

# GSRecover v2.11 for Win32

## Program Description

GSRecover is a Goldstar Software tool designed to display the File Control Record (FCR) of a file, provide an analysis of the Page Allocation Table (PAT) in a file, and extract records from a Btrieve file without using Btrieve. The net result is that we can often extract MORE data than a pure database-level recovery, leading to a higher success rate for the recovery efforts.

One of the enhanced capabilities of GSRecover is to build and work with a Dynamic PAT (DynPat). This is similar to the real Page Allocation Table in the file, but is built dynamically at runtime by searching the data file for each logical page, maintaining a list of the most recent version. This allows us to find data pages that might not be in the current PAT and to recover data even when a PAT has sustained partial or complete damage (as with the old PATFIX problem).

**WARNING:** All of the DynPat functions suffer from a critical problem. The PSQL database will write pages being changed in a transaction. If a transaction is aborted, it is possible to have NEWER pages show up in the DynPat than exist in the real file. This is an unfortunately side-effect with no known cure. If the application in use uses transactions and could possibly have aborted a transaction, using these features will FORCE the transaction to commit, even if it should not have committed! Use these features at your own risk, or for extreme data recovery purposes ONLY.

## Platform and Package

Win32 Console; Btrieve Advanced Recovery ToolKit; GSLic

## Pricing

Available as part of the BART ToolKit only.

## Command Line Syntax and Help Screen

GSRecover Version 2.11: 05/19 (C)2025 Goldstar Software Inc.  
Using Low-Level File I/O Version 2.42.15

Usage: GSRECOVER SrcFile [Options]

```
This utility extracts information from Btrieve data files without Btrieve.
/BE[#]: Blank the Extent Counter (to #, or 1 if missing).
/BL: Blank out the LastInsertTimestamp.
/BR: Blank out the RI VarPage Information.
/BS: Blank out the Segment Statistics Flag (if set).
/DD: Delta File: Display Pages from Delta File.
/DM: Delta File: Merge Delta Pages into Data File.
/E: Include Error Info (D) on bad record exports (for manual review).
/F[0|1|A|I][K]: Display FCR details from Active (or 0/1/Inactive) FCR copy.
    Add K to also print the Key Allocation Table with the FCR details.
/KS#: Display Key Statistics for the given key number (*=All).
/KV[A|H]#: Display Key Values in ASCII/HEX for the given key number.
/M[0|1]: Display MiscData Page (Default=0) or UserMiscData (1 -> v9 Only).
/PC: Compare Dynamic PAT to Real PAT & Report Differences.
/PD: Use DYNAMIC PAT for Reading the Data File.
/PR: Use REAL PAT for Reading the Data File (Default).
/PP: Print Dynamic PAT to StdOut (for testing purposes).
/PU: Update Real PAT with Dynamic PAT Data (***).
/R<filename>: Recover Data to <Filename>.
/S: Include SysData on Recovered (UNF) data file (Requires DXUTIL).
/T: Execute Test Module.
/U: Open files in UNPROTECTED mode. (Allows concurrent access.)
*** Note: Use of this feature without a proper BACKUP is prohibited.
```

## Examples and Sample Usage

GSRecover offers some key functionality in the recovery of Btrieve database files. It is broken down into several functions, and some functions may be eliminated from future versions as they are moved to other tools.

**GSRECOVER <filename> /BE[#]**

If an extended file has been partially deleted (i.e. one or more extents have been deleted by mistake), even simple operations like BUTIL –CLONE may return a Status 13. This option forces the extent counter in the FCR to any value. If the optional number is 0 or is omitted, the extent counter is changed to 1.

### **GSRECOVER <filename> /BL**

If you find that a file has a synthetic time stored for the LastInsertTimestamp value, then this option will blank that value (set it to 0). This can help you to eliminate synthetic time from a system with files NOT using system data, and should be far faster than a full Rebuild operation.

### **GSRECOVER <filename> /BR**

In some cases, users copy around database files that have been bound or that include relational integrity information, making the file impossible to access from another server. This option blanks out the MiscData field in the data file so that the file can then be accessed.

### **GSRECOVER <filename> /BS**

If a user creates (or rebuilds) a v13 file on Zen v15.10 or above, the engine will maintain additional statistics on each key segment to improve SQL performance. However, the resulting file will no longer be readable on PSQL v13.30, Zen v14, or Zen v15.00. This option blanks out the Segment Statistics Flag in the data file so that the file can then be accessed in these older engines. As this does NOT fix the segment statistics themselves, if there are any records in the file, you will be directed to rebuild the file immediately prior to any additional disk writes to prevent problems with that file.

### **GSRECOVER <filename> /DD**

If a data file is stuck in continuous operations mode and the engine crashes, a Delta file may be left behind that may or may not be damaged. The /DD option will walk the tree structure inside the delta file and display each page located therein, along with the first 8 bytes of the page (i.e. the page header) in hex format. This can be used to determine if a specific page is stored within the delta file.

### **GSRECOVER <filename> /DM**

In some cases, a file can get stuck in Continuous Operations mode for a long period of time, generating a very large delta file as a result. While it is possible to force the engine to roll in the delta file while users are in the system, the engine only rolls in 100 pages for each system transaction (10 seconds, by default). This can cause the roll-in process to take quite some time. On a heavily-used file, the users can generate more new writes than the engine can merge within a period of time, and the engine will only get further and further behind. In this case, changing the system transaction interval from 10000 to 100 is a good place to start. However, if you can take the system off-line, GSRecover's /DM switch can merge the pages back into the file much more quickly.

### **GSRECOVER <filename> /F**

The first function of GSRecover is to display the contents of the File Control

Record. The FCR is the first pair of pages in the data file, and it contains information largely visible from the BUTIL -STAT report. However, there are numerous other tidbits buried in the FCR that may come in handy for file recovery. The above form of the command will display the currently active FCR page to the screen, with all details splayed out. You can append a 0 or 1 to the FCR to see the specific FCR copy, I to see the Inactive FCR, or A to see the Active FCR (which is the default). The /K option can also be added to the /F option in order to display the key allocation table from the file, which contains information about each individual key and segment in the file.

### **GSRECOVER <filename> /KS#**

This /KS# option causes GSRecover to calculate and display key statistics for the specified key. Valid key values are 0 through the highest defined key, or 125 for the System Data Key or 124 for the UpdateSysData key. Providing an asterisk, as in /KS\*, will display statistics for all normal keys. An example of the output is shown below:

```
Opening file person.mkd...

Key Number Selected      :          0
Total Key Pages Scanned:          7
Maximum Key Page Depth  :          2
Total Key Entries Found:        1500
Total Key Page Bytes     :        28672
Total Overhead Bytes     :          112
Total Key Space Used     :        24000
Total Key Space Wasted   :         4560
Percentage Space Wasted:        15.90%

# Keys at BTree Level 1:          5
# Keys at BTree Level 2:        1495
```

These statistics can be used to determine the overhead created by in a given key in terms of total file space, as well as the amount of wasted space from partially-filled key pages, which may indicate a benefit from rebuilding or index balancing. Here is an example of one key in a file before and after rebuild:

#### **BEFORE:**

```
Key Number Selected      :          0
Total Key Pages Scanned:        4380
Maximum Key Page Depth  :          3
Total Key Entries Found:       294902
Total Key Page Bytes     :       6727680
Total Overhead Bytes     :        52560
Total Key Space Used     :       3538824
Total Key Space Wasted   :       3136296
Percentage Space Wasted:        46.62%

# Keys at BTree Level 1:          62
# Keys at BTree Level 2:        4253
# Keys at BTree Level 3:       290587
```

#### **AFTER:**

```
Key Number Selected      :          0
Total Key Pages Scanned:        2323
Maximum Key Page Depth  :          3
Total Key Entries Found:       294902
Total Key Page Bytes     :       3568128
Total Overhead Bytes     :        27876
Total Key Space Used     :       3538824
Total Key Space Wasted   :         1428
Percentage Space Wasted:         0.04%

# Keys at BTree Level 1:          17
# Keys at BTree Level 2:       2286
```

```
# Keys at BTree Level 3:      292599
```

As you can see, the rebuilt file shows a MUCH smaller key structure and almost no wasted space due to partially-filled index pages. As a result, the rebuild made the file much more efficient.

The final statistics provided in the list indicate the number of key values at each B-Tree, which can also be an interesting statistic when comparing different page files.

For more detailed statistics, use the KeyData tool instead.

### **GSRECOVER <filename> /KV[A|H]#**

This /KV[A|H]# option causes GSRecover to run through the index data structures in the file for the specified key and display the key value for each key page entry. Valid key values are 0 through the highest defined key, or 125 for the System Data Key and 124 for UpdateSysData. You can include the optional A or H to display the key data in ASCII or in hexadecimal, as dictated by the key data types. If not specified, Hex is the default. Preceding each key value is the record pointer value (for unique or repeating-duplicate keys) or a Head and Tail pointer for linked-duplicate keys, like this.

```
Only RecPtr 00000130: .Civarelli.....Vincent.....
Only RecPtr 000009B1: .Civello.....Joseph.....
Head 00000144, Tail 000009D5: .Clinton.....Bill.....
Only RecPtr 00000627: .Coachman.....Crystal.....
Only RecPtr 000001B8: .Cob.....Grady.....
```

This information can be used to determine which key values may be missing from an exported data set on a corrupted file, which can be helpful in locating the extent of data corruption, or finding out which data has been lost.

### **GSRECOVER <filename> /M**

This /M0 option causes GSRecover to display the Miscellaneous Data Fragment from the file and exit. MiscData is used when files are bound to a database and for other purposes. Starting with PSQLv9.5, Pervasive has added a UserMiscData block as well, which can be displayed with the /M1 switch. (Using /M1 on any older file formats will return the file MiscData. An example of the output is shown below:

Miscellaneous Data Fragment Found:

```
00000000: FF FF FF FF FF FF BC 02 00 00 44 45 4D 4F 44 41      .....<...DEMODA
00000010: 54 41 20 20 20 20 20 20 20 20 20 20 20 6D 79      TA              my
00000020: 74 65 73 74 20 20 20 20 20 20 20 20 20 20 20      test
00000030: 20 20 00 00 00 00 08 00 00 00                      .....
```

If the RI data shown here is causing a problem (i.e. Status 73, 172, etc.) and must be removed, then it can be forcibly cleared via the /BR switch. However, you MUST rebuild the file after clearing this data, as this option only clears the pointer to the MiscData page – it does not actually free up the page.

### **GSRECOVER <filename> /PC**

This operation builds the DynPat and compares the resulting DynPat structure to the actual PAT structure of the file. Any discrepancies between the two will be displayed as shown here:

```
LogPage 00000004: DPAT=00000005(0044) U=8010, PAT=000000af(0044) U=800d
```

The output contains the Logical Page Number, then the data from the Dynamic

PAT, including the Physical Page Number, Page Type Code, and the Usage Count for this page. Finally, the data from the current PAT is shown. Note in this example, Logical Page 4 exists in the “real” PAT at Physical Page 0xAF with a usage count of 0x800D, but the DynPat algorithm found a newer copy with a usage count of 0x8010 at Physical Page 0x05. This could indicate that the real PAT is damaged. In this sample file, we simulated this condition by aborting a transaction with several changes. (See WARNING above.)

### **GSRECOVER <filename> /PD**

The /PD switch instructs GSRecover to leverage the DynPat for the data recovery efforts. This switch may be used with the /R option to ignore the existing PAT and attempt a recovery of the data by simply searching for data pages and using the newest one it can find.

### **GSRECOVER <filename> /PR**

The /PR function instructs GSRecover to use the real PAT when reading the file. This switch is the default, but it may be needed for future options.

### **GSRECOVER <filename> /PP**

This test function prints the DynPat to the stdout stream. It can be used for diagnostic purposes, and it may be removed in the future.

### **GSRECOVER <filename> /PU**

This operation builds the DynPat and compares the resulting DynPat structure to the actual PAT structure of the file, as in /PC. If any discrepancies are found, they are automatically corrected to the value detected by the DynPat. This can be used to recover from a damaged PAT situation.

**WARNING:** This function is DESTRUCTIVE and writes back data to the existing file! Performing this operation without a backup is NOT recommended. See the notes regarding the accuracy of the DynPat before attempting to use this operation.

### **GSRECOVER <filename> /T**

This operation performs a test function that may change with each version. It is for developer use only.

### **GSRECOVER <filename> /R<filename>**

This operation performs a BUTIL –RECOVER function on the database by walking each data page and exporting records in the UNF format. Records will be written to the second filename specified. Since this tool does not actually use the Btrieve interface to walk the file, it may be able to recover additional data that the MKDE cannot recover, and it may skip over some areas that the MKDE chokes on.

If a database uses variable records, then this tool will recover the fixed-length portion of records where the variable portion is no longer available. When this happens, a message will be displayed with the record location (during the run), and a total number of truncated records will be reported at the end of the run.

Adding the /S option will ALSO export the System Data for each record in the

file, placed at the beginning of each record in the UNF file, starting with an “S” indicator. The only utility currently capable of reloading the data from this format is DXUTIL, by Goldstar Software. We are lobbying Actian to add this functionality into BUTIL, but no changes have been made yet.

If you add the /E option, then any records less than the Fixed Record Length will also be exported for reference purposes. Also, records with compression buffer errors will include a display of a “D,” before the exported data. These “bad” record exports can be used to help locate the invalid data in the file for a manual repair attempt.

### **Return Codes**

GSRecover will return one of the following return codes:

- 1: Licensing error.
- 2: Incompatible options presented.
- 3: Unable to open database file.
- 4: Insufficient memory.
- 0: Normal completion for non-recovery functions.
- n: A positive number is returned to indicate the number of records exported from the /R function. This can allow the use of batch files or calling applications to intelligently decide if it makes sense to continue with the recovery process. (To extract the number of records in a Btrieve file from a batch file, use BtrvInfo with the /RC switch.)

### **Other Information**

GSRecover is part of the BART Toolkit.

For more information on these utilities contact us at [www.goldstarsoftware.com](http://www.goldstarsoftware.com)

### **Version History**

Version 1.0: First released version.

Version 1.1: Major Release: Updated LowLevel IO for 9.5 files.

Version 1.2: Added ability to display Miscellaneous Data Fragment.

Version 1.3: Fixed MiscData bug and added ability to blank the MiscData pointer.

Version 1.4: Added GSLic capability.

Version 1.5: Removed Key information from /F option, Added /K option to display the key tables.

Version 1.51: Added display of calculated TimeStamp for LastInsert in the FCR.

Version 1.60: First build that supports the use of VATs on compressed records.

Version 1.61: Added display support for long owner names from v10.10 engine.

Version 1.7: Updated return value to return the number of records recovered.

Version 1.8: Added ability to use the /R function with the Dynamic PAT.

Version 1.81: Fixed Last Insert timestamp display issue; Properly detect page compression and terminate immediately; Show physical page size for 9.5C files.

Version 1.90: Added Key Statistics and Key Values Options.

Version 1.91: Fixed error with GSRecover failing to find records that had been updated exactly 255 times in their lifetime.

Version 1.93: Added the /BE option, and added the display of the truncated record counter at the end of a Recover run.

Version 1.94: Added ability to display FCR data on v9.5 file with page compression, since this data is valid.

Version 1.95: Display error and exit if attempting to manipulate an Encrypted file.

Version 1.96: Updated licensing code.

Version 1.97: Added /DD and /DM switches to handle delta files.

Version 1.98: Increased the maximum record size from 1MB to 100MB.

Version 2.00: Added support for the v13 file format.

Version 2.01: Added support for AutoTimeStamp fields.

Version 2.02: Added support for Timestamp2 fields and parse file sub-version.

Version 2.03: Used profiling to improve performance considerably!

Version 2.04: Added support for the UpdateSysKey in the Export function.  
Added corrected display of new LastInsertTimestamp for new file format.

Version 2.05: Added display of v15.10 SegmentStats in the Sub-Version of the FCR and in the KAT data display, as well as the ability to clear the Segment Statistics Flag.

Version 2.06: Updated display logic for LastInsertedTimestamp to properly display timestamp value for each file format; Added ability to display Key Data for system keys 125 and 124.

Version 2.07: Added the /BL option to blank the LastInsertTimestamp field.

Version 2.08: Distinguish between long owner names created with v13 and older versus long owner names created with v14/15.

Version 2.09: Fixed issue with blanking the extent counter on 9.5 files.

Version 2.10: Improved support for v13/v16 file formats.

Version 2.11: Fixed a few issues in the FCR Display for v13/v16 formats.

## **Known Problems**

Support for files using VATs is incomplete. Need to test on VATs without compression, need to test on v9.5 files (with and without compression), and need to test on VATs with very large records.

GSRecover will not work correctly on files with page compression or encryption.

The /DD switch currently has minimal error handling and may not work properly if the delta file is damaged. If you have a problem with this, please submit your sample data to Goldstar Software for review.

The /DM switch currently is only tested with single-extent files, or with files where all extents are fully populated. If a 7.x or 8.x file is just short of a 2GB extent and the page immediately preceeding the extent boundary is NOT in the delta file for some reason, then that extent may be left too short. Again, if you have a problem with this, please submit your sample data to Goldstar Software for review.

Files with variable records over 100MB in size will be truncated. If you have such a file, please contact Goldstar Software for assistance.