Product Tools v.1.1.0

for the Planetary Data System

NASA Jet Propulsion Laboratory,
California Institute of Technology

28 September 2007

Table of Contents

1.1 Overview

About Product Tools

The next generation of Planetary Data System (PDS) Product Tools will support design/generation, validation and submission of archival products to the PDS. These tools are intended to replace the current generation of tools (e.g., lvtool, kwvtool, tbtool, etc.) used by the Discpline Nodes, Engineering Node and the PDS user community as a whole. This project consists of a library of software classes to support the development of the next generation tools.

1.2 Release Notes

Release Notes

The purpose of this section is to provide a description of a Product Tools 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 1.1.0

This is a maintenance release of the Product Tools software including fixes and changes requested by the early adopters of the application.

The major changes for this release are as follows:

- Moved the Validation Tool main class to the *vtool* project
  The Validation Tool main class has been split out from the *product-tools* project in order to provide more specific documentation and to turn the *product-tools* project into a library of classes that can be included by other projects.
- Improved data dictionary handling
  Many improvements and fixes were applied to the data dictionary classes. Many of these were initiated in order to support the Label Template Design Tool. They include better handling of aliases, support for namespaces in identifiers, access to all element definition fields, support for SPECIFIC_OBJECT_DEFINITION in GROUP definitions and the ability to write out a dictionary file.
- Elements may not be duplicated
  An element may not be duplicated within a given block of a label. An error message will be generated stating that multiple elements of the same name were found.
- Unit handling
  Messages regarding unknown unit values will now have a severity of WARNING instead of ERROR.
- Documentation Updates
  Added the release notes to the PDF version of the Product Tools Guide and included README files in the binary and source distributions. Also added documentation detailing the grammar to the Development section.

The liens for this release are the same as they were for release 1.0.0.

Release 1.0.0

This release of the Product Tools software, including the Validation Tool, represents the operational release for the Phase I targeted capabilities. The tool currently supports data product label validation including syntactic validation against the grammar and semantic validation against the PDS data dictionary.

The major changes for this release are as follows:

- Sub-objects supported in explicit FILE objects
  Although the tool didn't require any changes, SCR 3-1089 was approved and implemented in release 1r66 of the PSDD, enabling this capability.
- Report Fine Tuning
  The majority of changes for this release involved fine tuning the reporting capability. The parameters related to reporting in this release have been modified to support these changes.
- Executable Scripts and Exit Status Values
  Batch and shell scripts have been provided with this release allowing for simpler execution of the tool. Exit status values are now returned by the tool enabling execution from a script.

The liens for this release are as follows:

- Checking for Label Padding
  Validation of proper label padding has not been implemented yet.
- Summary Report Message Ordering
  The summary report still lists the messages in alphabetical order instead of the preferred chronological order.
- Date/Time Interpretation
  When running the tool under Java version 1.5, date/time values are periodically interpreted incorrectly generating an error message in the resulting report. This appears to be a bug in Java version 1.5, but we will research it further to determine a workaround or fix for the next release.
- Developer Guide
  Although a minimal *Development* guide is available in this release, it still needs content detailing the entry points for the API.

1.3 Installation

Installation

This section describes how to install the tools contained in the *product-tools* package. The following topics can be found in this section:

- System Requirements
- Unpacking the Tools Package

System Requirements

Java Requirement

The tools contained in the *product-tools* package were developed using Java and will run on any platform with a supported Java Runtime Environment (JRE). The tools were specifically developed under Sun Java version 1.4, so the tool will execute correctly under versions 1.4 or 1.5.

Since the tools are 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. Testing has been performed with GNU's Java environment. The result is that release 4.1.1 at a minimum is required for nominal operations.

Data Dictionary Requirement

Release *1r64* or later of the Planetary Science Data Dictionary (PSDD) is required for the tools 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 Tools Package

Download the *product-tools* package from the PDS Software Download web page. The binary and source distributions are available in identical zip or tar/gzip packages. Unpack the selected binary distribution file with one of the following commands (where X.X.X is the current version):

```
[node: ~] unzip product-tools-X.X.X.zip
or
```

```
[node: ~] tar -xzvf product-tools-X.X.X.tar.gz
```

Unpack the selected source distribution file with one of the following commands (where X.X.X is the current version):

```
[node: ~] unzip product-tools-X.X.X-src.zip
or
[node: ~] tar -xzvf product-tools-X.X.X-src.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.

Binary Distribution

The binary distribution related commands above result in the creation of the *product-tools-X.X.X* 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.

- **docs/**

  This document directory contains a local web site with the Product Tools 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 (product-tools-X.X.X.jar) containing the product tools software.

Source Distribution

The source distribution related commands above result in the creation of the *product-tools-X.X.X* directory with the following directory structure:

- **README.txt**
  A README file that will contain useful information on how to build the software in a future release.
- **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.
- **build.xml**
  The Ant script for building the product tools software.
- **maven.xml**
  The Maven goals for the product tools software.
- **project.properties**
  The Maven properties for the product tools software.
- **project.xml**
  The Maven Project Object Model (POM) for the product tools software.
- **defaults/**
  This directory contains the Maven default settings and site layout for building the product tools software.
- **src/**
  This directory contains the source code files for the product tools software.

1.4 Development

Development

This section describes how a user can utilize the Application Program Interface (API) of the *product-tools* package to develop their own tool. The following topics can be found in this section:

- API Entry Points
- ODL Grammar

API Entry Points

We have not had a chance to document the entry points into the API as of the current release. The best source for this information currently is to view the *VTool.java* file from the *vtool* project's source package. Another source of documentation would be the JavaDocs documentation for this project.

ODL Grammar

The parsing of PDS labels has been accomplished using the ANTLR software package. The links below detail the grammar for both the parser and the lexer, which were generated from the *src/resources/grammar/odl.g* source file. If viewing this document in PDF form, see the appendix for the actual grammar.

- Definition of parser ODLParser .
- Definition of lexer ODLLexer .

1.5 Appendix A - ODL Grammar

ODL Grammar

This section details the grammar for both the parser and the lexer, which were generated from the *src/resources/grammar/odl.g* source file.

1.5.1 ODLParser

ODLParser Grammar

The following represents the grammar for the *ODLParser* class:

```
Definition of parser ODLParser, which is a subclass of LLkParser.


label

        : (
statement

                | . ( ~EOL )* EOL
                )*
                ( END
                |
                )
        ;


statement

        :
simple_statement

        |
group_statement

        |
object_statement

        ;


simple_statement

        : ( COMMENT
                |
                )
                EOL
        |
assignment_statement

                ( COMMENT
                |
```

```
                )
                EOL
        |
pointer_statement

                ( COMMENT
                |
                )
                EOL
        ;


group_statement

        : "GROUP"
nl
 EQUALS
nl
 IDENTIFIER
                ( COMMENT
                |
                )
                EOL
                (
simple_statement

                | . ( ~EOL )* EOL
                )*
                END_GROUP
                ( EQUALS IDENTIFIER
                |
                )
                ( COMMENT
                |
                )
                EOL
        ;


object_statement

        : "OBJECT"
nl
 EQUALS
nl
 IDENTIFIER
                ( COMMENT
                |
                )
                EOL
                (
statement

                | . ( ~EOL )* EOL
                )*
                END_OBJECT
                ( EQUALS IDENTIFIER
```

```
                |
                )
                ( COMMENT
                |
                )
                EOL
        ;


assignment_statement

        : ( ELEMENT_IDENT
                | IDENTIFIER
                )

nl
 EQUALS
nl

value

        ;


pointer_statement

        : POINT_OPERATOR
assignment_statement

        ;


nl

        : ( EOL )*
        ;


value

        :
scalar_value

        |
sequence_value

        |
set_value

        ;


scalar_value

        :
numeric_value
```

```
        |
date_time_value

        |
text_string_value

        |
symbol_value

        ;


sequence_value

        :
sequence_2d

        |
sequence_1d

        ;


set_value

        : SET_OPENING
nl

item_list
 SET_CLOSING
        ;


numeric_value

        : INTEGER
                ( UNITS
                |
                )

        | BASED_INTEGER
                ( UNITS
                |
                )

        | REAL
                ( UNITS
                |
                )

        ;


date_time_value

        : DATE
        | TIME
```

```
        | DATETIME
        ;


text_string_value

        : QUOTED
        ;


symbol_value

        : IDENTIFIER
        | SYMBOL
        ;


sequence_2d

        : SEQUENCE_OPENING
nl

sequence_list
 SEQUENCE_CLOSING
        ;


sequence_1d

        : SEQUENCE_OPENING
nl

scalar_list
 SEQUENCE_CLOSING
        ;


scalar_list

        :
        |
scalar_value

nl
 ( ( LIST_SEPARATOR
nl

                        |
                        )

scalar_value

nl
 )*
        ;
```

```
sequence_list

        :
        |
sequence_1d

nl
 ( ( LIST_SEPARATOR
nl

                        |
                        )

sequence_1d

nl
 )*
        ;


item_list

        :
        |
scalar_value

nl
 ( ( LIST_SEPARATOR
nl

                        |
                        )

scalar_value

nl
 )*
        ;
```

1.5.2 ODLLexer

ODLLexer Grammar

The following represents the grammar for the *ODLLexer* class:

```
Definition of lexer ODLLexer, which is a subclass of CharScanner.

/** Lexer nextToken rule:
 *  The lexer nextToken rule is synthesized from all of the user-defined
 *  lexer rules.  It logically consists of one big alternative block with
 *  each user-defined rule being an alternative.
 */

mSET_OPENING

|
mSET_CLOSING

|
mSEQUENCE_OPENING

|
mSEQUENCE_CLOSING

|
mLIST_SEPARATOR

|
mPOINT_OPERATOR

|
mEQUALS

|
mCOMMENT

|
mEOL

|
mWS

|
mELEMENT_IDENT

|
```

```
mUNITS

|
mNUMBER_OR_DATETIME

|
mQUOTED

|
mSYMBOL

mSET_OPENING

        : '{'
        ;

mSET_CLOSING

        : '}'
        ;

mSEQUENCE_OPENING

        : '('
        ;

mSEQUENCE_CLOSING

        : ')'
        ;

mLIST_SEPARATOR

        : ','
        ;

mPOINT_OPERATOR

        : '^'
        ;

mEQUALS

        : '='
        ;

mCOMMENT

        : "/*"
                ( '*'
                |
mEOL

                | ( '*'
                        | '\n'
                        | '\r'
                        )
```

```
                )*
                "*/"
        ;

mEOL

        : ( '\r'
                        ( '\n'
                        |
                        )
                | '\n'
                )

        ;

mWS

        : ( ' '
                | '\t'
                | '\f'
                )

        ;

protected
mIDENTIFIER

        :
mLETTER

                (
mLETTER

                |
mDIGIT

                | '_'
                )*
        ;

protected
mLETTER

        : (                         'a'..'z'
                |                         'A'..'Z'
                )
        ;

protected
mDIGIT

        : (                         '0'..'9' )
        ;

mELEMENT_IDENT

        :
```

```
mIDENTIFIER
 ':'
mIDENTIFIER

        |
mIDENTIFIER

        ;

mUNITS

        : '<'
                (
mWS

                |
                )

mLETTER

                (
mLETTER

                |
mDIGIT

                |
mSPECIALCHAR

                | '('
                | ')'
                | '/'
                |
mWS

                )*
                '>'
        ;

protected
mSPECIALCHAR

        : ( '_'
                | '$'
                | '#'
                | '.'
                | '-'
                | ':'
                | '+'
                | '*'
                )
        ;

protected
mUNITS_FACTOR

        :
```

```
mIDENTIFIER

                (
mEXP_OP

mINTEGER

                |
                )
        ;

protected
mEXP_OP

        : "**"
        ;

protected
mINTEGER

        : (
mSIGN

                |
                )

mDIGITS

        ;

protected
mMULT_OP

        : ( '*'
                | '/'
                )
        ;

mNUMBER_OR_DATETIME

        :
mBASED_INTEGER

        |
mDATETIME

        |
mTIME

        |
mREAL

        |
mREAL

        |
mREAL
```

```
        |
mREAL

        |
mINTEGER

        ;

protected
mDIGITS

        : (
mDIGIT
 )+
        ;

protected
mBASED_INTEGER

        :
mDIGITS
 '#'
                ( '+'
                | '-'
                |
                )
                (
mEXTENDED_DIGIT
 )+ '#'
        ;

protected
mDATETIME

        :
mYEAR
 '-'
mMONTH
 '-'
mDAY
 'T'
mTIME

        |
mYEAR
 '-'
mDOY
 'T'
mTIME

        |
mDATE

        ;

protected
```

```
mTIME

        :
mHOUR

                ( ':'
mMINUTE

                        ( ':'
mSECOND

                                ( '.'
mFRACTION

                                |
                                )
                        |
                        )
                |
                )
                ( 'Z'
                |
                )
        ;

protected
mSIGN

        : '+'
        | '-'
        ;

protected
mREAL

        : (
mSIGN

                |
                )
                (
mDIGITS

                |
                )
                '.'
                (
mDIGITS

                |
                )
                ( ( 'E'
                        | 'e'
                        )

mINTEGER
```

```
                |
                )
        ;

protected
mEXTENDED_DIGIT

        :
mDIGIT

        |
mLETTER

        ;

protected
mYEAR

        :
mDIGIT

mDIGIT

mDIGIT

mDIGIT

        ;

protected
mMONTH

        :
mDIGIT

mDIGIT

        ;

protected
mDAY

        :
mDIGIT

mDIGIT

        ;

protected
mDOY

        :
mDIGIT

mDIGIT
```

```
mDIGIT

        ;

protected
mDATE

        :
mYEAR
 '-'
mDOY

        |
mYEAR

                ( '-'
mMONTH

                        ( '-'
mDAY

                        |
                        )
                |
                )

        ;

protected
mHOUR

        :
mDIGIT

mDIGIT

        ;

protected
mMINUTE

        :
mDIGIT

mDIGIT

        ;

protected
mSECOND

        :
mDIGIT

mDIGIT

        ;
```

```
protected
mFRACTION

        :
mDIGIT

                (
mDIGIT

                        (
mDIGIT

                        |
                        )
                |
                )
        ;

mQUOTED

        : '"'
                (
mEOL

                | ( ( '"'
                                | '\r'
                                | '\n'
                                ) )
                )*
                '"'
        ;

mSYMBOL

        : '\'' ( ( '\''
                        | '\\'
                        | '\r'
                        | '\n'
                        ) )* '\''
        ;

protected
mIGNORE

        : .
        ;
```