

CRCcalc GUI

CDMS Checksum Calculator

CRC • MD5 • SHA-512

User Manual

Calculate and verify file checksums

Version 3.0

Author TC Consulting

Document date August 17, 2026

Contents

Contents.....	2
1. Overview	3
Algorithms.....	3
Operations	3
2. Getting started.....	3
System requirements.....	3
Program files	4
Starting the program	4
3. The main window.....	4
Controls reference	4
4. Calculating checksums	5
Step by step	5
Ways to choose files	5
Reading the results grid	5
Saving the results.....	6
5. Verifying checksums	6
Step by step	6
Accepted input formats	6
How the algorithm is detected	7
6. Working with results.....	7
7. Checksum file format.....	8
8. Command-line tool (optional)	8
Usage	8
Examples.....	8
9. Troubleshooting.....	9
10. Quick reference.....	9

1. Overview

CRCcalc GUI is a Windows desktop tool for generating and verifying file checksums. A checksum (or hash) is a short fingerprint calculated from a file's contents; if even a single byte of the file changes, the fingerprint changes. Checksums are therefore used to confirm that files have been copied, archived, or transmitted without corruption.

The program offers two operations and three algorithms.

Algorithms

Algorithm	Description	Typical use
CDMS CRC	The 16-bit cyclic redundancy check used by the Accuray CDMS system. Values always begin with the tag CBC304027.	Matching existing CDMS checksum records.
MD5	128-bit message digest, shown as 32 hexadecimal characters.	General-purpose integrity checks.
SHA-512	512-bit secure hash, shown as 128 hexadecimal characters. The strongest of the three, and the slowest to compute.	High-assurance verification of important data.

Operations

Operation	What it does
Calculate	Reads one or more files and produces a checksum for each. Results are listed on screen and can be saved to a file.
Verify	Reads a previously saved checksum file, recalculates each file's checksum, and reports whether it still matches (GOOD), has changed (BAD), or is absent (MISSING).

Two tools in one package

The graphical program described in this manual is CRCcalcGUI.exe. A matching command-line tool, crccalc.exe, is included for scripting and automation (see Section 8).

2. Getting started

System requirements

- **Operating system:** Microsoft Windows (64-bit).
- **Installation:** None required. The program is a single self-contained executable that runs in place.
- **Permissions:** Read access to the files being checked; write access to the folder where results are saved.

Program files

File	Purpose
CRCcalcGUI.exe	The graphical application (this manual).
crccalc.exe	Optional command-line version.
README.md	Quick reference and build notes.

Starting the program

Double-click CRCcalcGUI.exe. The main window opens ready for a Calculate operation using the CDMS CRC algorithm. No project or file needs to be opened first.

3. The main window

All work is done from a single window. It is divided into four areas, from top to bottom:

- **Controls panel** – choose the operation, algorithm, input and options, and start or stop a run.
- **Results grid** – one row per file, colour-coded by outcome.
- **Log pane** – a running text account of the operation and its summary.
- **Status bar and buttons** – progress, a one-line status, and actions for the results.

Controls reference

Control	Function
Operation	Selects Calculate or Verify. The other controls enable or disable to suit.
Algorithm	CDMS CRC, MD5 or SHA-512. Used in Calculate mode; in Verify mode the algorithm is detected automatically for each line.
File, path or wildcard	The input box. Its label changes with the mode and options (see below).
Browse file...	Pick a single file with a standard Open dialog.
Browse folder...	Pick a folder; the box is filled in to hash everything inside it.
Recurse sub-directories	When hashing by wildcard, also search every sub-folder, keeping the same file mask.
Input is a list of files	Treat the input as a text file that lists one file path per line.
Also write results to file	Save the calculated checksums to the named file as they are produced.
Start / Stop	Begin the operation, or interrupt a long run. The window stays responsive throughout.
Export grid...	Save the on-screen results as a CSV file.
Open output	Open the saved checksum file (Calculate mode).
Open folder	Open the folder that contains the results.

The input label tells you what is expected

The label above the input box changes to match the current mode: “*File, path or wildcard*” for a normal calculation, “*List file (one path per line)*” when the list option is ticked, and “*Checksum file to verify*” in Verify mode.

4. Calculating checksums

Use Calculate to produce checksums for a set of files.

Step by step

1. Set Operation to Calculate.
2. Choose an Algorithm (CDMS CRC, MD5 or SHA-512).
3. Specify the input – see “Ways to choose files” below.
4. Optionally tick Recurse sub-directories and/or Also write results to file.
5. Press Start. Progress is shown in the status bar and progress bar; for SHA-512 the percentage of the current file is displayed.
6. Review the results in the grid and log. Use Export grid or Open output to keep them.

Ways to choose files

Method	How	Example
Single file	Type the path or use Browse file...	C:\data\plan.cdms
All files in a folder	Use Browse folder..., or type a path ending in a backslash	C:\data\
Wildcard	Type a mask; combine with Recurse to include sub-folders	C:\data*.cdms
List file	Tick Input is a list of files and point to a text file of paths	filelist.txt

Recursion keeps your file mask

When Recurse sub-directories is on, the same mask is applied in every sub-folder. For example *.cdms with recursion finds every .cdms file in the whole tree — and nothing else.

Reading the results grid

Each processed file appears as one row with three columns: the Result (the algorithm name in Calculate mode, or the verdict in Verify mode), the Hash value, and the File path. Rows are colour-coded so problems stand out at a glance.

Result	Meaning
GOOD	The file's current hash matches the stored value.
BAD	The file exists but its hash does NOT match — the file has changed or is corrupt.
MISSING	No file was found on disk for this entry.
INVALID	The hash token was not recognised as CRC, MD5, SHA-512 or a size value.

Saving the results

There are two ways to save, for two different purposes:

Save method	Produces	Best for
Also write results to file	A checksum file in hash *filename format.	Re-verifying the files later (Section 5).
Export grid...	A Result, Hash, File spreadsheet (CSV).	Opening in Excel, reports, or record-keeping.

Either saved file can be verified

Both the checksum file and the exported CSV can be loaded back into Verify mode. You do not have to remember which format you saved.

5. Verifying checksums

Use Verify to check a set of files against previously recorded checksums.

Step by step

1. Set Operation to Verify.
2. In the Checksum file to verify box, choose the saved checksum file or exported CSV (use Browse file...).
3. Press Start.
4. Each entry is checked and shown in the grid as GOOD, BAD, MISSING or INVALID. The log ends with a summary and lists any failures.

Accepted input formats

Verify reads either of the two files the program can save:

- **Native checksum file** – lines of the form `hash *fullpath` (produced by *Also write results to file* and by the command-line tool).
- **Grid CSV export** – the `Result,Hash,File` file from *Export grid*. The header row and quoted paths (including file names that contain commas) are handled automatically.

How the algorithm is detected

Verify does not need to be told which algorithm to use. It inspects each stored hash and chooses accordingly:

Stored value looks like...	Treated as
Begins with CBC304027	CDMS CRC
Exactly 32 hexadecimal characters	MD5
Longer than 32 characters	SHA-512
Begins with 000000	A stored file-size value
None of the above	INVALID

If Verify reports nothing

If every count is zero and the grid stays empty, the file being read is probably not a checksum file at all (for example, an ordinary text or log file). Make sure you are pointing Verify at a file saved by this program — either the `hash *filename checksum` file or the exported CSV.

6. Working with results

Action	Effect
Stop	Interrupts the current run at the next file. Rows already produced remain on screen; the summary notes that the run was stopped.
Export grid...	Writes the visible rows to a CSV file (Result, Hash, File).
Open output	Opens the checksum file written during a Calculate run, if any.
Open folder	Opens the folder containing the results in Windows Explorer.

The window never freezes

Hashing runs on a background thread, so the window keeps responding while a large job is in progress and Stop always takes effect promptly.

7. Checksum file format

A native checksum file is plain text. Each result line contains the hash, a space, an asterisk, and the full file path:

```
CDMS Checksum Calculator v3.0
Generated: 2026-08-17 09:15:02
Algorithm: MD5

900150983CD24FB0D6963F7D28E17F72 *C:\data\abc.txt
9E107D9D372BB6826BD81D3542A419D6 *C:\data\fox.txt
```

The first few lines are an informational header; Verify ignores them automatically, so a saved file can be read straight back in. The exported CSV uses a comma-separated layout instead:

```
Result,Hash,File
MD5,900150983CD24FB0D6963F7D28E17F72,"C:\data\abc.txt"
```

8. Command-line tool (optional)

For scripting and scheduled tasks, the package includes crccalc.exe, which shares the same engine as the graphical program. Run it from a Command Prompt.

Usage

```
crccalc [options] <filename or path>
```

Option	Meaning
-r	Recurse sub-directories.
-@<file>	Hash every file listed in <file>.
-v<file>	Verify against the checksum file <file>.
-o	Also write results to output.txt.
-crc	Use CDMS CRC (the default).
-md5	Use MD5.
-sha512	Use SHA-512.

Examples

```
crccalc *.cdms
crccalc -@filelist -md5
crccalc *.* -sha512 -r > checksums.txt
crccalc -vchecksums.txt
```

Redirect to capture output

The command-line tool prints results to the screen; redirect with `> checksums.txt` to save them. Live progress for SHA-512 is written separately and will not appear in the redirected file.

9. Troubleshooting

Situation	Explanation and remedy
Verify shows 0 good / 0 bad / 0 missing and an empty grid.	The file is not a recognised checksum file. Point Verify at a hash <code>*filename</code> file or an exported CSV saved by this program.
A row shows MISSING.	The recorded file is not at the stored path — it was moved, renamed or deleted, or the checksum file came from another machine with different paths.
A row shows BAD.	The file exists but its contents differ from when the checksum was taken. The file has changed or is corrupt — investigate before relying on it.
A wildcard finds nothing.	Check the mask and folder. To include sub-folders, tick Recurse sub-directories.
SHA-512 is much slower than CRC or MD5.	This is expected; SHA-512 does far more computation. Use it when the extra assurance is worth the time.
A run seems long on many/large files.	Progress is shown continuously and Stop is always available; the window remains usable while it works.

10. Quick reference

Task	Do this
Hash one file	Calculate → pick algorithm → Browse file... → Start
Hash a whole folder tree	Calculate → type <code>folder*</code> → tick Recurse → Start
Save checksums for later	Tick Also write results to file, then Start
Check files against a saved list	Verify → Browse to the checksum file → Start
Get a spreadsheet of results	Run, then Export grid...