiate a Design Session from an Existing Label or Template

  Added the capability to initiate a label template design session from an existing label or template via file or URL reference. The tool also includes a set of its own templates which can be utilized in a design session.

- Import Data Dictionaries

  Added the capability to import one or more PDS compliant data dictionaries via file or URL reference.

- Validate Template

  Added the capability to validate a label template. In order to validate a label template, the template is converted into a test label and then passed to the validation function.

- Reformat Template

  Added the capability to reformat the active label template. For this release the reformating is limited to lining up the equals signs, proper indenting of nested objects/elements and inserting appropriate spacing after comments.

- Export Template and Test Label

  Added the capability to export the label template and a test label for the template.

The liens for this release are as follows:

- Copy/Paste Objects/Elements from Dictionary to Template

  Although the capability to copy and paste obects and elements from the data dictionary listing has been added in this release, it is a very simple implementation which only copies the name of the object or element to the template.

- Export Data Dictionary Submissions

  Although the capability to add and modify object and element definitions for the Working Data Dictionary (WDD) has been added in this release, the capability to export this information has yet to be implemented.

- Export Specific Label Object Definitions (SLODs)

  This capability has yet to be implemented.

- Miscellaneous Issues
  The following miscellaneous issues have yet to be resolved:
  - Print function has not been implemented yet
  - Undo function has not been implemented yet
  - Help function has not been implemented yet
  - Quit function on the Mac OSX menu bar not supported yet

    When using the quit function from the Mac OSX menu bar the tool exits without saving changes.

  - Menu Management has not been implemented yet

    There are scenarios where certain menu options are not valid given the current state of the tool. In these scenarios those menu options should be grayed out.

  - Extra white space on template lines

    Extra white space is showing up at the end of lines of an imported label template causing the line numbers in the validation report to not be in sync with the template.

  - Initial window pane sizes need work

    On the Mac, the Validation Status pane is maximized where the Template Editor pane should be.

Release 0.1.0

This release of the Label Template Design Tool represents the source code release for the Code Walk-through. Subsequent releases will be accompanied with a little more detailed description.

1.3 Concept

Concept

This section is a bit of a hodgepodge right now but it attempts to provide some insight into the concept of the Label Template Design Tool (LTDTool) and how it is intended to be utilzed. Although there are a couple of Node-specific tools that generate labels from templates, there is not a generic tool that can be used by all of the Nodes. The intent of this tool is to generate a template that could work in those tools and be generic enough to work in other environments. Some representative templates have been provided with the tool, but it is up to the user to create and design their template and to determine which elements should have substitution variables and which should have static values. Substitution variables can be represented as follows:

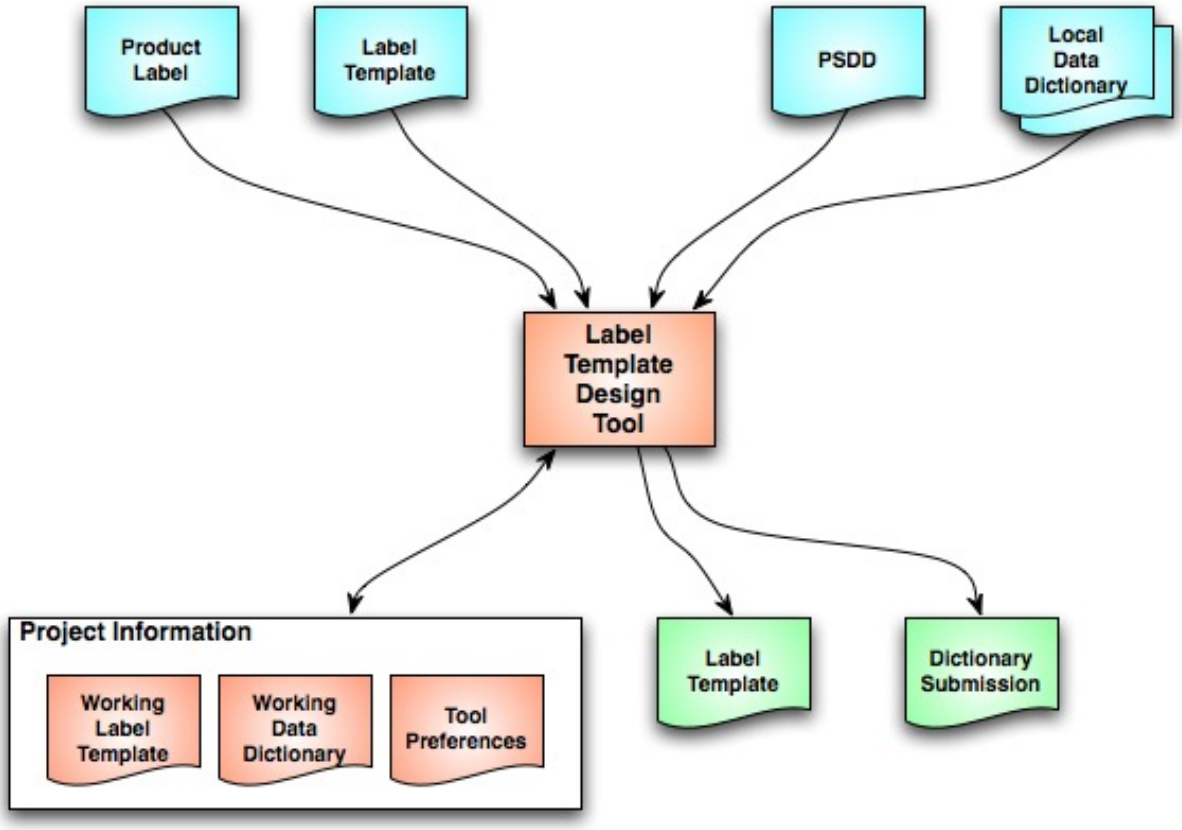
```
PRODUCT_ID                  = "${PRODUCT_ID}"

START_TIME                  = ${START_TIME}
```

In the example above the *PRODUCT_ID* element variable is quoted whether the resulting value requires quotation or not. For the *START_TIME* element, the variable is not quoted because date/time values may not be quoted according to the PDS standards. This goes for numeric values as well assuming the data type of the element has not been defined as a character string.

Inputs/Outputs

The following diagram details the inputs and outputs for LTDTool:



The inputs and outputs for the tool are described below.

Inputs

Although the tool will let the user read in just about any file to initiate a design session, the suggested inputs include PDS data products labels or previously created label templates. The current release does not perform any sort of file recognition, so it is up to the user to provide appropriate inputs. The user may also copy content from another application and paste it into the Template Editor pane as a form of input.

The tool also allows the user to import multiple PDS compliant data dictionaries. The assumption is that the user will load one instance of the Planetary Science Data Dictionary (PSDD) and any number of local data dictionaries appropriate for the design session. In this case, the tool will validate the contents of the data dictionary files. If an error occurs parsing the data dictionary file, the user will be notified. The Data Dictionary Listing pane will display the combined list of objects, elements or both (depending on the tab selected) for all of the data dictionaries imported.

The imported data dictionaries serve two purposes. The first is to provide a reference for adding objects and elements to the template in the current design session. The second is to be passed as an input to the validation function. This enables the validation function to compare the objects and elements found in the

template against definitions contained in the imported data dictionaries.

Outputs

Although not meant for public consumption, the tool stores the following files in the project directory (when a project is saved):

- Working Label Template

  This file contains the saved contents of the Template Editor pane.

- Working Data Dictionary

  This file contains any object or elements definitions that have been added or modified during the current design session.

- Tool Preferences

  This files contains any preferences specified by the user during the current design session (i.e., file specifications for the imported data dictionaries).

The main product of this tool is the label template itself. The tool allows the user to export the label template, for the current design session, to a file and directory specified by the user. For the current release, the working label template in the project directory is essentially the same file that is exported via the *Export Template* function. This may not be the case in future releases.

A secondary product of this tool is the working data dictionary. The tool allows the user to export the working data dictionary, for the current design session, to a file and directory specified by the user. For the current release, the working data dictionary in the project directory is essentially the same file that is exported via the *Export WDD* function. This may not be the case in future releases. The purpose of the WDD is described in the Working Data Dictionary section.

The tool stores one more file that is not detailed in the diagram above. This file is a properties file named *.ltdt* and is stored in the user's home directory. This file contains the names of the projects and their directory loations.

Working Data Dictionary

The intent of the Working Data Dictionary (WDD) is to enable the user to augment either the Planetary Science Data Dictionary (PSDD) or a Local Data Dictionary (LDD) to accomodate a label template that is being designed for a new data product. Designers of label templates are encouraged to utilize the object and elements defined in the PSDD or any LDDs, but it is expected for example that new elements or at least new values for a defined element may be necessary when designing a new label template. Users should be aware that any new or modified definitions captured in the WDD during a design session will need to be submitted to and approved by the PDS Standards body. Hence the need for the function to export a WDD to a file. The subsections below detail the data dictionary related changes that are allowed to be captured in the WDD.

New Object or Element

New object or element definitions intended for the PSDD or an LDD may be created and stored in the WDD. A minimal definition would be required for each object or element:

- Object Related Attributes:
  - Object Name

    The name of the object must be less than 61 characters in length consisting of <namespace>:<name>. Both the *namespace* and *name* components must contain all capital letters, numbers or underscores. Objects defined in the PSDD have an implied *namespace* of *PSDD*.

  - Object Type

    The valid values are *SPECIFIC* and *GENERIC*.

  - Required and optional sub-objects

    Any required or optional sub-objects must be defined in one of the imported data dictionaries in order for validation to work properly.

  - Required and optional elements

    Any required or optional sub-objects must be defined in one of the imported data dictionaries in order for validation to work properly.

  - Description

    A description of the object in free-form text.

- Element Related Attributes:
  - Element Name

    The name of the element must be less than 61 characters in length consisting of <namespace>:<name>. Both the *namespace* and *name* components must contain all capital letters, numbers or underscores. Elements defined in the PSDD have an implied *namespace* of *PSDD*.

  - Standard Value Type

    The valid values are *DYNAMIC*, *NONE*, *RANGE*, *STATIC* and *SUGGESTED*.

  - Standard Value Set

    A comma separated list of values corresponding to the defined general data type.

  - Minimum / Maximum

    The minimum and maximum values for number-based general data types.

  - Minimum Length / Maximum Length

    The minimum and maximum lengths for character-based general data types.

- General Data Type

  The valid values are *CHARACTER*, *CONTEXTDEPENDENT*, *DATE*, *IDENTIFIER*, *INTEGER*, *NONDECIMAL* and *REAL*.

- Unit ID

  The unit designation (e.g., KM, DEG, NONE, etc.).

- Description

  A description of the element in free-form text.

New objects are assumed to be specific objects instead of generic objects. For more information on object and element definitions and data dictionaries in general see the Planetary Science Data Dictionary Document .

Modified Object or Element

Modified object or element definitions intended for the PSDD or an LDD may be updated and stored in the WDD. The following modifications are allowed:

- New optional sub-object for an object
- New optional element for an object
- An optional element designated as a required element for an object
- New value or extended value range for an element

Once an object or element modification has been saved to the WDD, that object or element will appear in the Data Dictionary Listing pane twice. One entry will be the original definition and the other will be the modified definition. They can be identified by the source specified in the Definition Details pane. When the validation function is performed on the label template, the definition in the WDD will override the original definition.

1.4 Installation

Installation

This section describes how to install the Label Template Design Tool (LTDTool) contained in the *ltdtool* package. The following topics can be found in this section:

- System Requirements
- Unpacking the LTDTool Package

System Requirements

Java Requirement

The LTDTool 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 1.4, so the tool will execute correctly under versions 1.4 or 1.5. The tool will probably function in a 1.6 environment as well, but at the time of release we have not had a chance to test in that environment.

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. At the time of this release, other Java environments have not been tested with the tool.

Data Dictionary Requirement

Release *1r64* or later of the Planetary Science Data Dictionary (PSDD) is required for the validation feature of 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 Lookup web page.

Unpacking the LTDTool Package

Download the *ltdtool* package from the PDS Software Download 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:

```
[node: ~] unzip ltdtool-1.0.0.zip
or
```

```
[node: ~] tar -xzvf ltdtool-1.0.0.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 *ltdtool-1.0.0* 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 the executable jar file containing the Label Template Design Tool software along with a batch and shell script for executing the tool.

- **docs/**

  This document directory contains a local web site with the Label Template Design Tool 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 jar file (ltdtool-1.0.0.jar) containing the Label Template Design Tool software.

There is one more step for UNIX users intending to use the shell script for executing the tool. The permission on the script must be set to executable. This can be accomplished with the following commands:

```
[node: ~] cd ltdtool-1.0.0/bin
[node: ~/ltdtool-1.0.0/bin] chmod 775 LTDTool
```

For those of you wondering why we didn't take care of this prior to release, it appears that Maven , our build tool, is resetting the file permissions when the creating the distributions packages. Go figure.

1.5 Use and Operation

Use and Operation

The Label Template Design Tool (LTDTool) provides a Graphical User Interface (GUI) enabling users to design PDS label templates. This release of the tool allows templates to be created from scratch, from built-in templates or from existing templates and labels that the user may have available.

This section describes how to use LTDTool. The following topics can be found in this section:

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 LTDTool, the user's environment must first be configured appropriately. This section describes how to setup the user environment on UNIX-based and Windows machines. In addition to reassembling the command-line examples into a single line as mentioned above, the commands for setting environment variables must not contain space or line continutation characters in the value for the variable.

UNIX-Based Setup

This section details the environment setup for UNIX-based machines providing four different methods:

- Specify the Shell Script on the Command-Line
- Set the CLASSPATH Environment Variable
- Specify the CLASSPATH on the Command-Line
- Specify the Jar on the Command-Line

Specify the Shell Script on the Command-Line

The preferred method is to specify the shell script, *LTDTool*, on the command-line. Setting the *PATH*

environment variable to the location of this script, enables the shell script to be executed from any directory location on the user's machine.

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

```
[node:~] setenv PATH ${PATH}:\
$HOME/ltdtool-1.0.0/bin
```

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

```
[node:~] LTDTool <command-line arguments>
```

Set the CLASSPATH Environment Variable

An alternative method is to set the *CLASSPATH* environment variable. The following commands demonstrate how to set this variable, by appending to its current setting.

The first example and preferred method for setting the variable, appends the executable jar file found in the *bin* directory:

```
[node:~] setenv CLASSPATH ${CLASSPATH}:\
$HOME/ltdtool-1.0.0/bin/ltdtool-1.0.0-app.jar

[node:~] echo $CLASSPATH
```

The second example separately appends the dependent jar files, found in the *lib* directory:

```
[node:~] setenv CLASSPATH ${CLASSPATH}:\
$HOME/ltdtool-1.0.0/lib/antlr-2.7.6.jar:\
$HOME/ltdtool-1.0.0/lib/commons-cli-1.0.jar:\
$HOME/ltdtool-1.0.0/lib/commons-collections-3.1.jar:\
$HOME/ltdtool-1.0.0/lib/commons-configuration-1.2.jar:\
$HOME/ltdtool-1.0.0/lib/commons-io-1.2.jar:\
$HOME/ltdtool-1.0.0/lib/commons-lang-2.1.jar:\
$HOME/ltdtool-1.0.0/lib/commons-logging-1.0.3.jar:\
$HOME/ltdtool-1.0.0/lib/product-tools-1.1.2.jar:\
$HOME/ltdtool-1.0.0/lib/ltdtool-1.0.0.jar

[node:~] echo $CLASSPATH
```

The second command in both of the examples above, will display the current value of the *CLASSPATH* variable. Please note that the value for the *CLASSPATH* variable may not contain space characters. Once the *CLASSPATH* is set, the tool can be executed with the following command:

```
[node:~] java gov.nasa.pds.ltdt.LTDTool <command-line arguments>
```

Specify the CLASSPATH on the Command-Line

An alternative method to setting the *CLASSPATH* variable 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:

```
[node:~] java -Djava.ext.dirs=$HOME/ltdtool-1.0.0/lib \
gov.nasa.pds.ltdt.LTDTool <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:

```
[node:~] java -jar \
$HOME/ltdtool-1.0.0/bin/ltdtool-1.0.0-app.jar \
<command-line arguments>
```

Windows Setup

This section details the environment setup for Windows machines providing four different methods:

- Specify the Batch File on the Command-Line
- Set the CLASSPATH Environment Variable
- Specify the CLASSPATH on the Command-Line
- Specify the Jar on the Command-Line

Specify the Batch File on the Command-Line

The preferred method is to specify the batch file, *LTDTool.bat*, on the command-line. Setting the *PATH* environment variable to the location of this file, enables the batch file to be executed from any directory

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:\ltdtool-1.0.0\bin
```

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

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

Set the CLASSPATH Environment Variable

An alternative method is to set the *CLASSPATH* environment variable. The following commands demonstrate how to set this variable, by appending to its current setting.

The first example and preferred method for setting the variable, appends the executable jar file found in the *bin* directory:

```
C:\> set CLASSPATH=%CLASSPATH%;\
c:\ltdtool-1.0.0\bin\ltdtool-1.0.0-app.jar

C:\> echo %CLASSPATH%
```

The second example separately appends the dependent jar files, found in the *lib* directory:

```
C:\> set CLASSPATH=%CLASSPATH%;\
c:\ltdtool-1.0.0\lib\antlr-2.7.6.jar;\
c:\ltdtool-1.0.0\lib\commons-cli-1.0.jar;\
c:\ltdtool-1.0.0\lib\commons-collections-3.1.jar;\
c:\ltdtool-1.0.0\lib\commons-configuration-1.2.jar;\
c:\ltdtool-1.0.0\lib\commons-io-1.2.jar;\
c:\ltdtool-1.0.0\lib\commons-lang-2.1.jar;\
c:\ltdtool-1.0.0\lib\commons-logging-1.0.3.jar;\
c:\ltdtool-1.0.0\lib\product-tools-1.1.2.jar;\
c:\ltdtool-1.0.0\lib\ltdtool-1.0.0.jar

C:\> echo %CLASSPATH%
```

The second command in both of the examples above, will display the current value of the *CLASSPATH* variable. Please note that the value for the *CLASSPATH* variable may not contain space characters. Once the *CLASSPATH* is set, the tool can be executed with the following command:

```
C:\> java gov.nasa.pds.ltdt.LTDTool <command-line arguments>
```

Another way of setting the CLASSPATH is via the Windows Control Panel . If viewing this document in PDF form, see the appendix for details on this method.

Specify the CLASSPATH on the Command-Line

An alternative method to setting the *CLASSPATH* variable 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:\ltdtool-1.0.0\lib \
gov.nasa.pds.ltdt.LTDTool <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:\ltdtool-1.0.0\bin\ltdtool-1.0.0-app.jar \
<command-line arguments>
```

Tool Execution

LTDTool can be executed in various ways. This section describes how to run LTDTool and how to generate reports, as well as the behaviors and caveats of the tool.

Command-Line Options

The following table contains command-line options available to LTDTool:

LTDTool Command-Line Options

-p, --project <path>	Specify the project directory of a previously saved design session to be resumed.
-d, --dict <.full files>	Specify the PDS compliant data dictionary(s) to be imported for use during the design session.
-c, --config <file>	Specify a configuration file to set the default values for LTDTool.
-h, --help	Display LTDTool usage.
-V, --version	Display LTDTool version.

Running LTDTool

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

- Initiate a Session from Scratch
- Initiate a Session from a Previous Project
- Initiate a Session with a Single Dictionary
- Initiate a Session with Multiple Dictionaries
- Changing Tool Behaviors With The Configuration File

Note: The command-line examples below are executing LTDTool via the batch/shell script. Alternatively, the *java -jar* or *java gov.nasa.pds.ltdt.LTDTool* command can be used to execute the tool as mentioned in the Tool Setup section. Make sure the user environment is configured properly prior to calling the tool with your preferred method.

Initiate a Session from Scratch

The following command demonstrates initiation of a design session with no arguments:

```
[node:~] LTDTool
```

Initiate a Session from a Previous Project

The following command demonstrates initiation of a design session from a previously saved project:

```
[node:~] LTDTool -p /home/user/template1/
```

Initiate a Session with a Single Dictionary

The following command demonstrates initiation of a new design session with the import of a single data dictionary:

```
[node:~] LTDTool -d pdsdd.full
```

Initiate a Session with Multiple Dictionaries

The following command demonstrates initiation of a new design session with the import of a multiple data dictionaries:

```
[node:~] LTDTool -d pdsdd.full, localdd.full
```

Changing Tool Behaviors With The Configuration File

A configuration file can be specified on the command-line to change the default behaviors of the tool and to also provide users a way to initiate a design session 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 initiation of a design session using a configuration file:

```
[node:~] LTDTool -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 LTDTool even if it is surrounded by quotes. A common example would be a Windows path name (e.g. c:\template1). Use the forward slash character instead (c:/template1) or escape

the backslash character (c:\\template1).

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 Flag	Valid Value(s)
ltdtool.project	-p	Specify previously saved project directory.
ltdtool.dict	-d	Specify dictionary files.

The following example demonstrates how to set a configuration file:

```
# This is a LTDTool configuration file

ltdtool.project = /home/user/template1/
```

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

```
-p /home/user/template1/
```

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

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

ltdtool.dict = pdsdd.full, localdd.full
```

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

```
-d pdsdd.full, localdd.full
```

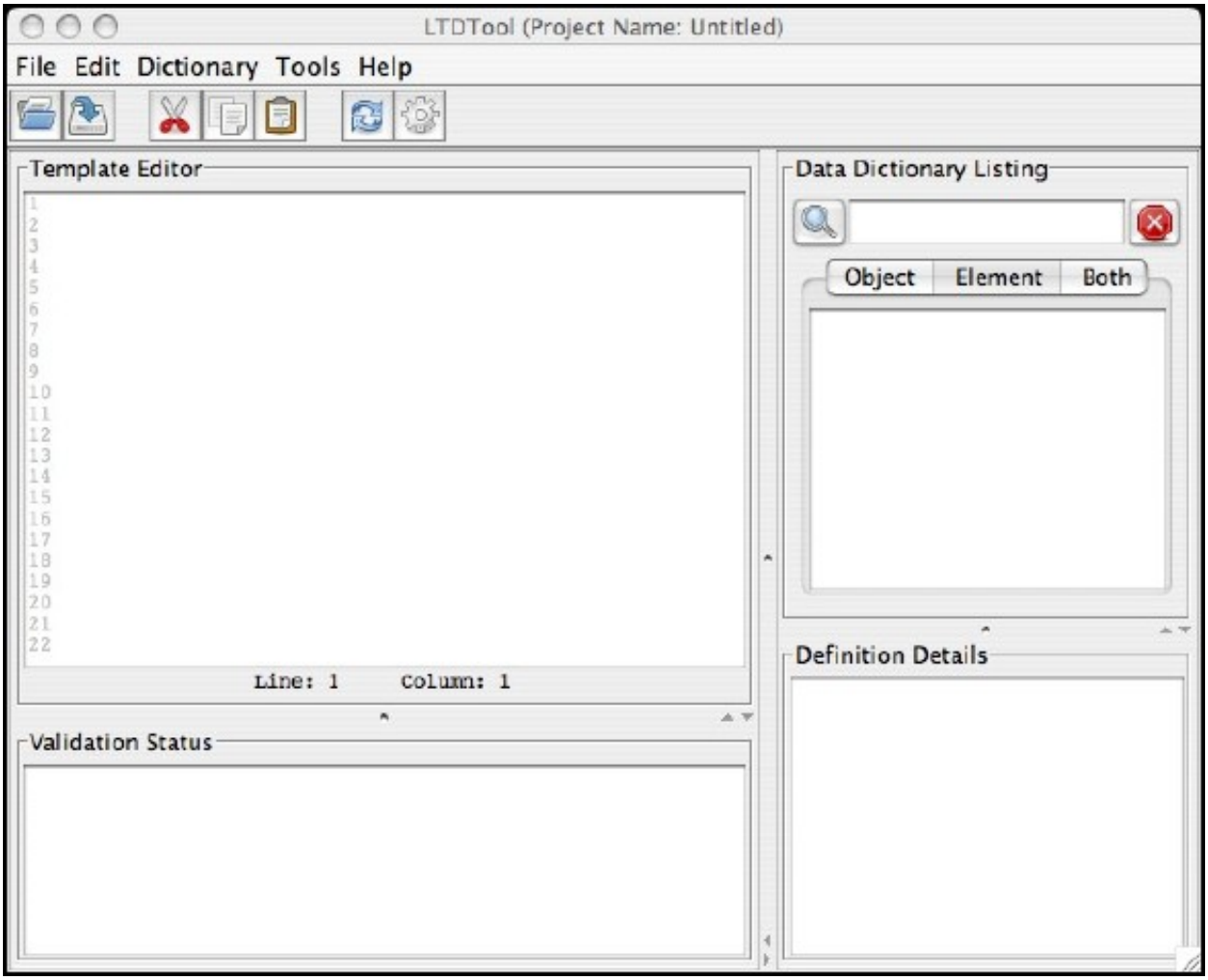
The following example demonstrates how to set a configuration file with multiple values that span across multiple lines:

```
# This is a LTDTool configuration file with multiple values
# that span across multiple lines

ltdtool.dict = pdsdd.full, \
               localdd.full
```

Tool Interface

The Graphical User Interface (GUI) of LTDTool allows the user to design one label template per session. After launching the tool with one of the methods described in the Tool Execution section above, the following window should appear on the user's desktop:



The main window of the tool consists of the following panes:

- Template Editor

  This pane contains the active label template for the current design session. This pane is populated through the *Create New Project* or *Open Project* menu options under the *File* menu.

- Validation Status

  This pane contains the validation report from the last request to validate the active label template. This pane is populated through the *Validate* menu option under the *Tools* menu.

- Data Dictionary Listing

  This pane contains the objects, elements or both from the PDS compliant data dictionaries that have been imported through the *Import* menu option under the *Data Dictionary* menu.

- Definition Details

  This pane contains the definition of the object or element currently highlighted in the *Data Dictionary Listing* pane. In an effort to alleviate the need to horizontally scroll in this pane, the <CR><LF> characters have been stripped from the object and element descriptions. Descriptions that contain paragraph breaks and bulleted lists will be a little harder to read without the added white space.

The main window also consists of the following menus:

- File
  - Create New Project...

    This menu option displays a dialog window enabling the user to create a new project from one of the provided templates in order to initiate a design session. The user may also create a new project by opening another label or template file from disk from within this dialog window. The user will be prompted to save any modifications made to the current project.

  - Open Project...

    This menu option displays a dialog window enabling the user to open a previously saved project in order to initiate a design session. The user will be prompted to save any modifications made to the current project.

  - Save Project

    This menu option saves the current state of the design session in the current project directory.

  - Save Project As...

    This menu option saves the current state of the design session in a new project directory. The name specified for the project becomes the of the directory where the project files will reside.

  - Close Project

    This menu option closes the project associated with the current design session. The user will be prompted to save any modifications made to the project.

  - Export Template...

This menu option exports the label template associated with the current design session to a file and directory specified by the user.

- Export WDD...

  This menu option exports the Working Data Dictionary (WDD) associated with the current design session to a file and directory specified by the user.

- Print

  This feature has not been implemented yet.

- Quit LTDTool

  This menu option exits the tool application. The user will be prompted to save any modifications made to the current project.

- Edit

  This menu consists of the standard options (Undo, Redo, Cut, Copy, Paste and Select All) which function as they do in any other application.

  There is one slight quirk regarding the use of the *Undo* function when it is performed immediately following the use of the *Reformat* function. All of the changes related to the reformatting of a template are considered one change when undoing them, but there is an intermediate step where the template is blanked out thus requiring the *Undo* function to be performed a second time to return the template back to its original format.

- Dictionary
  - Import...

    This menu option imports a PDS compliant data dictionary from a file.

  - Add Object...

    This menu option adds a new object definition to the Working Data Dictionary (WDD) for the current design session.

  - Add Element...

    This menu option adds a new element definition to the Working Data Dictionary (WDD) for the current design session.

  - Modify Object/Element...

    This menu option modifies the currently selected object or element definition from the *Data Dictionary Listing* pane and adds or modifies it in the Working Data Dictionary (WDD) for the current design session.

- Tools
  - Validate

This menu option generates a test label from the active label template and submits it for validation. The resulting validation report is displayed in the *Validation Status* pane. Because the substitution variables are replaced with representative values, some the validation messages may reference label content that was not part of the original template.

- Reformat

  This menu option reformats the active label template by aligning the equal signs, properly indenting nested objects/elements and inserting appropriate spacing after comments.

- Help
  - About LTDTool

    This menu option displays a message dialog detailing the current version of the tool along with the build date and copyright statement.

Common Errors

At this point in development there is a single common error that several users have encountered. The error is as follows:

```
[node:~] java gov.nasa.pds.ltdt.LTDTool ...
Exception in thread "main" java.lang.NoClassDefFoundError:
gov/nasa/pds/ltdt/LTDTool
```

The actual class name may vary but the above error is the result of the *CLASSPATH* environment variable not being set correctly. Verify that the variable contains all of the specified jar files and that there are no space characters in the value. See either the UNIX-Based Setup or Windows Setup section for details.

Mac OS X Quirks

There a couple of quirks to be aware of when executing LTDTool on the Mac OS X platform. The first is that the Max OS X platform provides the standard menu bar at the top of the desktop for Java applications. The *Quit* option from that menu should not be used to quit the application. Use of that *Quit* option will cause the application to exit without prompting the user to save any modifications to the current project. Please use the *Quit* option from the *File* menu within the LTDTool window which prompt the user to save any modifications and exit the application properly.

The other quirk is related to the fact that each line in a label template should be terminated with a combination of the <CR><LF> (carriage return and line feed) characters, which is a DOS/Windows artifact. The effect of this is that there appears to be an extra space at the end of each line in the label template. Before appending any content to a line, make sure the cursor is placed immediately after the last visible

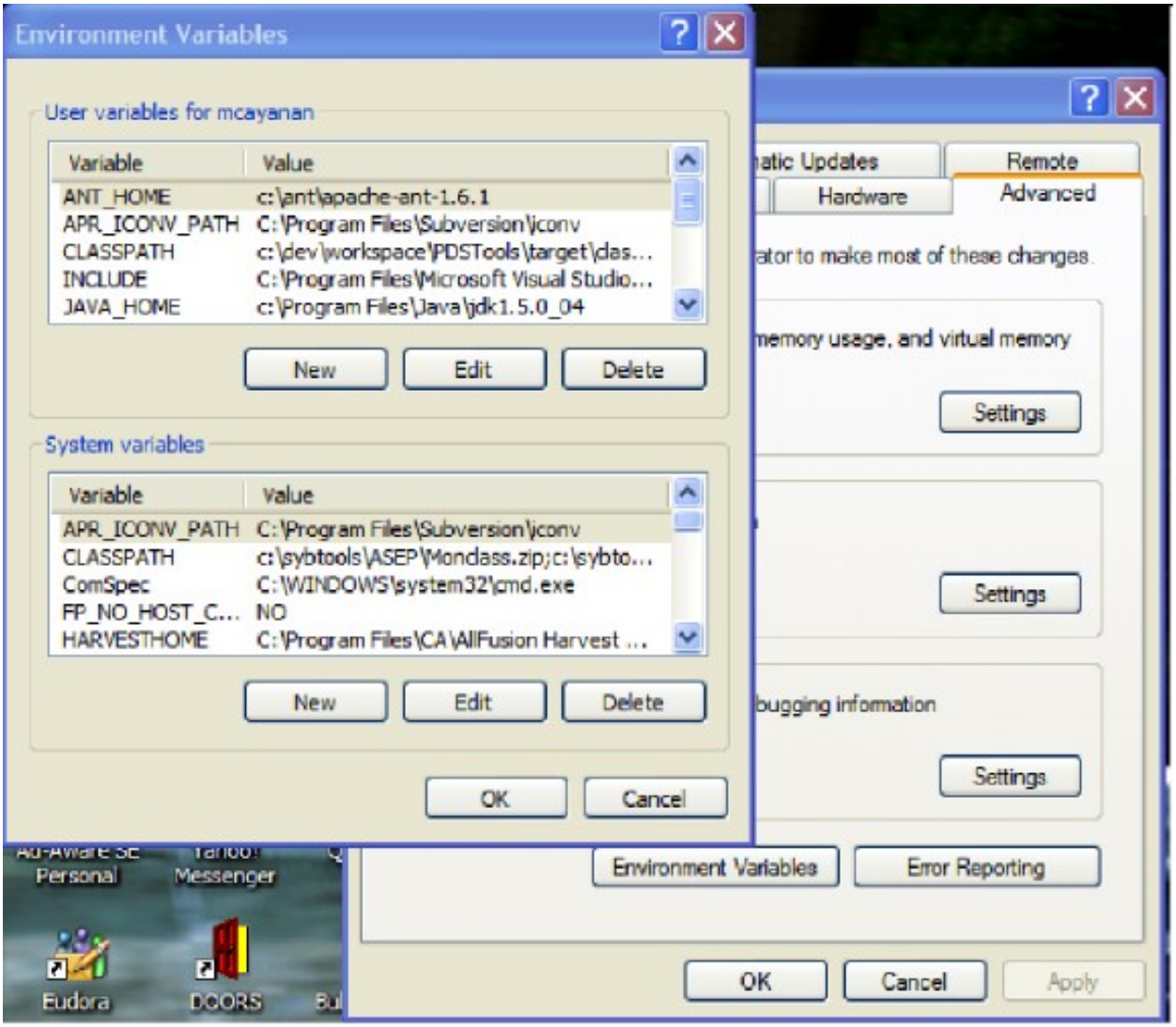
character in the line. The quirk is that if content is appended on a line after the <CR><LF> characters, it will get moved to the next line when the *Reformat* function is performed.

1.6 Appendix A - Using the Windows Control Panel

Windows Environment Setup

For those attempting to run the Validation Tool in a Windows environment, here are the steps for setting the CLASSPATH environment variable via the control panel:

- Right-click on My Computer 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 *CLASSPATH* variable in the User Variables list and select the Edit button.
- Append to the current contents of the variable, the path(s) to the jar files that came with the *product-tools* package. See the *Windows Setup* section for actual jar file paths for the current release. Separate each path with a semicolon.
- Select the *OK* button when you are finished editing the *CLASSPATH* 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 CLASSPATH changes to take effect.