

# KeyCheck v3.53 for Win32

## Program Description

KeyCheck is a straightforward, Win32, Btrieve file consistency checker. It was created to provide a simple method of validating key structures within an existing Btrieve/Pervasive.SQL/Zen file. It reads the file, first in physical order, then in order by each key. (With the /T switch, multiple threads will scan all keys at the same time, which will be substantially faster.) If an error is detected, the error is displayed, along with the total number of records read on that key path.

If a different record count is detected on a given key path, an additional warning will also be displayed. This can indicate a corrupted key, but it could just as easily indicate that you were scanning an active file, or one with NULL keys. This is normal and should NOT be construed to indicate an error condition. The same limitation exists for the /C option.

## Platform and Package

Win32 Console; Btrieve Utilities Resources & Tools; GSLic

## Pricing

\$50 Single-User, \$100 Site License; Limited-Function Trial Available

## Command Line Syntax and Help Screen

KeyCheck Version 3.53: 11/23 (C)2022 Goldstar Software Inc.  
Registered to Goldstar Software Inc. (Site License GS)

Usage: KEYCHECK Filename [/Option] [/Option] ...

```
This command scans the Btrieve file for corrupted keys.
*Use /B to identify Bad or missing (Status 2) records on a key path.
*Use /C to Check for unique value mismatches while scanning.
  Use /D to enable Debug mode.
  Use /E to Exit on the first error encountered.
  Use /K# to specify a single Key to traverse (or -1).
  Use /L<filename> to log results to a file (Default=KEYCHECK.LOG).
  Use /O=<owner> to provide a database file owner name.
*Use /P to pre-load each file into cache when starting.
*Use /Q to enable QuickRead (4-Byte Record Reads).
  Use /R to traverse records in Reverse order.
  Use /S to disable Status reporting.
*Use /T for multi-Threaded processing.
*Use /U<filename> to export data to a UNF File.
  Use /W to output statistics in CGI (Web) format.
A * above indicates that it works in Registered Mode ONLY.
```

## Examples and Sample Usage

KEYCHECK is functionally equivalent to the following:

```
BUTIL -RECOVER filename NUL
BUTIL -SAVE filename NUL n 1
BUTIL -SAVE filename NUL n 2
BUTIL -SAVE filename NUL n 3
...
BUTIL -SAVE filename NUL n lastkey
```

KEYCHECK /R is functionally equivalent to the following:

```
BUTIL -RECOVER filename NUL /J
BUTIL -SAVE filename NUL n 1 /J
BUTIL -SAVE filename NUL n 2 /J
BUTIL -SAVE filename NUL n 3 /J
...
BUTIL -SAVE filename NUL n lastkey /J
```

The command line options are defined below in alphabetical order:

### /B: Identify “Bad” Records On Key Path

When data files get corrupted due to a server crash that damages data pages, it is then impossible to extract those records. This option attempts to determine the key value along each key path that corresponds to the missing records. If successful, it reports the key value information to the screen of the missing record. If two records fail in a row, the system may not be able to continue and may stop

processing that key path. Specify the /K switch to scan one key only. (This is not compatible with the multi-threading scanner.)

/C: Check for Unique Values Mismatches in the KAT

The Key Allocation Table count of unique values for each key in the file is shown when you perform a BUTIL -STAT on the file. If the KAT data gets mangled somehow, this can impact SQL queries that depend on this information. This switch adds extra processing that determines the number of unique values that are found down each key path and compares that number with the value in the KAT. If a mismatch is found, a Status 21 will be reported on that key.

/E: Exit after first error is found.

By default, KeyCheck scans all keys looking for errors. If you are scripting this via a batch file, it is often helpful to know what error occurred, if any. However, this error may not be returned from KeyCheck if it is not the LAST error encountered. Use this switch to exit immediately with the appropriate status code when an error is found. Otherwise, KeyCheck will report the HIGHEST error code found.

/K: Specify a key to traverse.

Normally, KeyCheck will process ALL keys in a given file, in order from -1 (physical order) to the last key available. Use the /K switch if you wish to scan only ONE specific key, and provide the key number immediately after the "K" on the command line, as in "/K3". (Using a single key ignores the /T switch.) Note that you can scan the System Data keys 124 and 125, also.

/L: Log to a log file.

KeyCheck normally reports to the standard output device (usually the screen). However, you can add the /L option to append all messages to a logfile instead. The default log file is KEYCHECK.LOG, or you can specify your own filename after the /L option.

/O: Provide an Owner Name.

Some database files are protected via an owner name and cannot be opened without this name. If you encounter a file which is protected in this manner, you can use the "/O" switch to provide the owner name. For example, if the owner name is "FRED", the proper parameter is "/O=FRED". Please note that the owner name, like most passwords, is case sensitive.

/P: Pre-cache the data file before opening.

With PSQL v13 and above, the MKDE can be asked to pre-cache the database file when opening it. If you are scanning a larger file, but one that still fits in the L1 cache, then this option can greatly increase throughput by more efficiently loading the pages from the file into RAM before scanning the file. If your file is larger than the cache, or if you are doing this on a live system with many other processes running at the same time, then using this option can have an adverse impact on other users on the server at the same time.

/Q: Enable QuickRead scanning.

By default, KeyCheck will read EVERY byte of every record (up to 63K). While this is the most effective way to check an entire file, it is possible to increase performance (at the expense of less stringent checking) by adding the "/Q" switch. When this is enabled, KeyCheck will only read 4 bytes from each record instead of

the whole record. The performance difference, especially across a networked connection, can be dramatic.

/R: Scan in reverse order.

By default, Keycheck will scan all indices in forward order. It is possible that a key structure could have become corrupted on a reverse pointer only. Use the "/R" option (after a successful forward run) to scan backwards as well.

/S: Disable status reporting.

KeyCheck reports periodically on its progress. In single-threaded mode, a "twirly" spins after every 100 records to show that it is working, along with a percentage complete. It will update this number every 10% of the way through small files (under 100,000 records) or every 1% for large files (over 100,000 records). To improve performance this status reporting can be disabled with the "/S" flag. (Automatically done with CGI output.)

For the multi-threaded scanner, a progress update for each thread is displayed to the screen every one second by default, and the scanner will issue a Windows event sound for each key completed (either OK or ICONHAND). Using the /S switch will disable this progress report and stop the sounds (which will run faster for smaller files). When the scan is running, you can also press "Q" during the scan to "Quiet" the display, and "Q" again to enable it. To change the screen update rate, press 0 (1/4 second), 1 (1 second), 2 (2.5 seconds), or 3 (5 seconds).

/T: Enable the multi-threading scanner.

KeyCheck v3.2 and above contains a multi-threaded scanning engine that can scan all keys simultaneously. This can provide a substantial gain in scanning throughput, especially when run directly on the server itself. If you detect a problem (like an infinite key loop), you can press <Esc> to terminate cleanly before all threads finish.

/U<filename>: Export records read to a UNF file.

This feature is similar to a BUTIL -SAVE function in that it can write the records scanned to a UNF file as it is scanning. When used with the /K and /B options, it can be useful for exporting data AND determining which records are missing.

/W: Enable Web (CGI) output.

Normally, KeyCheck will dump its scanning data to the screen (stdout). However, it may be advantageous to be able to run KeyCheck on a web server to check files from a remote location. To use this, simply create a link like this:

```
<A REF="http://www.domain.com/scripts/keycheck.exe?c:\data\file.mkd%20/w"
target="_blank">FILE.MKD</A>
```

This link will call the KEYCHECK.EXE file (in the web server's SCRIPTS directory) and pass it the name of the data file, as well as the "/W" switch.

KeyCheck returns one of the following ErrorLevel values:

- 4 Out of Memory
- 3 Invalid Specified Key Number Value
- 2 Unable to Parse Owner Name from Command Line
- 1 No Parameters Specified (Help Screen Printed)
- ### Btrieve Status Code Returned

Batch files can check ERRORLEVEL and handle issues automatically. Status 0 will indicate a proper EndOfFile condition, while Status codes 2, 30, and 54

indicate file corruption. Status 21 can be returned if the /C option is selected and the unique key counts don't match. Status 19 can be returned if the system reads more than three times the indicated record count, which can happen when an index loop is encountered.

You may need to use the /E flag to ensure that a valid error is returned, since some errors may exist on some keys, but not on all.

## Other Information

KeyCheck for Win32 is provided as a registered program only. Goldstar Software can provide customized changes or other options to this program to registered users upon request. For more information, contact Goldstar Software Inc. at [www.goldstarsoftware.com](http://www.goldstarsoftware.com)

## Version History

Version 2.2: First Documented Version

Version 2.3: Added /E switch and improved error detection and reporting.

Version 2.4: Fixed a bug in the handling of owner-protected files.

Version 2.5: Added ability to report on missing records from data pages with the /B switch.

Version 2.6: Added ability to detect Unique Value Count Mismatches.

Version 2.7: Improved error reporting by returning the highest status code.

Version 2.8: Added ability to export data to a UNF file (like BUTIL).

Version 2.9: Improved status reporting for scanning from a batch file.

Version 2.91: Added ability to manually scan Key 125 (System Data).

Version 3.0: Added trial license capability.

Version 3.1: Added log file (/L) capability. Also modified the display routines so that basic progress information (key numbers and percent complete) is ALWAYS sent to stderr (i.e. the screen), even if logging is enabled or stdout is redirected.

Version 3.2: Added the multi-threading scanning engine (/T) option.

Version 3.21: Added handling of Status 54 in the /B option.

Version 3.30: Improved error handling on the /B switch to continue when certain errors occur; display number of "bad" records located.

Version 3.31: Improved performance and display of key value on /B switch.

Version 3.32: Fixed issue with returning highest error to the console in all cases. Added capability of using the /C switch with the multi-threaded scanner.

Version 3.33: Improved multi-threaded scanner to add support for the /S and /E switches.

Version 3.34: Improved consistency of logging. Disabled beeping at the end of the multi-threaded scanner when the /Silent switch is used.

Version 3.35: Fixed bug in which files without Key 0 failed.

Version 3.36: Updated licensing code.

Version 3.37: Improved display of bad key values with the /B switch.

Version 3.38: Added detection and display of ending record count, to make it easier to see when an active file is being scanned.

Version 3.39: Added 32-byte owner name support.

Version 3.40: Added ability to display missing record data on the System Data Key (/K125 /B).

Version 3.50: Added /P option to pre-cache data files prior to opening them. Also altered the output to include the term WARNING on any imperfect result, making it easier to scan the log file for problematic files.

Version 3.51: Converted code to use W3BTRV7; added support of files with more than 4B records; reduced twirly frequency from 100 records to 2000 records (to increase performance).

Version 3.52: Added support for scanning internal system data keys 124/125.

Version 3.53: Added loop detection to terminate the scan after 300% the file's record count.

### **Known Problems**

Extracting data to a UNF format (/U) only supports a maximum record size of 63K. This is more of a convenience and debugging feature and not intended to be used for file recovery.