Compatibility Changes in MongoDB 6.0

This page describes changes introduced in MongoDB 6.0 that can affect compatibility with older versions of MongoDB.

MongoDB 6.0 is a Major Release, which means that it is supported for both MongoDB Atlas and on-premises deployments. MongoDB 6.0 includes changes introduced in MongoDB Rapid Releases 5.1, 5.2, and 5.3. This page describes compatibility changes introduced in those Rapid Releases and MongoDB 6.0.

To learn more about the differences between Major and Rapid releases, see MongoDB Versioning.

Aggregation

`allowDiskUse` Changes

Starting in MongoDB 6.0, pipeline stages that require more than 100 megabytes of memory to execute write temporary files to disk by default. In earlier verisons of MongoDB, you must pass `{ allowDiskUse: true }` to individual `find` and `aggregate` commands to enable this behavior.

Individual `find` and `aggregate` commands may override the `allowDiskUseByDefault` parameter by either:

- Using `{ allowDiskUse: true }` to allow writing temporary files out to disk when `allowDiskUseByDefault` is set to `false`
- Using `{ allowDiskUse: false }` to prohibit writing temporary files out to disk when `allowDiskUseByDefault` is set to `true`

`$$SEARCH_META` Limitations

Starting in MongoDB 6.0, the Atlas Search `$$SEARCH_META` aggregation variable can be used anywhere after a `$search` stage in any pipeline, but it cannot be used after the `$lookup` or `$unionWith` stage in any pipeline. The `$$SEARCH_META` aggregation variable cannot be used in any subsequent stage after a `$searchMeta` stage.

Change Streams

Starting in MongoDB 5.3, during range migration, change stream events are not generated for updates to orphaned documents.

Filters

Starting in MongoDB 6.0, whenever possible, match filters are applied to change streams earlier than in prior releases. This improves performance. However, when a filter is narrowly defined, an earlier match may cause an operation that succeeds in prior versions to fail in 6.0.

Indexes

Last Remaining Shard Key Index Cannot be Dropped Inadvertently

Starting in MongoDB 6.0, passing `"*"` to `dropIndexes` or `db.collection.dropIndexes()` drops all indexes **except** for the `_id` index and the last remaining shard key index, if one exists. Attempts to explicitly drop the last remaining shard key index raise an error.

Existing Indexes Can Be Dropped During an Index Build

Starting in MongoDB 5.2, you can use `dropIndexes` or `db.collection.dropIndexes()` to drop existing indexes on the same collection even if there is an index build in progress. In earlier versions, attempting to drop a different index during an in-progress index build results in a `BackgroundOperationInProgressForNamespace` error.

2dsphere Document Index Keys

To prevent out of memory errors, `indexMaxNumGeneratedKeysPerDocument` limits the maximum number of 2dsphere index keys generated for a single document.

See `indexMaxNumGeneratedKeysPerDocument`.

Index Key Format

Starting in MongoDB 6.0, a change to the unique index key format was introduced. If you create a unique index in MongoDB 6.0, the index won't work with MongoDB versions earlier than 5.3.2 or 5.0.7.

Legacy `mongo` Shell Removed

The `mongo` shell is removed from MongoDB 6.0. The replacement is `mongosh`.

Platform Support

Starting in MongoDB 5.1.2 the following platforms are no longer supported:

Community Edition

- RHEL-72-s390x

Regular Expressions

`$regex` Find Queries No Longer Ignore Invalid Regex

Starting in MongoDB 5.1, invalid `$regex options` options are no longer ignored. This change makes `$regex options` more consistent with the use of `$regex` in the `aggregate` command and projection queries.

`$regex` Schema Validation Error Behavior

Starting in MongoDB 5.1, if a collection has schema validation rules that contain invalid `$regex options` the server:

- Prevents all insert and update operations until the schema validation rules containing the invalid regex pattern are modified with the `collMod` command.
- Writes a warning error to the `mongod` log file.

Removed Operators

Starting in MongoDB 5.1, these legacy query operators are removed:

Removed Operator	Alternative
$comment	`cursor.comment()`
$explain	`cursor.explain()`
$hint	`cursor.hint()`
$max	`cursor.max()`
$maxTimeMS	`cursor.maxTimeMS()`
$min	`cursor.min()`
$orderby	`cursor.sort()`
$query	See Cursor Methods
$returnKey	`cursor.returnKey()`
$showDiskLoc	`cursor.showRecordId()`
`db.getLastError()`	See Legacy Opcodes Removed
`db.getLastErrorObj()`	See Legacy Opcodes Removed
`getLastError`	See Legacy Opcodes Removed

Removed Options

MongoDB 6.0 removes the `--cpu` `mongod` option.

Removed Parameters

MongoDB 6.0 removes the following server parameters:

Removed Parameter	Description
`--tlsFIPSMode`	This option is removed from the MongoDB Community Edition. It is available in MongoDB Enterprise edition. FIPS was not a supported feature in MongoDB Community Edition. If your installation made use of FIPS anyway, you will need to reconfigure your TLS/SSL connections before upgrading.

Renamed Parameters

Starting in MongoDB 6.0, the following parameters have been renamed:

- `wiredTigerConcurrentReadTransactions` is now `storageEngineConcurrentReadTransactions`
- `wiredTigerConcurrentWriteTransactions` is now `storageEngineConcurrentWriteTransactions`

TTL `expireAfterSeconds` Behavior When Set to `NaN`

Setting TTL `expireAfterSeconds` to `NaN` experiences a behavior change from MongoDB 4.4 to MongoDB 6.0 that affects initial sync from MongoDB 4.4 and earlier and `mongorestore` from MongoDB 4.4 and earlier. Performing any of those actions causes an `expireAfterSeconds` of `NaN` to be treated as an `expireAfterSeconds` of `0`. Immediate document expiration may occur as a result.

Replica Sets

Assert Cluster Wide Write Concern is Set When Starting or Adding Shard

Starting in MongoDB 5.1, when starting, restarting or adding a shard server with `sh.addShard()` the Cluster Wide Write Concern (CWWC) must be set.

If the `CWWC` is not set and the shard is configured such that the default write concern is `{ w : 1 }` the shard server will fail to start or be added and returns an error.

See default write concern calculations for details on how the default write concern is calculated.

`rs.reconfig` Cluster Wide Write Concern Validation

Starting in MongoDB 5.1, you must set the Cluster Wide Write Concern (CWWC) prior to issuing any `reconfigs` that would otherwise change the default write concern of the new replica set member.

Security

Intra-Cluster Authentication

Starting in MongoDB 5.3, SCRAM-SHA-1 cannot be used for intra-cluster authentication. Only SCRAM-SHA-256 is supported.

In previous MongoDB versions, SCRAM-SHA-1 and SCRAM-SHA-256 can both be used for intra-cluster authentication, even if SCRAM is not explicitly enabled.

FIPS Mode Defaults SCRAM-SHA-1 Authentication to Off

Starting in MongoDB 5.1, instances running in FIPS mode have the SCRAM-SHA-1 authentication mechanism disabled by default. You can enable the SCRAM-SHA-1 authentication mechanism with the setParameter.authenticationMechanisms command.

This change will not affect drivers which target MongoDB `setFeatureCompatibilityVersion` 4.0+.

OCSP Must be Enabled

Starting in MongoDB 6.0, if `ocspEnabled` is set to `true` during initial sync, all nodes must be able to reach the OCSP responder.

If a member fails in the `STARTUP2` state, set `tlsOCSPVerifyTimeoutSecs` to a value that is less than `5`.

Time Series Collections

> **WARNING**
>
> If you create a sharded time series collection in MongoDB 5.1 or greater, downgrading to a version older than MongoDB 5.0.4 will result in data loss.

Before downgrading to a version older than 5.0.4, drop all sharded time series collections.

Secondary Indexes on Time Series Collections

If there are secondary indexes on time series collections and you need to downgrade the feature compatibility version (fCV), you must first drop any secondary indexes that are incompatible with the downgraded fCV. See `setFeatureCompatibilityVersion`.

General Changes

Deprecations

Deprecated	Description
`db.collection.reIndex()`	The `db.collection.reIndex()` method is deprecated in MongoDB v6.0.
`reIndex`	The `reIndex` command is deprecated in MongoDB v6.0.
Simple Network Management Protocol (SNMP)	Starting in MongoDB 6.0, SNMP is deprecated and will be removed in the next release. To monitor your deployment, use MongoDB Ops Manager.

`$mod` Error Behavior

Starting in MongoDB 5.1 (and 5.0.4), the `$mod` operator returns an error if the `divisor` or `remainder` values evaluate to certain values. See $mod behavior.

Legacy Opcodes Removed

MongoDB 6.0 removes support for the following legacy opcodes and database commands:

- OP_INSERT
- OP_DELETE
- OP_UPDATE

- OP_KILL_CURSORS
- OP_GET_MORE
- OP_QUERY
- `getLastError`

> ⚠ **WARNING**
>
> **Upgrade Drivers**
> To avoid disruption due to the removal of these opcodes, **upgrade your driver to the latest version.**

If you attempt to connect to a MongoDB 3.4 or older `mongod` instance with a MongoDB 5.1 or newer `mongo` shell, you will receive an error message like the following:

```
Connection handshake failed. Is your mongod 3.4 or older?
:: caused by :: network error while attempting to run command
'isMaster' on host '127.0.0.1:27017'
```

mongod Responses to Legacy Opcodes

Since MongoDB 3.6, MongoDB drivers have used OP_MSG instead of OP_QUERY and the other legacy opcodes and commands.

Starting in MongoDB 6.0:

- `mongod` will close the connection and will not respond to:
  - OP_INSERT
  - OP_DELETE
  - OP_UPDATE
  - OP_KILL_CURSORS

- `mongod` will keep the connection open and return an error for:
  - The `getLastError` database command
  - OP_GET_MORE
  - OP_QUERY finds
  - Most OP_QUERY RPC command messages

> **NOTE**
>
> **OP_QUERY RPC Commands**
>
> The OP_QUERY RPC protocol may be used with the following commands:
>
> - `_isSelf`
> - `authenticate`
> - `buildinfo`
> - `buildInfo`
> - `hello`
> - `ismaster`
> - `isMaster`
> - `saslContinue`
> - `saslStart`
>
> All other commands will be rejected if issued as OP_QUERY.

Removed Deprecated Array and String Functions for Server-Side JavaScript

MongoDB 6.0 upgrades the internal JavaScript engine used for server-side JavaScript, `$accumulator`, `$function`, and `$where` expressions and from MozJS-60 to MozJS-91. Several deprecated, non-standard array and string functions that existed in MozJS-60 are removed in MozJS-91.

For the complete list of removed array and string functions, see the next sections on this page.

NOTE

Only Static Functions are Removed

Only *static* JavaScript functions are removed. *Prototype function* equivalents of the removed functions can still be used.

For example:

- `Array.concat(<array1>, <array2>)` is a static function and no longer works in MongoDB 6.0.
- `<array1>.concat(<array2>)` is a prototype function and still works in MongoDB 6.0.

This behavior applies to both removed array and removed string functions.

Removed Array Functions

Starting in MongoDB 6.0, the following array functions are removed and cannot be used in server-side JavaScript with `$accumulator`, `$function`, and `$where` expressions:

- `Array.concat`
- `Array.every`
- `Array.filter`
- `Array.forEach`
- `Array.indexOf`
- `Array.join`
- `Array.lastIndexOf`
- `Array.map`
- `Array.pop`
- `Array.push`
- `Array.reduce`

- `Array.reduceRight`
- `Array.reverse`
- `Array.shift`
- `Array.slice`
- `Array.some`
- `Array.sort`
- `Array.splice`
- `Array.unshift`

Removed String Functions

Starting in MongoDB 6.0, the following array functions are removed and cannot be used in server-side JavaScript with `$accumulator`, `$function`, and `$where` expressions:

- `String.charAt`
- `String.charCodeAt`
- `String.concat`
- `String.contains`
- `String.endsWith`
- `String.includes`
- `String.indexOf`
- `String.lastIndexOf`
- `String.localeCompare`
- `String.match`
- `String.normalize`
- `String.replace`
- `String.search`

- `String.slice`
- `String.split`
- `String.startsWith`
- `String.substr`
- `String.substring`
- `String.toLocaleLowerCase`
- `String.toLocaleUpperCase`
- `String.toLowerCase`
- `String.toUpperCase`
- `String.trim`
- `String.trimLeft`
- `String.trimRight`

Default `db.stats()` Settings

Starting in MongoDB 6.0, the `dbStats` command and the `db.stats()` method only report free space assigned to collections if the freeStorage parameter is set to 1.

Index Filters and Collations

Starting in MongoDB 6.0, an index filter uses the collation previously set using the `planCacheSetFilter` command.

Arrays in Collections and Views with `distinct` Command

Starting in MongoDB 6.0, the `distinct` command returns the same results for collections and views when using arrays.

See Arrays in Collections and Views.

Downgrade Considerations

The following sections provide information for removing backward-incompatible features from your deployment. If you are downgrading from MongoDB 6.0 to an earlier version, review the following sections to ensure that your deployment runs successfully after downgrading.

Clustered Collections

Starting in MongoDB 5.3, if you are using clustered collections, you must drop those collections before you can downgrade to an earlier MongoDB version.

User Write Blocking

Starting in MongoDB 6.0, if you need to downgrade the feature compatibility version, ensure you disable cluster-to-cluster replication and user write blocking.

See Cluster-to-Cluster Sync and User Write Blocking.

Time Series Collections

You must drop time series collections before downgrading:

- MongoDB 6.0 or later to MongoDB 5.0.7 or earlier.
- MongoDB 5.3 to MongoDB 5.0.5 or earlier.

See Time Series Collections.

Cluster Parameters

Starting in MongoDB 6.0, ensure that all `setClusterParameter` operations have completed. fCV downgrade cannot occur successfully if there are any ongoing `setClusterParameter` operations on sharded clusters.

SELinux Policy Data

Starting in MongoDB 5.1, you must run the following command from the directory into which the SELinux policy was previously cloned before you can downgrade to an earlier MongoDB version:

```
sudo make uninstall
```

Key Management Interoperability Protocol (KMIP) Settings

Starting in MongoDB 5.3 Enterprise, if you are using the following KMIP settings, you must remove them from the configuration file before you can downgrade to an earlier MongoDB version:

- `security.kmip.keyStatePollingSeconds`
- `security.kmip.activateKeys`

Time-based Retention of Change Streams Pre- and Post-Image Collections

Starting in MongoDB 6.0, if you are using `changeStreamOptions.preAndPostImages.expireAfterSeconds` to control time-based retention of change streams pre- and post-image collections, you must ensure there are no active `setClusterParameter` operations when downgrading.

Audit Log Encryption Settings

Starting in MongoDB 6.0 Enterprise, if you are using audit log encryption, you must remove the following settings from the configuration file before you can downgrade to an earlier MongoDB version:

- `auditLog.auditEncryptionKeyIdentifier`
- `auditLog.localAuditKeyFile`

Existing encrypted audit logs remain encrypted, and you can keep any procedures you have developed for storage and processing of encrypted logs.

See Audit Log.

Change Streams with Document Pre- and Post-Images

Starting in MongoDB 6.0, if you are using document pre- and post-images for change streams, you must disable changeStreamPreAndPostImages for each collection using the `collMod` command before you can downgrade to an earlier MongoDB version.

Change Streams with Expanded Events

If your application uses change streams, ensure that it does not require the `showExpandedEvents` option, which will not be available after downgrade.

LDAP with `srv:` and `srv_raw:`

If your cluster's configuration is using the new `"srv:"` or `"srv_raw:"` URL types in its LDAP configuration, it will be unable to restart after a downgrade. Remove the new URL types from your cluster's configuration before or downgrading.

Collections with Encrypted Fields

You must drop collections that use encrypted fields before you can complete the fCV downgrade. The downgrade will not complete if there are collections using `encryptedFields`.

Manage Cookies

Social

GitHub

LinkedIn

X

Facebook

Stack Overflow

YouTube

Twitch