

# Hierarchical metastore event processing (Preview)

Date published: 2026-03-31

Date modified: 2026-03-31

## Legal Notice

© Cludera Inc. 2026. All rights reserved.

The documentation is and contains Cludera proprietary information protected by copyright and other intellectual property rights. No license under copyright or any other intellectual property right is granted herein.

Unless otherwise noted, scripts and sample code are licensed under the Apache License, Version 2.0.

Copyright information for Cludera software may be found within the documentation accompanying each component in a particular release.

Cludera software includes software from various open source or other third party projects, and may be released under the Apache Software License 2.0 ("ASLv2"), the Affero General Public License version 3 (AGPLv3), or other license terms.

Other software included may be released under the terms of alternative open source licenses. Please review the license and notice files accompanying the software for additional licensing information.

Please visit the Cludera software product page for more information on Cludera software. For more information on Cludera support services, please visit either the Support or Sales page. Feel free to contact us directly to discuss your specific needs.

Cludera reserves the right to change any products at any time, and without notice. Cludera assumes no responsibility nor liability arising from the use of products, except as expressly agreed to in writing by Cludera.

Cludera, Cludera Altus, HUE, Impala, Cludera Impala, and other Cludera marks are registered or unregistered trademarks in the United States and other countries. All other trademarks are the property of their respective owners.

Disclaimer: EXCEPT AS EXPRESSLY PROVIDED IN A WRITTEN AGREEMENT WITH CLOUDERA, CLOUDERA DOES NOT MAKE NOR GIVE ANY REPRESENTATION, WARRANTY, NOR COVENANT OF ANY KIND, WHETHER EXPRESS OR IMPLIED, IN CONNECTION WITH CLOUDERA TECHNOLOGY OR RELATED SUPPORT PROVIDED IN CONNECTION THEREWITH. CLOUDERA DOES NOT WARRANT THAT CLOUDERA PRODUCTS NOR SOFTWARE WILL OPERATE UNINTERRUPTED NOR THAT IT WILL BE FREE FROM DEFECTS NOR ERRORS, THAT IT WILL PROTECT YOUR DATA FROM LOSS, CORRUPTION NOR UNAVAILABILITY, NOR THAT IT WILL MEET ALL OF CUSTOMER'S BUSINESS REQUIREMENTS. WITHOUT LIMITING THE FOREGOING, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, CLOUDERA EXPRESSLY DISCLAIMS ANY AND ALL IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, QUALITY, NON-INFRINGEMENT, TITLE, AND FITNESS FOR A PARTICULAR PURPOSE AND ANY REPRESENTATION, WARRANTY, OR COVENANT BASED ON COURSE OF DEALING OR USAGE IN TRADE.

*This document has been released as part of a technical preview for features described herein. Technical preview components are provided as a convenience to our customers for their evaluation and trial usage. These components are provided 'as is' without warranty or support. Further, Cludera assumes no liability for the usage of technical preview components, which should be used by customers at their own risk.*

## Contents

<b>Legal Notice</b>	<b>2</b>
<b>Contents</b>	<b>3</b>
<b>Hierarchical metastore event processing</b>	<b>4</b>
<b>Multi-level event processing</b>	<b>4</b>
<b>Event synchronization and ordering</b>	<b>4</b>
<b>Catalogd configuration properties</b>	<b>5</b>

*This document has been released as part of a technical preview for features described herein. Technical preview components are provided as a convenience to our customers for their evaluation and trial usage. These components are provided 'as is' without warranty or support. Further, Cloudera assumes no liability for the usage of technical preview components, which should be used by customers at their own risk.*

# Hierarchical metastore event processing

Hierarchical metastore event processor uses a multi-layered approach to improve synchronization speed and handle event dependencies.

In Cloudera Data Warehouse Runtime 2025.0.21.0 and higher versions, the metastore event processor supports a multi-layered approach to improve synchronization speed and handle event dependencies.

By default, the metastore event processor is single-threaded. Notification events are processed sequentially with a maximum limit of 1000 events fetched and processed in a single batch. Multiple locks address concurrency issues that can arise when catalog DDL operation processing and metastore event processing try to access or update catalog objects at the same time.

Waiting for a lock or file metadata loading of a table can slow event processing and affect subsequent events, even if those events are not dependent on the previous one. This results in a long synchronization time for Hive Metastore (HMS) events.

## Multi-level event processing

You can convert existing metastore event processing into multi-level event processing by using the `enable_hierarchical_event_processing` flag. This method partitions events based on their dependency, maintains the order of events within that dependency, and processes them independently when possible.

The hierarchical model consists of the following layers:

- Event dispatcher – Dispatches metastore events and maintains a fixed pool of database event executors.
- Database event executor – Manages table event dispatching, processes database events, and manages table event executors.
- Table event executor – Processes events for multiple tables by using table processors.

Multi-level event processing ensures linearizability for a specific table by processing all table-specific events in the order they occurred.

## Event synchronization and ordering

To maintain metadata consistency, certain events act as synchronization barriers and are not processed in parallel:

*This document has been released as part of a technical preview for features described herein. Technical preview components are provided as a convenience to our customers for their evaluation and trial usage. These components are provided 'as is' without warranty or support. Further, Cloudera assumes no liability for the usage of technical preview components, which should be used by customers at their own risk.*

- A database synchronization barrier restricts the table processors of the database from processing events that occur after the database event, until the database event is fully processed.
- A rename synchronization barrier wraps an ALTER TABLE rename event. It synchronizes the source and target table processors to ensure the source processor removes the table before the target processor creates the renamed table.
- A transaction synchronization barrier is used when the system creates pseudo-commit and pseudo-abort transaction events for each table involved, because COMMIT\_TXN and ABORT\_TXN can involve multiple tables. These pseudo events are then processed independently at their respective table processors.

## Catalogd configuration properties

Use the following properties to configure the event executors and processing limits for the metastore event processor.

Property	Description
enable_hierarchical_event_processing	Enables hierarchical event processing on catalogd. This is disabled by default.
num_db_event_executors	Sets the number of database-level event executors.
num_table_event_executors_per_db_event_executor	Sets the number of table-level event executors within a database event executor.
min_event_processor_idle_ms	Sets the minimum time to retain idle DB and table processors when they have no events to process.

*This document has been released as part of a technical preview for features described herein. Technical preview components are provided as a convenience to our customers for their evaluation and trial usage. These components are provided 'as is' without warranty or support. Further, Cloudera assumes no liability for the usage of technical preview components, which should be used by customers at their own risk.*

## CLOUDERA TECHNICAL PREVIEW DOCUMENTATION

max_outstanding_events_on_executors	Sets the limit for the maximum number of outstanding events to process on event executors.
hms_event_polling_interval_s	Defines the interval for polling events. This property now supports double values to allow for millisecond precision.

*This document has been released as part of a technical preview for features described herein. Technical preview components are provided as a convenience to our customers for their evaluation and trial usage. These components are provided 'as is' without warranty or support. Further, Cloudera assumes no liability for the usage of technical preview components, which should be used by customers at their own risk.*