Validation Users Guide

Table of Contents

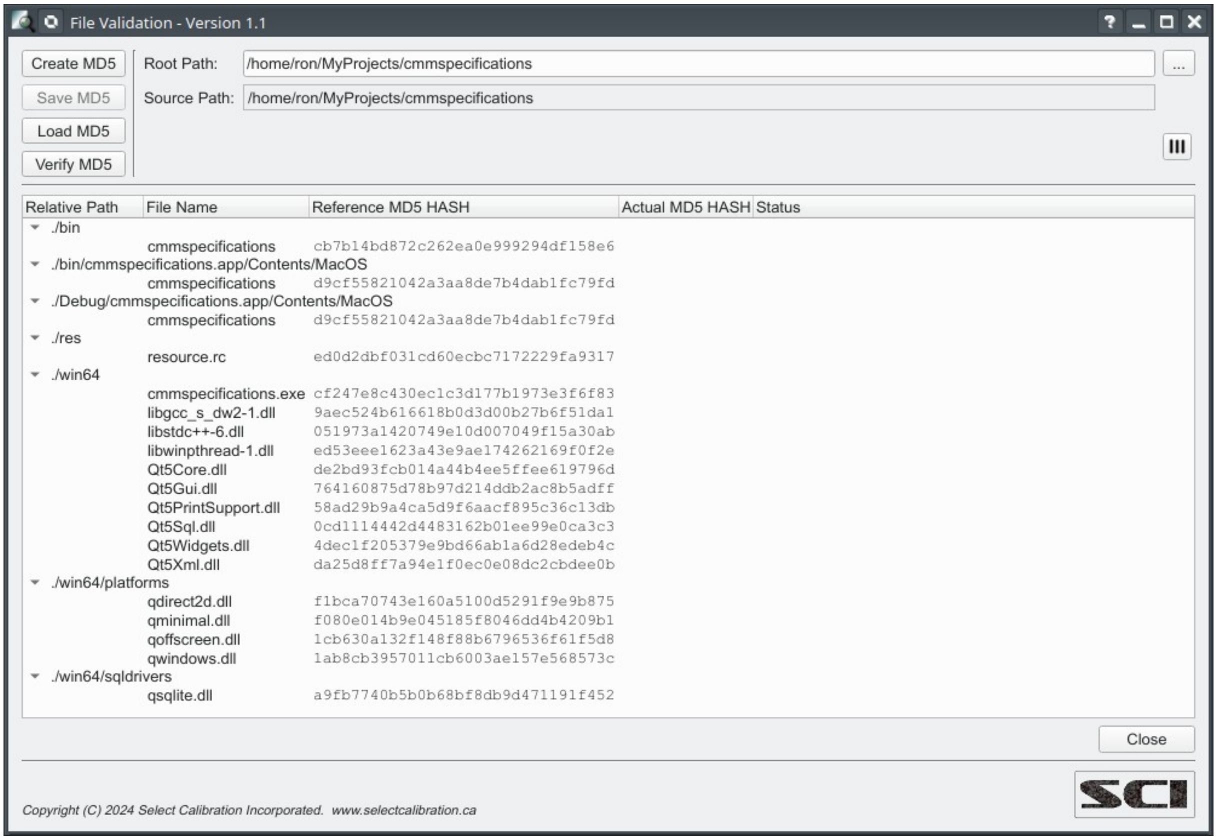
Introduction

The *Validation* utility creates a set of MD5 hash values from one or more files which can be used to determine if the contents or those files change in the future. The reference hash for each file is created from a set of trusted sources and used to compare to the same files at some future point in time.

Changes can be the result of file system corruption or from direct modification by a virus or malware running on an un-trusted computer. File system corruption can happen when files are installed on a USB drive as these are typically formatted with FAT32 and do not have any kind of activity journal. USB drives can be removed without being properly ejected which can lead to file system corruption.

Overview

The *Validation* utility consists of a single window with the majority of the display area used for the file and status display information. Illustration 1 shows the main view of the *Validation* utility.



Root Path: /home/ron/MyProjects/cmmspecifications

Source Path: /home/ron/MyProjects/cmmspecifications

Relative Path	File Name	Reference MD5 HASH	Actual MD5 HASH	Status
./bin				
	cmmspecifications	cb7b14bd872c262ea0e999294df158e6		
./bin/cmmspecifications.app/Contents/MacOS				
	cmmspecifications	d9cf55821042a3aa8de7b4dab1fc79fd		
./Debug/cmmspecifications.app/Contents/MacOS				
	cmmspecifications	d9cf55821042a3aa8de7b4dab1fc79fd		
./res				
	resource.rc	ed0d2dbf031cd60ecbc7172229fa9317		
./win64				
	cmmspecifications.exe	cf247e8c430ec1c3d177b1973e3f6f83		
	libgcc_s_dw2-1.dll	9aec524b616618b0d3d00b27b6f51da1		
	libstdc++-6.dll	051973a1420749e10d007049f15a30ab		
	libwinpthread-1.dll	ed53eee1623a43e9ae174262169f0f2e		
	Qt5Core.dll	de2bd93fcb014a44b4ee5ffee619796d		
	Qt5Gui.dll	764160875d78b97d214ddb2ac8b5adff		
	Qt5PrintSupport.dll	58ad29b9a4ca5d9f6aacf895c36c13db		
	Qt5Sql.dll	0cd1114442d4483162b01ee99e0ca3c3		
	Qt5Widgets.dll	4dec1f205379e9bd66ab1a6d28edeb4c		
	Qt5Xml.dll	da25d8ff7a94e1f0ec0e08dc2cbdee0b		
./win64/platforms				
	qdirect2d.dll	f1bca70743e160a5100d5291f9e9b875		
	qminimal.dll	f080e014b9e045185f8046dd4b4209b1		
	qoffscreen.dll	1cb630a132f148f88b6796536f61f5d8		
	qwindows.dll	1ab8cb3957011cb6003ae157e568573c		
./win64/sqldrivers				
	qsqlite.dll	a9fb7740b5b0b68bf8db9d471191f452		

Copyright (C) 2024 Select Calibration Incorporated. www.selectcalibration.ca

Illustration 1: Validation utility main window.

Table 1: Options:

Option	*Description*
Root Path	Reference location used to either create a new reference MD5 hash file or the location of a set of files to be compared.
Source Path	*Root Path* that was active when the reference MD5 hash file was created.
Create MD5	Create new reference MD5 hash data. The files used for this purpose are determined by the contents of the *Root Path* folder and all sub folders with matching file name filters described in the settings dialog.
Save MD5	Save the active reference MD5 hash data to a file.
Load MD5	Load an existing reference MD5 hash data file.
Verify MD5	Perform a comparison between the loaded MD5 hash data and the files found at the *Root Path* location.
▮▮▮	Settings dialog.
Close	Close the *Validation* utility.

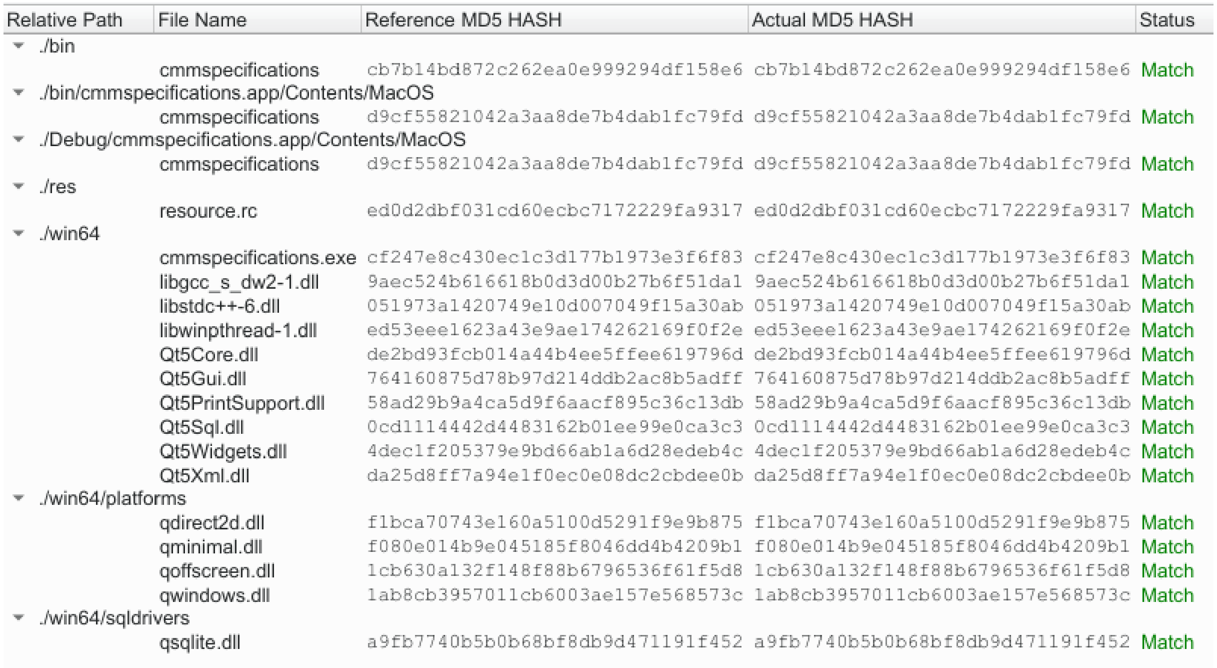
The idea of using a separate root path entry and relative sub directories allows comparison of files contained in a folder that is different from the directory used to create the reference file even if it was generated on a different operating system. For example, the controlled files used by SCI are on two USB drives which will end up with two unique drive letters when opened on the same computer running some version of Windows OS. Comparison of all files can be performed, on either USB drive, provided the user navigates to the proper root folder of each USB drive. The reference data can be generated from a computer running GNU/Linux with this setup.

Verification Output

The output of the *Verification* utility shows the reference and current hash value for each file and the result of the comparison between the two. Illustration 2 shows an example of the output generated from the root folder of the *CMMSpecifications* utility which has binary files for GNU/Linux, MacOS, and Win64.

One interesting observation from illustration 2 is the inclusion of file r*esource.rc* into the list of validation files. Unbeknownst to me this particular file had the execute attribute inproperly set as shown in the file listing below which is why it was added to the list of files to be validated.

```
...
-rw-r--r--  1 ron users    995 Feb 13  2023 import.png
-rw-r--r--  1 ron users    770 Feb 17  2023 limits.png
-rw-r--r--  1 ron users   5495 Jul  6  2020 logo.png
-rw-r--r--  1 ron users    896 Feb 13  2023 new_database.png
-rw-r--r--  1 ron users   1004 Feb 13  2023 open_database.png
-rw-r--r--  1 ron users    957 Feb 13  2023 output.png
-rwxr-xr-x  1 ron users     63 Nov  4  2010 resource.rc
-rw-r--r--  1 ron users  10826 Feb 17  2023 toolbar.xcf
-rw-r--r--  1 ron users  16958 Feb 12  2023 winicon.ico
...
```



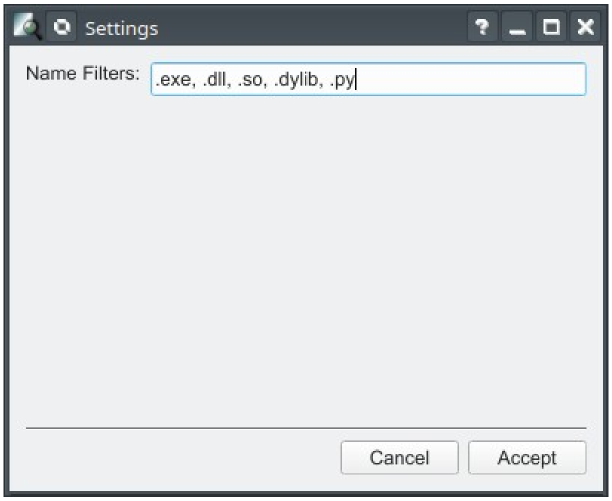
Relative Path	File Name	Reference MD5 HASH	Actual MD5 HASH	Status
./bin				
	cmmspecifications	cb7b14bd872c262ea0e999294df158e6	cb7b14bd872c262ea0e999294df158e6	Match
./bin/cmmspecifications.app/Contents/MacOS				
	cmmspecifications	d9cf55821042a3aa8de7b4dab1fc79fd	d9cf55821042a3aa8de7b4dab1fc79fd	Match
./Debug/cmmspecifications.app/Contents/MacOS				
	cmmspecifications	d9cf55821042a3aa8de7b4dab1fc79fd	d9cf55821042a3aa8de7b4dab1fc79fd	Match
./res				
	resource.rc	ed0d2dbf031cd60ecbc7172229fa9317	ed0d2dbf031cd60ecbc7172229fa9317	Match
./win64				
	cmmspecifications.exe	cf247e8c430ec1c3d177b1973e3f6f83	cf247e8c430ec1c3d177b1973e3f6f83	Match
	libgcc_s_dw2-1.dll	9aec524b616618b0d3d00b27b6f51da1	9aec524b616618b0d3d00b27b6f51da1	Match
	libstdc++-6.dll	051973a1420749e10d007049f15a30ab	051973a1420749e10d007049f15a30ab	Match
	libwinpthread-1.dll	ed53eee1623a43e9ae174262169f0f2e	ed53eee1623a43e9ae174262169f0f2e	Match
	Qt5Core.dll	de2bd93fcb014a44b4ee5ffee619796d	de2bd93fcb014a44b4ee5ffee619796d	Match
	Qt5Gui.dll	764160875d78b97d214ddb2ac8b5adff	764160875d78b97d214ddb2ac8b5adff	Match
	Qt5PrintSupport.dll	58ad29b9a4ca5d9f6aacf895c36c13db	58ad29b9a4ca5d9f6aacf895c36c13db	Match
	Qt5Sql.dll	0cd1114442d4483162b01ee99e0ca3c3	0cd1114442d4483162b01ee99e0ca3c3	Match
	Qt5Widgets.dll	4dec1f205379e9bd66ab1a6d28edeb4c	4dec1f205379e9bd66ab1a6d28edeb4c	Match
	Qt5Xml.dll	da25d8ff7a94e1f0ec0e08dc2cbdee0b	da25d8ff7a94e1f0ec0e08dc2cbdee0b	Match
./win64/platforms				
	qdirect2d.dll	f1bca70743e160a5100d5291f9e9b875	f1bca70743e160a5100d5291f9e9b875	Match
	qminimal.dll	f080e014b9e045185f8046dd4b4209b1	f080e014b9e045185f8046dd4b4209b1	Match
	qoffscreen.dll	1cb630a132f148f88b6796536f61f5d8	1cb630a132f148f88b6796536f61f5d8	Match
	qwindows.dll	1ab8cb3957011cb6003ae157e568573c	1ab8cb3957011cb6003ae157e568573c	Match
./win64/sqldrivers				
	qsqlite.dll	a9fb7740b5b0b68bf8db9d471191f452	a9fb7740b5b0b68bf8db9d471191f452	Match

Illustration 2: Validation data output example.

The set of files and file types can be controlled from the settings dialog but anything marked as executable by the operating system will be automatically included regardless of settings. In the above example executable files and scripts from GNU/Linux, MacOS, and Win64 are shown even though some of those files extensions are not included in the name filter entry from the settings.

Settings

The settings are accessed from the settings button from the main window. Illustration 3 shows an example of the settings dialog.



Settings

Name Filters: .exe, .dll, .so, .dylib, .py

Illustration 3: Settings dialog.

The *Name Filters* option allows various types of files to be included in the validation process by adding suitable file extensions to the *Name Filters* list. The *Validation* utility automatically targets executable files for common operating systems regardless of user preferences.

The current list of settings for the Validation utility are spartan but may be expanded in the future.

Table 2: Settings Options:

Option	*Description*
Name Filters	Comma separated list of file extensions (including the dot) that will identify files to be used when creating the reference hash data.

Creation of Reference Data

The reference data is an XML file containing previously calculated MD5 hashes from one or more files. When performing a verification test the data from this file is required to determine which files are to be tested and if the tested files have been altered. An example of this file is shown below:

```
<?xml version="1.0" encoding="utf-8"?>
<Validation_MD5_Data File_Version="1">
    <Root_Path>/Users/ron/MyProjects/cmmspecifications</Root_Path>
    <Path_Entry>
        <Relative_Path>./bin</Relative_Path>
        <File>
            <Name>cmmspecifications</Name>
            <Hash>cb7b14bd872c262ea0e999294df158e6</Hash>
        </File>
    </Path_Entry>
    <Path_Entry>
        <Relative_Path>./bin/cmmspecifications.app/Contents/MacOS</Relative_Path>
        <File>
            <Name>cmmspecifications</Name>
            <Hash>d9cf55821042a3aa8de7b4dab1fc79fd</Hash>
        </File>
    </Path_Entry>
…
```

Creation Sequence:

1. Select or enter the *Root Path* to the directory containing the files of interest in the *Validation* utility.
2. Click the *Create MD5* button to create the reference file.
3. Click the *Save MD5* button to save the reference data to a file for future comparisons.

The created reference file can be in any location and does not need to be in the Root Path directory. The default path will be the root path folder.

Verification of File Data

The verification function compares one or more existing file hashes to previously created file hashes to make sure the data has not been altered.

Verification Sequence:

1. Click the *Load MD5* button to load a previously created reference file. This file can be in any location on the computer and, for practical reasons, should not be externally accessible.
2. Change the *Root Path* entry of the Validation utility to the location containing the files that are to be tested.
3. Click the *Verify MD5* button to perform the comparison. The comparison result for each file is shown under the status column of the data display in the *Validation* utility.

For verification purposes the *Root Path* can be a changed to any location of interested provided the files and folders contained are in the same structure as what existed when the reference MD5 hashes were created.

The comparison result will be either Match, Different, or Missing depending if the MD5 hashes match, don't match, or the file is not found.

Using a MD5 hash of the file data is more secure than something like a CRC value as it is possible to control the final result of a CRC value if changes are intentional and only a CRC value is considered.

Revision History

Date	*Version*	*Changes*
Dec 2, 2024	1.0	New Program
	1.1	Added a thread to handle calculation of MD5 hashes. Added option to save current reference MD5 data.