Update Tool v.0.1.0

for the Planetary Data System

NASA Jet Propulsion Laboratory,
California Institute of Technology

29 August 2011

Table of Contents

1.1 Overview

About Update Tool

The Update Tool provides a command-line application for transferring node logs from remote locations and for updating the Sawmill database for the associated profiles. This software also provides the core functionality for the Profile Setup web application.

Please send comments, change requests and bug reports to the PDS Operator at pds_operator@jpl.nasa.gov.

1.2 Release Notes

Release Notes

The purpose of this section is to provide a description of a Report Service release with respect to the Update Tool component including any impact that the new or modified capabilities will have on the Discipline Nodes or the PDS user community. If viewing the web-based version of this document, a somewhat itemized list of changes for each release can be found on the Release Changes page.

Release 0.1.0

This release of the Update Tool is a component of the integrated release 1.2.0 of the PDS 2010 System. This release is intended as a prototype release in support of the assessment of the PDS4 standards and the system components to date. This release includes initial capabilities for transferring logs and updating the Sawmill database.

1.3 Installation

Installation

This section describes how to install the Update Tool contained in the *rs-update* package. The following topics can be found in this section:

- System Requirements
- Unpacking the Package

System Requirements

The Update 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 1.6, so the tool will execute correctly under 1.6 and future versions.

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

Unpacking the Package

Download the *rs-update* package from the internal PDS Maven repository. 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 rs-update-0.1.0-bin.zip
or
% tar -xzvf rs-update-0.1.0-bin.tar.gz
```

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

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

- **README.txt**

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

- **LICENSE.txt**

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

- **bin/**

  This directory contains batch and shell scripts for executing the tool.

- **lib/**

  This directory contains the dependent jar files for the tool along with the executable jar file (rs-update-0.1.0.jar) containing the Update Tool software.

1.4 Operation

Operation

The following topics can be found in this section:

- Tool Setup
- Tool Execution

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

Tool Setup

In order to execute the Update Tool, 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, *RSUpdate*, 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/rs-update-0.1.0/bin
```

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

```
% RSUpdate <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, *RSUpdate.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:\rs-update-0.1.0\bin
```

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

```
C:\> RSUpdate <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

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

Command-Line Options

The following table describes the command-line options available:

Command-Line Option	Description
-h, --help	Display usage.
-p, --properties-home	Specify the directory path where the *.properties files are located. Default location is *../conf/* .
-n, --profile-name	Specify the name of a specific profile to update. Default updates ALL active profiles.
-s, --sawmill-off	Specify flag to turn off updating the Sawmill database after downloading the remote logs. Default = ON
-t, --transfer-off	Specify flag to turn off the downloading of remote logs. Default = ON

Execute Update Tool

This section demonstrates execution of the tool using the command-line options. The examples below execute the tool via the batch/shell script. Alternate methods for executing the tool can be found in the Tool Setup section.

The following command demonstrates how to run the Update Tool to update all Sawmill profiles by downloading the remote logs and performing a Sawmill database update using all default settings.

```
% RSUpdate
```

The following command demonstrates how to run the Update Tool with the output going to a log file, *log.txt* instead of standard out:

```
% RSUpdate -l log.txt
```

The following command demonstrates how to run the Update Tool for a specified profile only, i.e. *img-web*:

```
% RSUpdate -n img-web
```

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 Update Tool.

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=$HOME/rs-update-0.1.0/lib \
gov.nasa.pds.harvest.HarvestLauncher \
-Dpds.registry="<Registry URL>" \
-Dpds.security.keystore="$HOME/rs-update-0.1.0/keystore/tomcat_self_sign_keystore" \
<policy file> <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 $HOME/rs-update-0.1.0/lib/rs-update-0.1.0.jar \
-Dpds.registry="<Registry URL>" \
-Dpds.security.keystore="$HOME/rs-update-0.1.0/keystore/tomcat_self_sign_keystore" \
<policy file> <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 Update Tool.

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:\rs-update-0.1.0\lib \
gov.nasa.pds.harvest.HarvestLauncher -Dpds.registry="<Registry URL>" \
-Dpds.security.keystore="c:\rs-update-0.1.0\keystore\tomcat_self_sign_keystore" \
<policy file> <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:

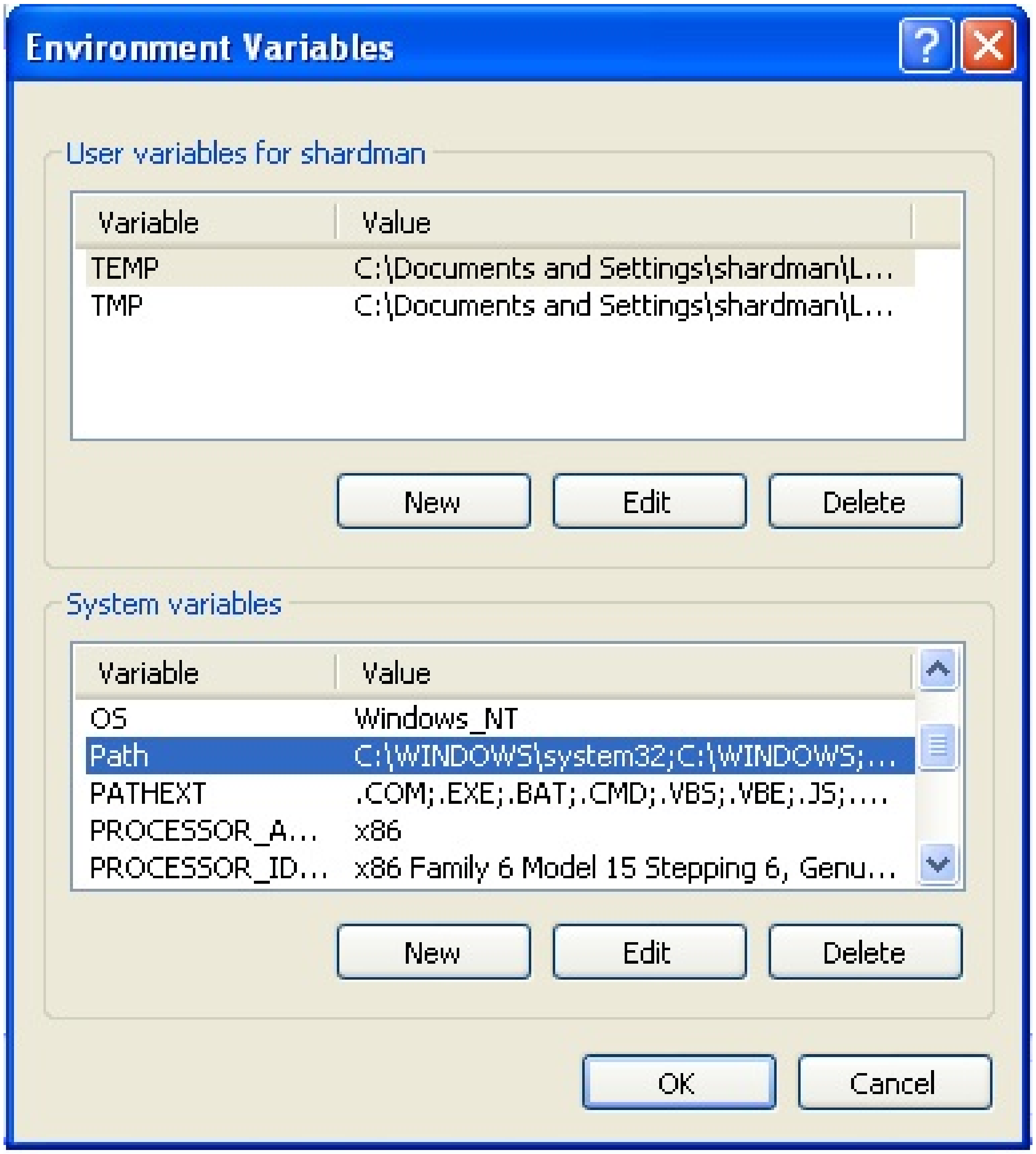
```
C:\> java -jar c:\rs-update-0.1.0\lib\rs-update-0.1.0.jar \
-Dpds.registry="<Registry URL>" \
-Dpds.security.keystore="c:\rs-update-0.1.0\keystore\tomcat_self_sign_keystore" \
<policy file> <command-line arguments>
```

Specify the Path in the Control Panel

The method for setting the executable path permanently for Harvest Tool 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:



User variables for shardman

Variable	Value
TEMP	C:\Documents and Settings\shardman\L...
TMP	C:\Documents and Settings\shardman\L...

System variables

Variable	Value
OS	Windows_NT
Path	C:\WINDOWS\system32;C:\WINDOWS;...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;....
PROCESSOR_A...	x86
PROCESSOR_ID...	x86 Family 6 Model 15 Stepping 6, Genu...

- 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 *rs-update* 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.