

Apache CouchDB 1.2.0 Release Notes

Apache CouchDB 1.2.0 has been released and is available for download.

Grab your copy here:

<http://couchdb.apache.org/>

Windows packages are now available. Grab them at the same download link.

This release also coincides with a revamped project homepage!

This is a big release with lots of updates. Please also note that this release contains [breaking changes](#).

These release notes are based on the [NEWS](#) file.

Performance

- **Added a native JSON parser**

Performance critical portions of the JSON parser are now implemented in C. This improves latency and throughput for all database and view operations. We are using the fabulous [yajl library](#).

- **Optional file compression (database and view index files)**

This feature is enabled by default.

All storage operations for databases and views are now passed through Google's [snappy compressor](#). The result is simple: since less data has to be transferred from and to disk and through the CPU & RAM, all database and view accesses are now faster and on-disk files are smaller. Compression can be changed to gzip compression with options that specify the compression ratio or it can be fully disabled as well.

- **Several performance improvements, especially regarding database writes and view indexing**

Combined with the two preceding improvements, we made some less obvious algorithmic improvements that take the Erlang runtime system into account when writing data to databases and view index files. The net result is much improved performance for most common operations including building views.

The JIRA ticket ([COUCHDB-976](#)) has more information.

- **Performance improvements for the built-in changes feed filters `_doc_ids` and `_design`**

Security

The security system got a major overhaul making it way more secure to run CouchDB as a public database server for CouchApps. Unfortunately we had to break a bit of backwards compatibility with this, but we think it is well worth the trouble.

- **Documents in the `_users` database can no longer be read by everyone**

Documents in the `_users` databases can now only be read by the respective authenticated user and administrators. Before, all docs were world-readable including their password hashes and salts.

- **Confidential information in the `_replication` database can no longer be read by everyone**

Similar to documents in the `_users` database, documents in the `_replicator` database now get passwords and OAuth tokens stripped when read by a user that is not the creator of the replication or an administrator.

- **Password hashes are now calculated by CouchDB instead of the client**

Previously, CouchDB relied on the client to hash and salt the user's password. Now, it accepts plain text passwords and hashes them before they are committed to disk, following traditional best practices.

- **Allow persistent authentication cookies**

Cookie based authentication can now keep a user logged in over a browser restart.

- **OAuth secrets can now be stored in the users system database**

This is better for managing large numbers of users and tokens than the old, clumsy way of storing OAuth tokens in the configuration system and configuration system.

- **Updated bundled `erlang_oauth` library to the latest version**

The Erlang library that handles OAuth authentication has been updated to the latest version.

Build System

- **cURL is no longer required to build CouchDB as it is only required by the command line JavaScript test runner**

This makes building CouchDB on certain platforms easier.

HTTP API

- **Added a `data_size` property to database and view group information URIs**

With this you can now calculate how much actual data is stored in a database file or view index file and compare it with the file size that is already being reported. The difference is CouchDB-specific

overhead most of which can be reclaimed during compaction. This is used to power the automatic compaction feature (see below).

- **Added optional field `since_seq` to replication objects/documents**

This allows you to start a replication from a certain database update sequence instead from the start.

- **The `_active_tasks` API now exposes more granular fields for each task type**

The replication and compaction tasks, e.g. report their progress in the task info.

- **Added built-in changes feed filter `_view`**

With this you can use a view's map function as a changes filter instead of duplicating.

Core Storage

- **Added support for automatic compaction**

This feature is disabled by default, but it can be enabled in the configuration page in Futon or the `.ini` files.

Compaction is a regular maintenance task for CouchDB. This can now be automated based on multiple variables:

- A threshold for the `file_size` to `disk_size` ratio (say 70%)
- A time window specified in hours and minutes (e.g 01:00-05:00)

Compaction can be cancelled if it exceeds the closing time. Compaction for views and databases can be set to run in parallel, but that is only useful for setups where the database directory and view directory are on different disks.

In addition, if there's not enough space ($2 \times \text{data_size}$) on the disk to complete a compaction, an error is logged and the compaction is not started.

Replicator

- **A new replicator implementation that offers more performance and configuration options**

The replicator has been rewritten from scratch. The new implementation is more reliable, faster and has more configuration than the previous implementation. If you have had any issues with replication in previous releases, we strongly recommend giving 1.2.0 a spin.

Configuration options include:

- Number of worker processes
- Batch size per worker

- Maximum number of HTTP connections
- Number of connection retries

See `default.ini` for the full list of options and their default values.

This allows you to fine-tune replication behaviour tailored to your environment. A spotty mobile network connection can benefit from a single worker process and small batch sizes to reliably, albeit slowly, synchronise data. A full-duplex 10GigE server-to-server connection on a LAN can benefit from more workers and higher batch sizes. The exact values depend on your particular setup and we recommend some experimentation before settling on a set of values.

Futon

- **Futon's Status screen (active tasks) now displays two new task status fields: `Started on` and `Updated on`**
- **Simpler replication cancellation**

Running replications can now be cancelled with a single click.

Log System

- **Log correct stack trace in all cases**

In certain error cases, CouchDB would return a stack trace from the log system itself and hide the real error. Now CouchDB always returns the correct error.

- **Improvements to log messages for file-related errors**

CouchDB requires correct permissions for a number of files. Error messages related to file permission errors were not always obvious and are now improved.

Various Bugfixes

- Fixed old index file descriptor leaks after a view cleanup
- Fixes to the `_changes` feed heartbeat option when combined with a filter. It affected continuous pull replications with a filter
- Fix use of OAuth with VHosts and URL rewriting
- The `requested_path` property of query server request objects now has the path requested by clients before VHosts and rewriting
- Fixed incorrect reduce query results when using pagination parameters
- Made `icu_driver` work with Erlang R15B and later
- Improvements to the build system and etap test suite
- Avoid invalidating view indexes when running out of file descriptors

Breaking Changes

This release contains breaking changes:

http://wiki.apache.org/couchdb/Breaking_changes

It is very important that you understand these changes before you upgrade.

More Information

See the [CHANGES](#) file and the Git [commit log](#) for more information.