

---

# MySQL NDB Cluster 9.2 Release Notes

## Abstract

This document contains release notes for the changes in each release of MySQL NDB Cluster that uses version 9.2 of the [NDB \(NDBCLUSTER\)](#) storage engine.

Each NDB Cluster 9.2 release is based on a mainline MySQL Server release and a particular version of the [NDB](#) storage engine, as shown in the version string returned by executing `SELECT VERSION()` in the `mysql` client, or by executing the `ndb_mgm` client `SHOW` or `STATUS` command; for more information, see [MySQL NDB Cluster 9.2](#).

For general information about features added in NDB Cluster 9.2, see [What is New in MySQL NDB Cluster 9.2](#). For a complete list of all bug fixes and feature changes in MySQL NDB Cluster, please refer to the changelog section for each individual NDB Cluster release.

For additional MySQL 9.2 documentation, see the [MySQL 9.2 Reference Manual](#), which includes an overview of features added in MySQL 9.2 that are not specific to NDB Cluster ([What Is New in MySQL 9.2](#)), and discussion of upgrade issues that you may encounter for upgrades from MySQL 9.1 to MySQL 9.2 ([Changes in MySQL 9.2](#)). For a complete list of all bug fixes and feature changes made in MySQL 9.2 that are not specific to [NDB](#), see [MySQL 9.2 Release Notes](#).

Updates to these notes occur as new product features are added, so that everybody can follow the development process. If a recent version is listed here that you cannot find on the download page (<https://dev.mysql.com/downloads/>), the version has not yet been released.

The documentation included in source and binary distributions may not be fully up to date with respect to release note entries because integration of the documentation occurs at release build time. For the most up-to-date release notes, please refer to the online documentation instead.

For legal information, see the [Legal Notices](#).

For help with using MySQL, please visit the [MySQL Forums](#), where you can discuss your issues with other MySQL users.

Document generated on: 2025-07-03 (revision: 30212)

## Table of Contents

Preface and Legal Notices .....	1
Changes in MySQL NDB Cluster 9.2.0 (2025-01-22, Innovation Release) .....	3
Index .....	6

## Preface and Legal Notices

This document contains release notes for the changes in each release of MySQL NDB Cluster that uses version 9.2 of the [NDB](#) storage engine.

### Legal Notices

Copyright © 1997, 2025, Oracle and/or its affiliates.

#### License Restrictions

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

#### Warranty Disclaimer

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

### **Restricted Rights Notice**

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

### **Hazardous Applications Notice**

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

### **Trademark Notice**

Oracle, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

### **Third-Party Content, Products, and Services Disclaimer**

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

### **Use of This Documentation**

This documentation is NOT distributed under a GPL license. Use of this documentation is subject to the following terms:

You may create a printed copy of this documentation solely for your own personal use. Conversion to other formats is allowed as long as the actual content is not altered or edited in any way. You shall not publish or distribute this documentation in any form or on any media, except if you distribute the documentation in a manner similar to how Oracle disseminates it (that is, electronically for download on a Web site with the software) or on a CD-ROM or similar medium, provided however that the

documentation is disseminated together with the software on the same medium. Any other use, such as any dissemination of printed copies or use of this documentation, in whole or in part, in another publication, requires the prior written consent from an authorized representative of Oracle. Oracle and/or its affiliates reserve any and all rights to this documentation not expressly granted above.

## Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

## Access to Oracle Support for Accessibility

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

## Changes in MySQL NDB Cluster 9.2.0 (2025-01-22, Innovation Release)

MySQL NDB Cluster 9.2.0 is a new Innovation release of NDB Cluster, based on MySQL Server 9.2 and including features in version 9.2 of the [NDB](#) storage engine, as well as fixing recently discovered bugs in previous NDB Cluster releases.

**Obtaining MySQL NDB Cluster 9.2.** NDB Cluster 9.2 source code and binaries can be obtained from <https://dev.mysql.com/downloads/cluster/>.

For an overview of major changes made in NDB Cluster 9.2, see [What is New in MySQL NDB Cluster 9.2](#).

This release also incorporates all bug fixes and changes made in previous NDB Cluster releases, as well as all bug fixes and feature changes which were added in mainline MySQL 9 through MySQL 9.2.0 (see [Changes in MySQL 9.2.0 \(2025-01-21, Innovation Release\)](#)).

- [Compilation Notes](#)
- [Bugs Fixed](#)

## Compilation Notes

- **macOS:** A `uint64_t` value used with `%zu` caused a `[-Wformat]` compiler warning on MacOS. (Bug #37174692)
- Removed a warning in `storage/ndb/src/common/util/cstrbuf.cpp`. (Bug #37049014)
- The file `storage/ndb/tools/restore/consumer.cpp` contained dead code (only), and has been removed. (Bug #36237561)

## Bugs Fixed

- **Microsoft Windows:** Successive iterations of the sequence `ndb_sign_keys --create-key` followed by `ndb_sign_keys --promote` were unsuccessful on Windows. (Bug #36951132)
- **NDB Disk Data:** `mysqld` did not use a disk scan for [NDB](#) tables with 256 disk columns or more. (Bug #37201922)
- **NDB Replication:** The replication applier normally retries temporary errors occurring while applying transactions. Such retry logic is not performed for transactions containing row events where the `STMT_END_F` flag is missing; instead, the statement is committed in an additional step while applying

the subsequent `COMMIT` query event when there are still locked tables. Problems arose when committing this statement, because temporary errors were not handled properly. Replica skip error functionality was also affected in that it attempted to skip only the error that occurred when a transaction was committed a second time.

The binary log contains an epoch transaction with writes from multiple server IDs on the source. The replica then uses `IGNORE_SERVER_IDS` (`<last_server_id_in_binlog>`) to cause the `STMT_END_F` to be filtered away, thus committing the statement from the `COMMIT` query log event on the applier. Holding a lock on one of the rows to be updated by the applier triggered error handling, which caused replication to stop with an error, with no retries being performed.

We now handle such errors, logging all messages in diagnostics areas (as is already done for row log events) and then retrying the transaction. (Bug #37331118)

- **NDB Replication:** When a MySQL server performing binary logging connects to an NDB Cluster, it checks for existing binary logs; if it finds any, it writes an `Incident` event to a log file of its own so that any downstream replicas can detect the potential for lost events. Problems arose under some circumstances because it was possible for the timestamps of events logged in this file to be out of order; the `Incident` event was written following other events but had a smaller timestamp than these preceding events. We fix this issue by ensuring that a fresh timestamp is used prior to writing an incident to the binary log on startup rather than one which may have been obtained and held for some time previously. (Bug #37228735)
- **NDB Cluster APIs:** The `Ndb_cluster_connection` destructor calls `g_eventLogger::stopAsync()` in order to release the buffers used by the asynchronous logging mechanism as well as to stop the threads responsible for this logging. When the `g_eventLogger` object was deleted before the `Ndb_cluster_connection` destructor was called, the application terminated after trying to use a method on a null object. This could happen in either of two ways:
  - An API program deleted the logger object before deleting the `Ndb_cluster_connection`.
  - `ndb_end()` was called before the `Ndb_cluster_connection` was deleted.

We solve this issue by skipping the call to `stopAsync()` in the `Ndb_cluster_connection` destructor when `g_eventLogger` is `NULL`. This fix also adds a warning to inform API users that deleting `g_eventLogger` before calling the `Ndb_cluster_connection` destructor is incorrect usage.

For more information, see [API Initialization and Cleanup](#). (Bug #37300558)

- **NDB Cluster APIs:** Removed known causes of API node versus data node state misalignments, and improved the handling of state misalignments when detected. In one such case, separate handling of scan errors in the `NDB` kernel and those originating in API programs led to cleanup not being performed after some scans. Handling of `DBTC` and API state alignment errors has been improved by this set of fixes, as well as scan protocol timeout handling in `DBSPJ`; now, when such misalignments in state are detected, the involved API nodes are disconnected rather than the data node detecting it being forced to shut down. (Bug #20430083, Bug #22782511, Bug #23528433, Bug #28505289, Bug #36273474, Bug #36395384, Bug #36838756, Bug #37022773, Bug #37022901, Bug #37023549)

References: See also: Bug #22782511, Bug #23528433, Bug #36273474, Bug #36395384, Bug #36838756.

- **ndbinfo Information Database:** At table create and drop time, access of `ndbinfo` tables such as `operations_per_fragment` and `memory_per_fragment` sometimes examined data which was not valid.

To fix this, during scans of these `ndbinfo` tables, we ignore any fragments from tables in transient states at such times due to being created or dropped. (Bug #37140331)
- Work done previously to support opening `NDB` tables with missing indexes was intended to allow the features of the MySQL server to be used to solve problems in cases where indexes cannot be rebuilt

due to unmet constraints. With missing indexes, some of the SQL handler functionality is unavailable—for example, the use of indexes to select rows for modification efficiently, or to identify duplicates when processing modifications, or to push joins relying on indexes. This could lead to the unplanned shutdown of an NDB Cluster SQL node.

In such cases, the server now simply returns an error. (Bug #37299071)

- Recent refactoring of the transporter layer added the reporting of the presence of socket shutdown errors, but not their nature. This led to confusion in the common case where a socket shutdown is requested, but the socket is already closed by the peer. To avoid such confusion, this logging has been removed. (Bug #37243135)

References: This issue is a regression of: Bug #35750771.

- It was not possible to create an [NDB](#) table with 256 or more [BLOB](#) columns when also specifying a reduced inline size, as in the following SQL statement:

```
CREATE TABLE t1 (  
  pk INT PRIMARY KEY,  
  b1 BLOB COMMENT 'NDB_COLUMN=BLOB_INLINE_SIZE=100',  
  b2 BLOB COMMENT 'NDB_COLUMN=BLOB_INLINE_SIZE=100',  
  .../  
  b256 BLOB COMMENT 'NDB_COLUMN=BLOB_INLINE_SIZE=100'  
) ENGINE=NDBCLUSTER;
```

(Bug #37201818)

- In some cases, the occurrence of node failures during shutdown led to the cluster becoming unrecoverable without manual intervention.

We fix this by modifying global checkpoint ID (GCI) information propagation ([CopyGCI](#) mechanism) to reject propagation of any set of GCI information which does not describe the ability to recover the cluster automatically as part of a system restart. (Bug #37163647)

References: See also: Bug #37162636.

- In some cases, node failures during an otherwise graceful shutdown could lead to a cluster becoming unrecoverable without manual intervention. This fix modifies the generic GCI info propagation mechanism ([CopyGCI](#)) to reject propagating any set of GCI information which does not describe the ability to recover a cluster automatically. (Bug #37162636)
- Improved variable names used in [start\\_resend\(\)](#), and enhanced related debug messages to users and developers with additional information. (Bug #37157987)
- In certain cases, a [COPY\\_FRAGREQ](#) signal did not honor a fragment scan lock. (Bug #37125935)
- In cases where [NDB](#) experienced an API protocol timeout when attempting to close a scan operation, it considered the [DBTC ApiConnectRecord](#) involved to be lost for further use, at least until the API disconnected and API failure handling within [DBTC](#) reclaimed the record.

This has been improved by having the API send a [TCRELEASEREQ](#) signal to [DBTC](#) in such cases, performing API failure handling for a single [ApiConnectRecord](#) within [DBTC](#). (Bug #37023661)

References: See also: Bug #36273474, Bug #36395384, Bug #37022773, Bug #37022901, Bug #37023549.

- For tables using the [NDB](#) storage engine, the column comment option [BLOB\\_INLINE\\_SIZE](#) was silently ignored for [TINYBLOB](#) columns, and (silently) defaulted to the hard-coded 256 byte value regardless of the size provided; this was misleading to users.

To fix this problem, we now specifically disallow [BLOB\\_INLINE\\_SIZE](#) on [TINYBLOB](#) columns altogether, and [NDB](#) now prints a warning saying that the column size is defaulting to 256 bytes. (Bug #36725332)

- Testing revealed that a fix for a previous issue which added a check of the `ApiConnectRecord` failure number against the system's current failure number did not initialize the `ApiConnectRecord` failure number in all cases. (Bug #36155195)

References: This issue is a regression of: Bug #36028828.

- When attempting to alter the default value for a column using `ALTER TABLE INPLACE`, the new value was not persisted in the `NDB` dictionary.

This happened even though changing the default value using `INPLACE` is not supported, and a user wishing to change the default value needs to perform a copying `ALTER TABLE` instead.

Now in such cases we check for this issue, detecting and preventing any attempt at an unsupported `INPLACE` change of the column's default value. (Bug #35106114)

- `ndb_config` did not always handle very long file paths correctly.

Our thanks to Dirkjan Bussink for the contribution. (Bug #116748, Bug #37310680)

- Errors of unknown provenance were logged while assigning node IDs during cluster synchronization, leading to user doubt and concern. Logging of the data node `QMGR` block and the `ndb_mgmd` process relating to node ID allocation issues has therefore been improved, to supply more and better information about what is being reported in such cases. (Bug #116351, Bug #37189356)
- A multi-range scan sometimes lost its fragment lock for the second and subsequent ranges of the scan. (Bug #111932, Bug #35660890)

## Index

### A

API timeouts, 3  
ApiConnectRecord, 3

### B

binary log rotation, 3  
BLOB\_INLINE\_SIZE, 3

### C

CopyGCI, 3

### D

DBSPJ, 3  
DBTC, 3  
DEFAULT, 3

### I

INPLACE, 3

### L

logging, 3

### M

macOS, 3  
Microsoft Windows, 3

### N

NDB Cluster, 3

NDB Cluster APIs, 3  
NDB Disk Data, 3  
NDB Replication, 3  
ndbinfo Information Database, 3  
Ndb\_cluster\_connection, 3  
ndb\_config, 3  
node IDs, 3

## **S**

scans, 3  
STMT\_END\_F, 3  
system restarts, 3

## **T**

TCRELEASEREQ, 3

---