Rocky Enterprise Linux 9.2 Manual Pages on command 'myisampack.1'

$ man myisampack.1

MYISAMPACK(1) MySQL Database System MYISAMPACK(1)

NAME

myisampack - generate compressed, read-only MyISAM tables

SYNOPSIS

myisampack [options] file_name ...

DESCRIPTION

The myisampack utility compresses MyISAM tables. myisampack works by compressing each column in the table separately. Usually, myisampack packs the data file 40% to 70%.

When the table is used later, the server reads into memory the information needed to decompress columns. This results in much better performance when accessing individual rows, because you only have to uncompress exactly one row.

MySQL uses mmap() when possible to perform memory mapping on compressed tables. If mmap() does not work, MySQL falls back to normal read/write file operations.

Please note the following:

? If the mysqld server was invoked with external locking disabled, it

is not a good idea to invoke myisampack if the table might be updated by the server during the packing process. It is safest to compress tables with the server stopped.

? After packing a table, it becomes read only. This is generally intended (such as when accessing packed tables on a CD).

? myisampack does not support partitioned tables.

Invoke myisampack like this:

myisampack [options] file_name ...

Each file name argument should be the name of an index (.MYI) file. If you are not in the database directory, you should specify the path name to the file. It is permissible to omit the .MYI extension.

After you compress a table with myisampack, use myisamchk -rq to rebuild its indexes. myisamchk(1).

myisampack supports the following options. It also reads option files and supports the options for processing them described at Section 4.2.2.3, ?Command-Line Options that Affect Option-File Handling?.

? --help, -? Display a help message and exit.

? --backup, -b Make a backup of each table's data file using the name tbl_name.OLD.

? --character-sets-dir=dir_name The directory where character sets are installed. See Section 10.15, ?Character Set Configuration?.

? --debug[=debug_options], -# [debug_options] Write a debugging log. A typical debug_options string is d:t:o,file_name. The default is d:t:o.

This option is available only if MySQL was built using WITH_DEBUG. MySQL release binaries provided by Oracle are not built using this option.

? --force, -f Produce a packed table even if it becomes larger than the original or if the intermediate file from an earlier invocation of myisampack exists. (myisampack creates an intermediate file named tbl_name.TMD in the database directory while it compresses the table. If you kill myisampack, the .TMD file might not be

deleted.) Normally, myisampack exits with an error if it finds that tbl_name.TMD exists. With --force, myisampack packs the table anyway.

? --join=big_tbl_name, -j big_tbl_name Join all tables named on the command line into a single packed table big_tbl_name. All tables that are to be combined must have identical structure (same column names and types, same indexes, and so forth).
big_tbl_name must not exist prior to the join operation. All source tables named on the command line to be merged into big_tbl_name must exist. The source tables are read for the join operation but not modified.

? --silent, -s Silent mode. Write output only when errors occur.

? --test, -t Do not actually pack the table, just test packing it.

? --tmpdir=dir_name, -T dir_name Use the named directory as the location where myisampack creates temporary files.

? --verbose, -v Verbose mode. Write information about the progress of the packing operation and its result.

? --version, -V Display version information and exit.

? --wait, -w Wait and retry if the table is in use. If the mysqld server was invoked with external locking disabled, it is not a good idea to invoke myisampack if the table might be updated by the server during the packing process.

The following sequence of commands illustrates a typical table compression session:

```
$> ls -l station.*
-rw-rw-r--   1 jones    my         994128 Apr 17 19:00 station.MYD
-rw-rw-r--   1 jones    my          53248 Apr 17 19:00 station.MYI
$> myisamchk -dvv station
MyISAM file:     station
Isam-version:  2
Creation time: 1996-03-13 10:08:58
Recover time:  1997-02-02  3:06:43
Data records:             1192  Deleted blocks:             0
```

Datafile parts: 1192 Deleted data: 0

Datafile pointer (bytes): 2 Keyfile pointer (bytes): 2

Max datafile length: 54657023 Max keyfile length: 33554431

Recordlength: 834

Record format: Fixed length

table description:

Key Start Len Index Type Root Blocksize Rec/key

1 2 4 unique unsigned long 1024 1024 1

2 32 30 multip. text 10240 1024 1

Field Start Length Type

1 1 1

2 2 4

3 6 4

4 10 1

5 11 20

6 31 1

7 32 30

8 62 35

9 97 35

10 132 35

11 167 4

12 171 16

13 187 35

14 222 4

15 226 16

16 242 20

17 262 20

18 282 20

19 302 30

20 332 4

21 336 4

22 340 1

23 341 8

24	349	8
25	357	8
26	365	2
27	367	2
28	369	4
29	373	4
30	377	1
31	378	2
32	380	8
33	388	4
34	392	4
35	396	4
36	400	4
37	404	1
38	405	4
39	409	4
40	413	4
41	417	4
42	421	4
43	425	4
44	429	20
45	449	30
46	479	1
47	480	1
48	481	79
49	560	79
50	639	79
51	718	79
52	797	8
53	805	1
54	806	1
55	807	20
56	827	4

```
57    831   4

$> myisampack station.MYI

Compressing station.MYI: (1192 records)

- Calculating statistics

normal:     20  empty-space:   16  empty-zero:     12  empty-fill:  11

pre-space:   0  end-space:     12  table-lookups:   5  zero:         7

Original trees:  57  After join: 17

- Compressing file

87.14%

Remember to run myisamchk -rq on compressed tables

$> myisamchk -rq station

- check record delete-chain

- recovering (with sort) MyISAM-table 'station'

Data records: 1192

- Fixing index 1

- Fixing index 2

$> mysqladmin -uroot flush-tables

$> ls -l station.*

-rw-rw-r--   1 jones    my         127874 Apr 17 19:00 station.MYD

-rw-rw-r--   1 jones    my          55296 Apr 17 19:04 station.MYI

$> myisamchk -dvv station

MyISAM file:     station

Isam-version:  2

Creation time: 1996-03-13 10:08:58

Recover time:  1997-04-17 19:04:26

Data records:               1192  Deleted blocks:              0

Datafile parts:             1192  Deleted data:                0

Datafile pointer (bytes):      3  Keyfile pointer (bytes):     1

Max datafile length:    16777215  Max keyfile length:     131071

Recordlength:                834

Record format: Compressed

table description:

Key Start Len Index   Type                 Root  Blocksize    Rec/key
```

1 2 4 unique unsigned long 10240 1024 1

2 32 30 multip. text 54272 1024 1

Field Start Length Type Huff tree Bits

1 1 1 constant 1 0

2 2 4 zerofill(1) 2 9

3 6 4 no zeros, zerofill(1) 2 9

4 10 1 3 9

5 11 20 table-lookup 4 0

6 31 1 3 9

7 32 30 no endspace, not_always 5 9

8 62 35 no endspace, not_always, no empty 6 9

9 97 35 no empty 7 9

10 132 35 no endspace, not_always, no empty 6 9

11 167 4 zerofill(1) 2 9

12 171 16 no endspace, not_always, no empty 5 9

13 187 35 no endspace, not_always, no empty 6 9

14 222 4 zerofill(1) 2 9

15 226 16 no endspace, not_always, no empty 5 9

16 242 20 no endspace, not_always 8 9

17 262 20 no endspace, no empty 8 9

18 282 20 no endspace, no empty 5 9

19 302 30 no endspace, no empty 6 9

20 332 4 always zero 2 9

21 336 4 always zero 2 9

22 340 1 3 9

23 341 8 table-lookup 9 0

24 349 8 table-lookup 10 0

25 357 8 always zero 2 9

26 365 2 2 9

27 367 2 no zeros, zerofill(1) 2 9

28 369 4 no zeros, zerofill(1) 2 9

29 373 4 table-lookup 11 0

30 377 1 3 9

31 378 2 no zeros, zerofill(1) 2 9

32 380 8 no zeros 2 9

33 388 4 always zero 2 9

34 392 4 table-lookup 12 0

35 396 4 no zeros, zerofill(1) 13 9

36 400 4 no zeros, zerofill(1) 2 9

37 404 1 2 9

38 405 4 no zeros 2 9

39 409 4 always zero 2 9

40 413 4 no zeros 2 9

41 417 4 always zero 2 9

42 421 4 no zeros 2 9

43 425 4 always zero 2 9

44 429 20 no empty 3 9

45 449 30 no empty 3 9

46 479 1 14 4

47 480 1 14 4

48 481 79 no endspace, no empty 15 9

49 560 79 no empty 2 9

50 639 79 no empty 2 9

51 718 79 no endspace 16 9

52 797 8 no empty 2 9

53 805 1 17 1

54 806 1 3 9

55 807 20 no empty 3 9

56 827 4 no zeros, zerofill(2) 2 9

57 831 4 no zeros, zerofill(1) 2 9

myisampack displays the following kinds of information:

? normal

The number of columns for which no extra packing is used.

? empty-space

The number of columns containing values that are only spaces. These occupy one bit.

? empty-zero

The number of columns containing values that are only binary zeros. These occupy one bit.

? empty-fill

The number of integer columns that do not occupy the full byte range of their type. These are changed to a smaller type. For example, a BIGINT column (eight bytes) can be stored as a TINYINT column (one byte) if all its values are in the range from -128 to 127.

? pre-space

The number of decimal columns that are stored with leading spaces. In this case, each value contains a count for the number of leading spaces.

? end-space

The number of columns that have a lot of trailing spaces. In this case, each value contains a count for the number of trailing spaces.

? table-lookup

The column had only a small number of different values, which were converted to an ENUM before Huffman compression.

? zero

The number of columns for which all values are zero.

? Original trees

The initial number of Huffman trees.

? After join

The number of distinct Huffman trees left after joining trees to save some header space.

After a table has been compressed, the Field lines displayed by myisamchk -dvv include additional information about each column:

? Type

The data type. The value may contain any of the following descriptors:

? constant

All rows have the same value.

? no endspace

Do not store endspace.

? no endspace, not_always

Do not store endspace and do not do endspace compression for all values.

? no endspace, no empty

Do not store endspace. Do not store empty values.

? table-lookup

The column was converted to an ENUM.

? zerofill(N)

The most significant N bytes in the value are always 0 and are not stored.

? no zeros

Do not store zeros.

? always zero

Zero values are stored using one bit.

? Huff tree

The number of the Huffman tree associated with the column.

? Bits

The number of bits used in the Huffman tree.

After you run myisampack, use myisamchk to re-create any indexes. At this time, you can also sort the index blocks and create statistics needed for the MySQL optimizer to work more efficiently:

myisamchk -rq --sort-index --analyze tbl_name.MYI

After you have installed the packed table into the MySQL database directory, you should execute mysqladmin flush-tables to force mysqld to start using the new table.

To unpack a packed table, use the --unpack option to myisamchk.

COPYRIGHT

published by the Free Software Foundation; version 2 of the License.

This documentation is distributed in the hope that it will be useful,

but WITHOUT ANY WARRANTY; without even the implied warranty of

MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU

General Public License for more details.

You should have received a copy of the GNU General Public License along

with the program; if not, write to the Free Software Foundation, Inc.,

51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA or see

http://www.gnu.org/licenses/.

SEE ALSO

For more information, please refer to the MySQL Reference Manual, which

may already be installed locally and which is also available online at

http://dev.mysql.com/doc/.

AUTHOR

Oracle Corporation (http://dev.mysql.com/).