

Periodically rebuilding your [Unified Index](#) is a best practice.

[Cron Job to Rebuild Search Index](#) should merge with current page.

Tiki is applying filters (and partially permissions to view) and in some case reading directly information from the Unified-Index. This allow very good performances and a lot of flexibility. Therefore, it is important to keep a fresh Unified-Index. [Features](#) but also [plugins](#) and [modules](#) will also relies on the freshness of the Unified-Index to display updated content (pages content, items, user informations, etc).

You can see when the index was last fully rebuilt usually by start a rebuild process from the Tiki interface (see below).

In recent Tiki version the index is stored in the database and you can rebuild (refresh) the Unified-Index with different method.

#### From the Tiki interface

It is possible to rebuild the Unified-Index from a browser. However due to possible timeout during long operation it is recommended to do this ONLY on relatively small Tiki (based on the number and the size of your Tiki objects). For medium to high load sites, it is strongly recommended do that operation from the command line (much less limited), see the next paragraph.

- From the admin dashboard, on the main tiki menu click on the item Control Panels, then click on the Search panel. At the top of the page you will find a **Rebuild Index** button.
- From your URL address bar `tiki-admin.php?page=search&rebuild=now`

#### From the command line

The search index can be rebuilt manually from the command line or you can use the Tiki [Scheduler](#) since [Tiki17](#) - see [Rebuild the Unified Index with log sample](#) or a [Cron](#) job since [Tiki9](#) to runs the command automatically - see [Cron Job to Rebuild Search Index](#).

Below are the commands that may be used to rebuild the index. Check [Console](#) documentation.

You can also rebuild it using the unified console.php command, with the appropriate parameters. For example:

#### Basic command

---

```
php console.php index:rebuild <-- or --> php console.php i:r <-- or add visual verbose, to follow what is going on --> php console.php i:r -p
```

#### Multitiki sites

For multitiki sites, you can rebuild with commands like:

---

```
php console.php index:rebuild --site=site1.example.com php console.php index:rebuild --site=site2.example.com ...
```

#### Successful rebuild

If the rebuild is successful a message like the following will be produced (for cron jobs, this can usually be sent to you via email as part of automatically running the command):

---

Started rebuilding index... Unified search ----- Engine: MySQL, version 5.7.34 Indexed  
trackeritem: 210 tracker: 13 trackerfield: 104 article: 67 file: 285 file gallery: 20 comment: 0 user: 8  
group: 7 wiki page: 55 category: 18 Rebuilding index done Index: index\_6291d4b25591f Execution  
Statistics ----- Execution time: 16 secs Current Memory usage: 51.0 MiB Memory peak usage  
before indexing: 22.5 MiB Memory peak usage after indexing: 51.2 MiB Number of queries: 12383

## Relations Tracker field

If you use the [Relations Tracker Field](#), and are starting from an empty index, you should probably rebuild twice.

## Troubleshooting

If the rebuild is unsuccessful, instead of the above message you may get a message that indicates there has been an internal server error, or it may say "Rebuild in progress." This may be because the rebuild process uses more memory or takes more time than allowed by the server's php settings. Such settings can be changed on the fly as part of the rebuild command - examples of how to do this are shown below.

## Not enough memory

### Increase memory limit

One way to increase memory is to change the `memory_limit` php setting as follows (this example changes the memory limit to 4 gigabytes while the rebuild process is running):

---

```
php -dmemory_limit=4G console.php i:r --log
```

You could also direct php to use a specific `php.ini` file, where there may be a higher memory limit setting or no limit. In this case you would use the `-c` parameter followed by the path to the `php.ini` file, as in the example below:

---

```
php -c /etc/php5/cli/php.ini console.php i:r --log
```

## Timeout error

### Increase maximum execution time

Getting an internal server error may indicate the rebuild process takes longer than the `max_execution_time` php setting. That can be increased as part of the command as shown below where the `max_execution_time` is set to 300 seconds, or 5 minutes. (This command is also increasing the memory limit as described above):

---

```
php -dmemory_limit=4G -dmax_execution_time=300 console.php i:r --log
```

## Force rebuild

When the rebuild is unsuccessful with a "Rebuild in progress" message, this usually means that the rebuild failed previously in the middle of the process, leaving a temporary folder called `temp/unified-index-new` on the server. When a new rebuild is started and the program sees this folder, it thinks there is a rebuild already in progress and will stop. You can either delete this folder before rebuilding again or include the `--force` parameter in the rebuild command as follows:

---

```
php -dmemory_limit=4G -dmax_execution_time=300 console.php i:r --force --log
```

## Log to review possible problems with content

---

```
php console.php i:r -p --log
```

## Several index\_ files in the database

Typically happening when using the same database over and over you may end with several indexes in your database.

We provide a developer tool to clean properly this issue.

You need to be the web user and use bash to run the script.

---

```
bash doc/devtools/clear_mysql_search_indexes.bash
```

You need to rebuild your index after the operation.

## Deprecated Zend\_Search\_Lucene

With the now deprecated Zend\_Search\_Lucene, the unified-index is stored in your Tiki directory `temp/unified-index/`. While the rebuild is occurring, a directory `temp/unified-index-new/` will appear (This is to permit the existing index to be used until the new one is ready). If `temp/unified-index-new/` doesn't disappear after the indexing, something must have gone wrong. You can delete it and try the re-indexing again. You may want to run *sh setup.sh* to make sure the permissions are OK.

alias names for this page

---

[IndexRebuild](#) | [Index Rebuild](#)