Standards Change Request

MD5 Checksums **scr3-1034.v2**

Todd King **November 14, 2005**

Provenance:

Date: 2004-11-22, revision 1.0
Working Group: J. Wilf (lead), T. King, M. McAuley
Title: MD5 Checksums for Files (SCR3-1034.v1)

Date: 2004-11-22, revision 1.0
Working Group: J. Wilf (lead), T. King, M. McAuley
Title: MD5 Checksums for Objects (SCR3-1035.v1)

This SCR was originally proposed in two parts, one for files and one for objects (as defined in a PDS label). The file SCR was SCR3-1034.v1 and the object SCR was SCR3-1035.v1. This SCR details a simplified and easier to implement approach to checksum generation and use while preserving the intent of the original SCRs.

Problem:

Ensure the integrity of the PDS archive and the delivery of products to users.

The PDS archive is maintained on media which has theoretical life spans. As the media ages it will be necessary to copy the archive to fresh media in order to maintain the archive. Also, as new storage media become available it may be desirable to migrate the archive to the new media. As media ages or during the migration process it is possible to have bit level changes which are difficult to detect. One method to simplify the detection of changes to a file is using checksums. One of the most reliable and tested checksums is the 5th generation Manifest Digest (MD5) algorithm.

An MD5 checksum [MD5] is calculated for a bit stream of any length. For the purposes of PDS this will be a physical file. An MD5 checksum is a 128-bit number, represented as a 32-character string of hexadecimal digits, e.g., 754b9db19f79dbc4992f7166eb0f37ce. The MD5 checksum specification states:

> It is conjectured that it is computationally infeasible to produce two messages having the same message digest, or to produce any message having a given pre-specified target message digest.

In other words, no two files will have the same MD5 checksum unless they are identical. This has been demonstrated to be true in all but the most extreme circumstances.

While the MD5 algorithm Is not recommend by the National Institute of Standards and Technology (NIST) for secure transmissions (SHA-1 is preferred) the MD5 checksum is faster to compute than the SHA-1 checksum and is well suited for integrity checking.

Only file level MD5 checksums are necessary even if the file contains separate "objects" that are describe in a PDS label. If the file is valid so are all objects in the file. File level checksums also work for both attached and detached labels. It is desirable to have checksums included in a label so that the product is self-contained. This is feasible only for referenced files (pointers) in detached labels since modifying the contents of a label (adding a checksum value) changes the checksum of the label. With attached labels this would change the checksum for the entire file (label plus data).

MD5 Support

The generation of MD5 checksums is widely supported. The algorithm with source code written in C is available at [MD5]. MD5 checksum generation is also support in Java. The tool MD5deep [MD5-Deep] supports the recursive generation of MD5 checksums and the validation of files. The output of the MD5deep tool is one line per file consisting of the MD5 checksum, whitespace and the file name (with path). An example is:

```
$ md5deep -lr .
f8dd7758cb5231c9e7817c4710d00b6e  ./aareadme.htm
d8b83365f5e117b9665181944889da3d  ./aareadme.lbl
1e8d45f622e09b9e2998af1a6d67a296  ./aareadme.txt
7dcfa51691ddd149a5a091ebe87b9bb1  ./errata.txt
7f310bf58a37af7f9b16c4fe68a131fb  ./voldesc.cat
```

Current or Planned Uses

Imaging (Myche McAuley, e-mail 8/24/2005): Would like to use file level MD5 in detached labels and in manifest for archives and data deliveries.

Geo (Susie Slavney, e-mail 8/30/2005): Has used MD5 checksums to verify Mars Express HRSC data received from ESA. Would like to use MD5 as part of the PDS system to ensure stability of archives.

PPI (Steven Joy, personal communication 8/2005): The Cassini SIS requires MD5 checksums to be included in all labels. Products are currently being submitted with MD5 checksums.

SBN () :

Proposed Solution:

1. Adopt the format of the manifest generated by the tool "md5deep" as the format for the PDS MD5 checksum manifest.

2. Place the manifest in the file called "md5sums.txt" in the root folder (directory) of a collection of files. For a volume this would be in the root directory of a volume, for an on-line collection it would be at the root directory of the collection.
3. Permit MD5 checksums to be included in the label metadata for each file reference (pointer) so that products can contain a level of self-referential integrity.

Requested Changes:

Changes to the Standards Reference
==========
The following changes to the PDS Standards Reference are required to support this SCR:

Add to Chapter 19.3.1 ROOT Directory Files

MD5SUMS.TXT **Optional**

This file contains an MD5 checksum for every file on the volume. The format is to be one line per file consisting of the MD5 checksum, space(s), and file name including relative path from the root of the volume.

Each figure in chapter Chapter 19. Volume Organization and Naming will need updating to include a "MD5SUMS.TXT" file in the root directory.

Changes to the Data Dictionary
==========
Modify the description of the MD5_CHECKSUM keyword currently in the dictionary to:

> The MD5 algorithm takes as input a bit stream (file) of arbitrary length and produces as output a 128-bit 'fingerprint' or 'message digest' of the input. It is conjectured that it is computationally infeasible to produce two messages having the same message digest, or to produce any message having a given pre-specified target message digest. In other words, no two files will have the same MD5 checksum unless they are identical. Any alterations to a file will result in a different checksum. An MD5 checksum can be used to
>
> Most standard MD5 checksum calculators return a 32 character hexadecimal value containing lower case letters. To accommodate this convention the PDS requires that the value assigned to the MD5_CHECKSUM keyword be a string characters enclosed in double quotes and consisting solely of lowercase letters (a-f) and numbers (0-9).
>
> Example: MD5_CHECKSUM = "0ff0a5dd0f3ea4e104b0eae98c87f36c"

Changes to the PDS Tool Suite
==========

There are no immediate changes necessary in any PDS tool. External validation of files in the PDS archive is possible with existing non-PDS tools. Support of internal checksums can be added to tools as the need arrises.

Impact Assessment:

The PDS Standards Reference will be modified as described above. In addition, there are the following impacts to the PDS System and its operation:

1. Data Providers will have to calculate and save the MD5 checksums as described above. Minimal impact since tools already exists to perform this operation.
2. Including MD5 checksums in on-line delivery of products. Minimal impact since the generation of MD5 checksums is support in Java which is the implementation language for OODT and DITDOS3.

Additional Information:

[MD5] The document describing the MD5 algorithm can be found at: http://www.faqs.org/rfcs/rfc1321.html

[MD5-DEEP] MD5deep resources: http://md5deep.sourceforge.net/