RediSearch 2.0 release notes

Requirements

RediSearch v2.0.13 requires:

- Minimum Redis compatibility version (database): 6.0.0
- Minimum Redis Enterprise Software version (cluster): 6.0.0

v2.0.15 (December 2021)

This is a maintenance release for RediSearch 2.0.

Update urgency: MODERATE - Program an upgrade of the server, but it's not urgent.

Details:

- Bug fixes:
  - #2388 Garbage Collection (GC) for empty ranges in numeric index
  - #2409 Introduction of FORK_GC_CLEAN_NUMERIC_EMPTY_NODES true module argument to enable #2388 (off by default)
  - #325 Used Redis allocator in hiredis (RSCoordinator)
  - #2362 Crash on empty field name

v2.0.13 (November 2021)

This is a maintenance release for version 2.0.

Details:

- Enhancements:
  - #2243 Add LOAD * for FT.AGGREGATE which will load all fields

- Bug fixes:
  - #2269 #2291 Remove TAG values from trie with no entries on Garbage Collection.
  - #2287 Uninitialized read on FT. ADD
  - #2342 Check for NULL result on intersect iterator
  - #2350 Crash on FT. AGGREGATE with LIMIT 0 0

v2.0.12 (September 2021)

This is a maintenance release for version 2.0.

Details:

- Enhancements:
  - #2184 API: getter functions for score, language and stop words list
  - #2188 Introduced the UNF parameter to SORTABLE to disable normalisation on TAG/TEXT fields
  - #2218 API: added RediSearch_CreateDocument2
Bug fix:
- #2153 Restore FT.INFO complexity to O(1)
- #2203 FT.AGGREGATE returns inaccurate results when TAG field is not set in hash

v2.0.11 (August 2021)
This is a maintenance release for version 2.0.
Details:
- Enhancements:
  - #2156 TAG fields can now be case sensitive using the CASESENSITIVE parameter
  - #2113 An already existing document that can’t be updated is removed from the index (JIRA MOD-1266)
  - #267 #287 Updated Hiredis to support Intershard TLS
- Bug Fixes:
  - #2117 #2115 Fix crash on coordinator on first value reducer

v2.0.10 (July 2021)
This is a maintenance release for version 2.0.
Details:
- Enhancements:
  - #2025 Improve performances on numeric range search
  - #1958 #2033 Support of sortable on the GEO type
  - #2079 Update to Snowball 2.1.0, adds Armenian, Serbian and Yiddish stemming support
  - #2002 Add stopwords list support in the API
- Bug fix:
  - #2045 Possible crash when loading an RDB file (silently ignore double load of alias)
  - #2099 #2101 Fixes possible crash with CRDT on FT.DROPINDEX
  - #1994 Skip intersect iterator qsort if INORDER flag is used
  - #257 Switch coordinator to send _FT.CURSOR instead FT.CURSOR to prevent data access without holding the lock

v2.0.9 (May 2021)
This is a maintenance release for version 2.0.
Details:
- Bug fix in RSCoordinator:
  - #259: Fix deadlock on cursor read by performing cursor command on background thread

v2.0.8 (May 2021)
This is a maintenance release for version 2.0.
This release fixes an important regression introduced by the 2.0 release. The payload is supposed to be returned only when the WITHPAYLOADS parameter is set.
Details:

- Bug fixes:
  - #1959 Renamed parse_time() to parsetime()
  - #1932 Fixed crash resulted from LIMIT arguments
  - #1919 Prevent GC fork from crashing

- Minor enhancements:
  - #1880 Optimisation of intersect iterator
  - #1914 Do not return payload as a field

**v2.0.7 (May 2021)**

This is a maintenance release for version 2.0.

Details:

- Major enhancements:
  - #1864 Improve query time by predetermining reply array length.
  - #1879 Improve loading time by calling RM_ScanKey instead of RM_Call

- Major bug fix:
  - #1866 Fix a linking issue causing incompatibility with Redis 6.2.0.
  - #1842 #1852 Fix macOS build.

- Minor bug fixes:
  - #1850 Fix a race condition on drop temporary index.
  - #1855 Fix a binary payload corruption.
  - #1876 Fix crash if the depth of the reply array is larger than 7.
  - #1843 #1860 Fix low-level API issues.

**v2.0.6 (February 2021)**

This is a maintenance release for version 2.0.

Details:

- Minor additions:
  - #1696 The maximum number of results produced by FT.AGGREGATE is now configurable: MAXAGGREGATERESULTS.
  - #1708 Stemming updated with support of new languages: Basque, Catalan, Greek, Indonesian, Irish, Lithuanian, Nepali.

- Minor bugfixes:
  - #1668 Fixes support of stop words in tag fields. Solves also the following related issues: #166, #984, #1237, #1294.
  - #1689 Consistency fix and performance improvement when using FT.SUGGET with RSCoordinator.
  - #1774 MINPREFIX and MAXFILTEREXPANSION configuration options can be changed at runtime.
  - #1745 Enforce 0 value for REDUCER COUNT.
  - #1757 Return an error when reaching the maximum number of sortable fields, instead of crashing.
  - #1762 Align the maximum number of sortable fields with the maximum number of fields (1024)

**v2.0.5 (December 2020)**

This is a maintenance release for version 2.0.

Details:
Minor features:

- #1696 Add MAXAGGREGATERESULTS module configuration for FT.AGGREGATE. Similar to MAXSEARCHRESULTS for FT.SEARCH, it limits the maximum number of results returned.

**v2.0.4 (December 2020)**

This is a maintenance release for version 2.0.

Details:

- Bugfixes in RediSearch:
  - #1668 Stopwords are not filtered out on tag fields.

- Bugfixes in RSCoordinator:
  - #206 FT.AGGREGATE with LIMIT and offset greater than 0 returned fewer results than requested.

**v2.0.3 (November 2020)**

This is a maintenance release for version 2.0.

Minor bugfixes:

- Added OSS_GLOBAL_PASSWORD config argument to allow specify shards password on OSS cluster.
- Update min_redis_pack_version to 6.0.8

**v2.0.2 (November 2020)**

This is a maintenance release for version 2.0.

- Minor enhancements:
  - #1625 MAXPREFIXEXPANSIONS configuration which replaces the now deprecated config MAXEXPANSIONS. Same behaviour, more declarative naming.
  - #1614 Prevent multiple sortby steps on FT.AGGREGATE

- Minor bug fixes:
  - #1605 Rare bug where identical results would get a lower score

**v2.0.1 (October 2020)**

This is a maintenance release for version 2.0.

- Minor additions:
  - #1432 FT.AGGREGATE allows filtering by key and returning the key by using LOAD "@_key".

- Minor bug fixes:
  - #1571 Using FILTER in FT.CREATE might cause index to not be in sync with data.
  - #1572 Crash when using WITHSORTKEYS without SORTBY.
  - #1540 SORTBY should raise an error if the field is not defined as SORTABLE.

**v2.0.0 (September 2020)**

RediSearch 2.0 is a public preview release meeting GA standards. This release includes several improvements in performance and...
usability over RediSearch 1.0. These improvements necessitate a few backward-breaking changes to the API.

**Highlights**

For this release, we changed the way in which the search indexes are kept in sync with your data. In RediSearch 1.x, you had to manually add data to your indexes using the `FT.ADD` command. In RediSearch 2.x, your data is indexed automatically based on a key pattern.

These changes are designed to enhance developer productivity, and to ensure that your search indexes are always kept in sync with your data. To support this, we’ve made a few changes to the API.

In addition to simplifying indexing, RediSearch 2.0 allows you to scale a single index over multiple Redis shards using the Redis cluster API.

Finally, RediSearch 2.x keeps its indexes outside of the main Redis key space. Improvements to the indexing code have increased query performance 2.4x.

You can read more details in the RediSearch 2.0 announcement blog post, and you can get started by checking out this quick start blog post.

---

**Details**

- When you create an index, you must specify a prefix condition and/or a filter. This determines which hashes RediSearch will index.

- Several RediSearch commands now map to their Redis equivalents: `FT.ADD` -> `HSET`, `FT.DEL` -> `DEL` (equivalent to `FT.DEL with the DD flag in RediSearch 1.x`), `FT.GET` -> `HGETALL`, `FT.MGET` -> `HGETALL`.

- RediSearch indexes no longer reside within the key space, and the indexes are no longer saved to the RDB.
You can upgrade from RediSearch 1.x to RediSearch 2.x.

Noteworthy changes

- **#1246**: `geodistance` function for `FT.AGGREGATE APPLY` operation.
- **#1394**: Expired documents (TTL) will be removed from the index.
- **#1394**: Optimization to avoid reindexing documents when non-indexed fields are updated.
- After index creation, an initial scan starts for existing documents. You can check the status of this scan by calling `FT.INFO` and looking at the `indexing` and `percent_indexed` values. While indexing is true, queries return partial results.
- **#1435**: `NOINITIALINDEX` flag on `FT.CREATE` to skip the initial scan of documents on index creation.
- **#1401**: Support upgrade from v1.x and for reading RDB’s created by RediSearch 1.x (more information).
- **#1445**: Support for load event. This event indexes documents when they are loaded from RDB, ensuring that indexes are fully available when RDB loading is complete (available from Redis 6.0.7 and above).
- **#1384**: `FT.DROPINDEX`, which by default does not delete documents underlying the index (see deprecated `FT.DROP`).
- **#1385**: Add index definition to `FT.INFO` response.
- **#1097**: Add Hindi snowball stemmer.

The `FT._LIST` command returns a list of all available indices. Note that this is a temporary command, as indicated by the _ in the name, so it’s not documented. We’re working on a SCAN-like command for databases with many indexes.

RediSearch will appear in Redis as 20000, which is equivalent to 2.0.0 in semantic versioning. Since the version of a module in Redis is numeric, we cannot explicitly add an GA flag.

Behavior changes

Make sure you review these changes before upgrading to RediSearch 2.0:

- **#1381**: `FT.SYNADD` is removed; use `FT.SYNUPDATE` instead. `FT.SYNUPDATE` requires both and index name and a synonym group ID. This ID can be any ASCII string.
- **#1437**: Documents that expire during query execution time will not appear in the results (but might have been counted in the number of produced documents).
- **#1221**: Synonyms support for lower case. This can result in a different result set on `FT.SEARCH` when using synonyms.

RediSearch will not index hashes whose fields do not match an existing index schema. You can see the number of hashes not indexed using `FT.INFO - hash_indexingfailures`. The requirement for adding support for partially indexing and blocking is captured here: #1455.

- Removed support for NOSAVE (for details see v1.6 docs).
- RDB loading will take longer due to the index not being persisted.
- Field names in the query syntax are now case-sensitive.

Deprecated commands:
- `FT.DROP` (replaced by `FT.DROPINDEX`, which by default keeps the documents)
- `FT.ADD` (mapped to `HSET` for backward compatibility)
- `FT.DEL` (mapped to `DEL` for backward compatibility)
- `FT.GET` (mapped to `HGETALL` for backward compatibility)
- **FT.MGET** (mapped to HGETALL for backward compatibility)

- **Removed commands:**
  - **FT.ADDHASH** (no longer makes sense)
  - **FT.SYNADD** (see #1381)
  - **FT.OPTIMIZE** (see v1.6 docs)

---

**Updated:** December 14, 2021