



# Red Hat build of Debezium 2.5.4

## 红帽构建的 Debezium 2.5.4 发行注记

红帽构建的 Debezium 新功能



红帽构建的 Debezium 新功能

## 法律通告

Copyright © 2024 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

XFS<sup>®</sup> is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack<sup>®</sup> Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## 摘要

描述红帽构建的 Debezium 产品，并提供了有关本发行版本中新内容的最新详情。

---

## 目录

<b>第 1 章 红帽构建的 DEBEZIUM</b> .....	<b>3</b>
<b>第 2 章 DEBEZIUM 2.5.4 发行注记</b> .....	<b>4</b>
2.1. DEBEZIUM 数据库连接器	4
2.2. DEBEZIUM 支持的配置	5
2.3. DEBEZIUM 安装选项	5
2.4. DEBEZIUM 2.5.4 功能和改进	5
2.5. 已弃用的功能	33
2.6. 已知问题	33



## 第 1 章 红帽构建的 DEBEZIUM

红帽构建的 Debezium 是一组全面的集成和事件处理技术，用于在混合和多环境中创建、扩展和部署基于容器的集成服务。红帽构建的 Debezium 提供了一个敏捷、分布式和以 API 为中心的解决方案，供组织用来连接和共享数字世界中应用程序和系统之间的数据。

红帽构建的 Debezium 包括以下功能：

- 实时消息传递
- 跨数据中心消息流
- API 连接
- 应用程序连接器
- 企业级集成模式
- API 管理
- 数据转换
- 服务组成和编配

### 其他资源

- [了解企业集成](#)

## 第 2 章 DEBEZIUM 2.5.4 发行注记

Debezium 是一个分布式更改数据捕获平台，用于捕获数据库表中发生的行级更改，然后将对应的更改事件记录传递给 Apache Kafka 主题。应用程序可以读取 [这些更改事件流](#)，并按发生更改事件的顺序访问更改事件。Debezium 基于 Apache Kafka 构建，并在 OpenShift Container Platform 或 Red Hat Enterprise Linux 上部署并与 AMQ Streams 集成。

以下主题提供发行版本详情：

- [第 2.1 节 “Debezium 数据库连接器”](#)
- [第 2.2 节 “Debezium 支持的配置”](#)
- [第 2.3 节 “Debezium 安装选项”](#)
- [第 2.4 节 “Debezium 2.5.4 功能和改进”](#)
- [第 2.4.1 节 “可能会造成问题的更改”](#)
- [第 2.4.2 节 “提升到正式发行\(GA\)的功能”](#)
- [第 2.4.3 节 “正式发行\(GA\)功能”](#)
- [第 2.4.4 节 “技术预览功能”](#)
- [第 2.4.5 节 “开发人员预览功能”](#)
- [第 2.4.6 节 “本发行版本中的其他更新”](#)
- [第 2.5 节 “已弃用的功能”](#)
- [第 2.6 节 “已知问题”](#)

### 2.1. DEBEZIUM 数据库连接器

Debezium 为以下通用数据库提供基于 Kafka Connect 的连接器：

- Db2
- JDBC Sink 连接器
- MongoDB
- MySQL
- Oracle
- PostgreSQL
- SQL Server

#### 2.1.1. 连接器使用备注

- Db2

- Debezium Db2 连接器不包含 Db2 JDBC 驱动程序(**jcc-11.5.0.0.jar**)。有关如何部署必要 JDBC 驱动程序的信息，请参阅 [Db2 连接器部署说明](#)。
- Db2 连接器需要使用抽象语法表示法(ASN)库，这些库作为 Linux 的 Db2 标准部分提供。
- 要使用 ASN 库，您必须有用于 IBM InfoSphere 数据复制(IIDR)的许可证。您不必安装 IIDR 来使用库。
- MySQL
  - 从这个版本开始，在 MariaDB 中使用 MySQL 连接器作为技术预览功能提供。如需更多信息，请参阅 [第 2.4.4.1 节“使用带有 MariaDB 的 MySQL 的 Debezium 连接器”](#)。
- Oracle
  - Debezium Oracle 连接器不包含 Oracle JDBC 驱动程序(**ojdbc8.jar**)。有关如何部署必要 JDBC 驱动程序的详情，请查看 [Oracle 连接器部署说明](#)。
- PostgreSQL
  - 要使用 Debezium PostgreSQL 连接器，您必须使用 **pgoutput** 逻辑解码输出插件，这是 PostgreSQL 版本 10 及更高版本的默认设置。

## 其他资源

- [Debezium 入门](#)
- [Debezium 用户指南](#)

## 2.2. DEBEZIUM 支持的配置

有关 Debezium 支持的配置的详情，包括支持的数据库版本的信息，请参阅 [Debezium 2.5.4 支持的配置页面](#)。

### 2.2.1. AMQ Streams API 版本

Debezium 在 AMQ Streams 2.6 上运行。

AMQ Streams 支持 **v1beta2** API 版本，它更新 AMQ Streams 自定义资源的 schema。旧的 API 版本已弃用。

## 2.3. DEBEZIUM 安装选项

您可以在 OpenShift 或 Red Hat Enterprise Linux 中使用 AMQ Streams 安装 Debezium：

- [在 OpenShift 上安装 Debezium](#)
- [在 RHEL 上安装 Debezium](#)

## 2.4. DEBEZIUM 2.5.4 功能和改进

- [第 2.4.1 节“可能会造成问题的更改”](#)
- [第 2.4.2 节“提升到正式发行\(GA\)的功能”](#)

- [第 2.4.3 节 “正式发行\(GA\)功能”](#)
- [第 2.4.4 节 “技术预览功能”](#)
- [第 2.4.5 节 “开发人员预览功能”](#)
- [第 2.4.6 节 “本发行版本中的其他更新”](#)

### 2.4.1. 可能会造成问题的更改

中断的变化代表连接器行为中的显著区别，需要与早期 Debezium 版本不兼容的配置更改。

Debezium 2.5.4 引入了影响以下组件的更改：

- [第 2.4.1.1 节 “破坏与所有连接器相关的更改”](#)
- [第 2.4.1.2 节 “CloudEvents 转换程序破坏更改”](#)
- [第 2.4.1.3 节 “JDBC sink 连接器破坏更改”](#)
- [第 2.4.1.4 节 “MongoDB 连接器破坏更改”](#)
- [第 2.4.1.5 节 “MySQL 连接器破坏更改”](#)

有关破坏上一个 Debezium 发行版本中更改的详情，请参考 [Debezium 2.3.7 发行注记](#)。

#### 2.4.1.1. 破坏与所有连接器相关的更改

##### ComputePartition SMT removed (DBZ-7141)

**ComputePartition** 单消息转换(SMT)以前用来计算将目标 Kafka 主题分区写入事件记录。这个 SMT 之前已弃用，现已被删除。在 [其位置使用分区 SMT](#)。

#### 2.4.1.2. CloudEvents 转换程序破坏更改

##### metadata.location 属性重命名为 metadata.source (DBZ-7060)

CloudEvents 配置属性 **metadata.location** 被 **metadata.source** 属性替代。如果您使用 CloudEvents converter 来发出符合 CloudEvents 格式的事件，请更新连接器配置中的属性。如需更多信息，请参阅 [Debezium 用户指南中的元数据和一些 CloudEvents 字段的配置](#)。

##### CloudEvent 标头更改(DBZ-7216)

CloudEvent 标头的 schema 名称前缀和字母 casing 与有效负载名称不一致。模式名称现在一致，事件标头和有效负载共享同一命名空间，并遵循与字母划分相同的规则。

#### 2.4.1.3. JDBC sink 连接器破坏更改

##### JDBC sink 值序列化更改(DBZ-7191)

在某些情况下，当 JDBC sink 连接器消耗包含 **null** 值的事件时，当向目标数据库写入事件记录时，它会错误地覆盖默认值，而不是 value **NULL**。

#### 2.4.1.4. MongoDB 连接器破坏更改

##### 新的读取首选项(DBZ-6591)

MongoDB 连接器的早期版本使用硬编码的 **辅助首选**。这个行为与默认和推荐的 MongoDB 配置冲突，其中所有 MongoDB 操作都 **从主** 成员读取。从 Debezium 2.5.4 开始，连接器现在总是遵循连接字符串中设置的读取首选项。

如果您希望保留原始行为，请通过在连接字符串中添加以下选项来自定义连接器配置：

```
&readPreference=secondaryPreferred
```

如果连接器从分片集群读取，`mongodb.connection.mode` 被设置为 `replica_set`，请使用新的 `mongodb.connection.string.shard.params` 属性来指定将读取首选项设置为 **二级** 的连接字符串。如需更多信息，请参阅 [Debezium 用户指南中的连接器文档](#)。

### 分片集群的默认连接模式现在设置为分片 (DBZ-7108)

在分片集群中，MongoDB 连接器的默认连接模式现在 **分片为分片**，而不是 `replica_set`。使用 `replica_set` 模式建立到分片集群的连接会绕过 `mongos` 路由器直接连接到分片。这个配置可能会导致问题，在 MongoDB 不鼓励使用。

升级连接器后，新的默认设置会导致现有偏移无效，并触发初始快照的静默重新执行 (DBZ-7272)。为防止此行为，在重启会触发快照时，添加了检查会失败连接器。

如果您的连接器从分片集群读取，在升级后，检查连接器配置并根据需要调整连接模式。您可以明确设置 `replica_set` 连接模式，或者删除现有偏移。

### 分片部署中的 MongoDB 快照 (DBZ-7139)

在 Debezium 的早期版本中，当您在分片部署中使用 MongoDB 连接器时，您可以指定用于执行集合快照的连接器的分片。不幸的是，虽然某些客户发现此行为非常有用，但并不被支持。这个选项不再可用。

## 2.4.1.5. MySQL 连接器破坏更改

### MySQL BIT 默认长度 (DBZ-7230)

如果没有设置，则 MySQL **BIT 数据类型** 没有隐式长度。如果没有提供任何 **1**，则这个错误是默认的长度。使用模式 `registry` 的用户不应遇到输出模式的任何更改。

### MySQL BIGINT 精度更改 (DBZ-6714)

在早期版本中，当 `bigint.unsigned.handling.mode` 属性在 MySQL 连接器配置中 **准确** 设置时，连接器不会在对 **BIGINT** 数据类型转换列时应用正确的精度。因此，包含不准确精度值的 schema for **BIGINT** 字段。十进制精度现在设置为 **20** 以代表 unsigned **BIGINT** 值。

如果您使用带有之前设置的连接器的 schema `registry`，如果 `registry` 强制执行严格的兼容性规则，这个更改可能会导致不兼容。升级连接器后，根据需要采取措施来确保输出值准确表示，且没有精度丢失，例如调整兼容性设置。

### MySQL 5.7 支持 best-effort (DBZ-6874)

MySQL 5.7 最近进入其生命周期结束，将不再接收更新或安全补丁。根据此公告，从 Debezium 2.5.4 使用 MySQL 5.7 开始不再被视为支持的配置。

## 2.4.2. 提升到正式发行 (GA) 的功能

在 Debezium 2.5.4 发行版本中，以下功能从 Developer Preview 或 技术预览 提升到正式发行 (GA)：

- [第 2.4.2.1 节 “JDBC sink 连接器（从开发者预览中获得）”](#)
- [第 2.4.2.2 节 “用于多副本和分片集群的 MongoDB 增量快照（从技术预览中获得）”](#)

- [第 2.4.2.3 节 “MongoDB 划分的集群改进（从技术预览中获得）”](#)

#### 2.4.2.1. JDBC sink 连接器（从开发者预览中获得）

与其它供应商的实现不同，Debezium JDBC sink 连接器可以集中更改由 Debezium 源连接器发出的事件，而无需首先通过事件扁平化转换过程。Debezium JDBC 连接器可以减少管道的处理空间并简化其配置。连接器还可以利用 Debezium 源连接器功能，如列类型传播。如需更多信息，请参阅 [Debezium 用户指南](#)

#### 2.4.2.2. 用于多副本和分片集群的 MongoDB 增量快照（从技术预览中获得）

现在，使用 MongoDB 连接器执行分片集群的增量快照多副本集群现在被完全支持。如需更多信息，请参阅 [Debezium 用户指南](#)。

#### 2.4.2.3. MongoDB 划分的集群改进（从技术预览中获得）

在以前的版本中，当您在分片集群部署中将 Debezium 连接器用于 MongoDB 时，连接器会直接打开与每个分片副本集的连接。这不是推荐的方法，而 MongoDB 推荐连接器 [使用 mongos 实例（路由器）打开连接](#)。

此发行版本与推荐的策略一致。可能需要调整您的配置，将连接器指向 **mongos** 实例。

为了保持向后兼容，可以使用新的配置属性 `mongodb.connection.mode`。此属性的默认值 `replica_set` 保留当前行为，以便连接器将每个分片作为单个副本集处理。将值设为 `sharded`，以指示连接器内部使用名为 `cluster` 进行人工命名的 ReplicaSet 实例。此配置会导致连接器使用最初提供的连接字符串来处理与 `thshard` 的所有成员的连接。

如需更多信息，请参阅 [MongoDB 分片集群支持](#)。

### 2.4.3. 正式发行(GA)功能

Debezium 2.5.4 为以下连接器提供新功能：

- [第 2.4.3.1 节 “临时阻止快照（所有源连接器）\(DBZ-6566\)”](#)
- [第 2.4.3.3 节 “连接器指标（所有源连接器）DBZ-6603中的自定义标签”](#)
- [第 2.4.3.5 节 “初始快照进度通知（所有连接器）\(DBZ-6416,DBZ-6878\)”](#)
- [第 2.4.3.6 节 “INSERT/DELETE 用于增量快照水位线（所有连接器）\(DBZ-6834\)”](#)
- [第 2.4.3.7 节 “重新选择列后处理器，以填充缺少的大对象\(LOB\) \(DBZ-7602\)”](#)
- [第 2.4.3.2 节 “TimeZone conversion SMT（所有源连接器）\(DBZ-6567\)”](#)
- [第 2.4.3.4 节 “支持 MongoDB 大消息\(DBZ-6726\)”](#)
- [第 2.4.3.8 节 “MongoDB 中的后镜像支持\(DBZ-7299,DBZ-7647\)”](#)

#### 2.4.3.1. 临时阻止快照（所有源连接器）(DBZ-6566)

增量快照是处理各种快照用例的流行方法。但是，在某些情况下，快照读取事件与流事件（创建、更新和删除）进行干预，则增量快照的使用可能无法被消耗的应用程序表示甚至支持。在这些情况下，可以执行临时阻塞快照。

虽然您可以通过向 Debezium 发送信号来触发临时阻止快照和临时阻塞快照，但当连接器处理阻塞快照信号时，它会将一个保存在快照进程期间的流上。因此，快照会在快照读取事件和流创建、更新和删除事件之间保持隔离。换句话说，阻塞快照过程与进程以及传统的快照类似。因此，吞吐量通常高于增量快照。



### 重要

在临时阻塞快照中，事务日志的读取放置在快照期间保留。与传统的快照一样，在使用临时快照模式时，事务日志必须可用。当流恢复时，如果从删除后所需的事务日志，则连接器将引发错误并停止。

启动临时阻塞快照的信号与用于启动临时增量快照的信号类似。以下示例显示了启动带有条件的特定表的阻塞快照的信号有效负载。此信号与增量快照的信号的唯一区别在于，信号的类型被设置为 **BLOCKING** 而不是 **INCREMENTAL**。

```
{
  "type": "execute-snapshot",
  "data": {
    "data-collections": ["public.my_table"],
    "type": "BLOCKING",
    "additional-condition": "last_update_date >= '2023-01-01'"
  }
}
```

如需更多信息，请参阅 Debezium 用户指南中为您的连接器类型 *阻止快照*。

#### 2.4.3.2. TimeZone conversion SMT（所有源连接器）(DBZ-6567)

客户通常会要求发出时区为 **UTC** 以外的值列的功能。在过去的版本中，您可以创建一个 **CustomConverter**，或编写自己的单个消息转换，以指定连接器如何发出临时列。

在本发行版本中，Debebe 提供一个新的时区转换，可让您在连接器发出的事件中修改临时列的时区。您可以根据管道的需求，将列从默认 UTC 时区转换为任何其他时区。要开始进行这个新转换，请在连接器中添加以下基本配置：

```
{
  "transforms": "tz",
  "transforms.tz.type": "io.debezium.transforms.TimezoneConverter",
  "transforms.tz.converted.timezone": "America/New_York"
}
```

添加上述配置后，当连接器发出事件时，UTC 中的所有时列都会转换为 'America/New\_York' 时区。

您不仅可以更改 temporal 字段的时区，您也可以使用 **include.fields** 属性修改其他字段的时区设置，如下例所示：

```
{
  "transforms": "tz",
  "transforms.tz.type": "io.debezium.transforms.TimezoneConverter",
  "transforms.tz.converted.timezone": "America/New_York",
  "transforms.tz.include.fields": "source:customers:created_at,customers:updated_at"
}
```

在前面的示例中，`include.fields` 中的第一个值会转换源表的 `created_at` 字段的时区，名称为 `customer`。下一个值将带有名称 `customer` 的主题的 `updated_at` 字段转换。

或者，如果要将转换应用到所有但字段子集，您可以设置 `exclude.fields` 属性，如下例所示：

```
{
  "transforms": "tz",
  "transforms.tz.type": "io.debezium.transforms.TimezoneConverter",
  "transforms.tz.converted.timezone": "America/New_York",
  "transforms.tz.exclude.fields": "source:customers:updated_at"
}
```

在前面的示例中，转换将所有临时字段转换为 `America/New_York` 时区，但源表名称是客户，该字段为 `updated_at`。如需更多信息，请参阅 [Debezium 用户指南中的 Debezium 事件记录中的转换时区值](#)。

#### 2.4.3.3. 连接器指标（所有源连接器） [DBZ-6603](#)中的自定义标签

现在，您可以指定自定义标签来附加到连接器 MBean 对象名称的名称中。自定义标签允许您配置更可靠的连接器监控。

Debezium 连接器公开指标，提供有关快照、流和架构历史记录进程行为的数据。这些指标为每个连接器可用，并通过连接器的 MBean 名称公开。

默认情况下，当您部署正确配置的连接器时，Debezium 会为连接器生成一个唯一的 MBean。您可以配置可观察堆栈来监控与该 MBean 关联的指标。如果您稍后更改连接器配置，这些更改可以修改原始的 MBean 名称。如果 MBean 名称发生变化，连接器实例和 MBean 中断之间的链接，而监控工具将无法再检索连接器的指标数据，除非您重新配置可观察堆栈以使用新的 MBean 名称。

为了避免 MBean 名称更改的结果的潜在 churn，您可以通过将 `custom.metric.tags` 属性添加到连接器配置来将连接器配置为使用自定义指标标签。`custom.metric.tags` 属性接受键值对。每个键代表 MBean 对象名称的标签，对应的值代表该标签的值。例如：`k1=v1,k2=v2`。

为连接器配置 `custom.metric.tags` 属性后，将 `observability` 堆栈配置为检索与指定标签关联的指标。然后，可观察性堆栈使用指定的标签，而不是 mutable MBean 名称来唯一标识连接器。之后，如果 Debezium 重新定义了它构造 MBean 名称的方式，或者连接器配置更改中的 `topic.prefix`，指标集合不会中断，因为指标提取任务使用指定的标签模式来识别连接器。

使用自定义标签的进一步优点是，您可以使用标签来反映数据管道的架构，以便以适合您的操作需求的方式组织指标。例如，您可以使用声明连接器活动类型、应用程序上下文或数据源的值来指定标签，如

`db1-streaming-for-application-abc`。如果您指定多个键值对，则所有指定的对都会附加到连接器的 MBean 名称中。

以下示例演示了标签如何修改默认的 MBean 名称。

#### 例 2.1. 自定义标签如何修改连接器 MBean 名称

默认情况下，Oracle 连接器使用以下 MBean 名称进行流传输指标：

```
debezium.oracle:type=connector-metrics,context=streaming,server=<topic.prefix>
```

如果将 `custom.metric.tags` 的值设置为 `database=salesdb-streaming,table=inventory`，Debezium 会生成以下自定义 MBean 名称：

```
debezium.oracle:type=connector-metrics,context=streaming,server=
<topic.prefix>,database=salesdb-streaming,table=inventory
```

#### 2.4.3.4. 支持 MongoDB 大消息(DBZ-6726)

Debezium MongoDB 连接器现在支持 MongoDB 6 的大型文档处理。这个更改可防止连接器在试图捕获超过 16MB 的文档时返回 `BSONObjectTooLarge` 异常。要将连接器配置为利用此功能，请设置 `oversize.handling.mode` 选项。



#### 注意

支持 MongoDB 大消息依赖于 MongoDB 数据库的底层功能。因为数据库对处理大型文档施加限制，因此如果连接器试图捕获不遵循大型文档的规则，则连接器可能仍然会返回异常。如需更多信息，请参阅 [MongoDB 文档](#)。

#### 2.4.3.5. 初始快照进度通知（所有连接器）(DBZ-6416,DBZ-6878)

Debezium 通知子系统提供了一种简单的方法，可以将第三方工具和应用程序与 Debezium 集成，以深入了解传统的 JMX 方法以外的持续更改数据捕获过程。在早期版本中，Debezium 通知子系统仅提供有关初始快照的基本信息，如快照启动时、每个表启动和结束快照时。notification 子系统现在包括告知您初始快照持续状态的功能。

初始快照通知使用 Initial Snapshot 的 `aggregatetType` 发送，包含公开快照当前状态的 `type` 字段。可能的值包括：

STARTED、ABORTED、PAUSED、RESUMED、IN\_PROGRESS、TABLE\_SCAN\_COMPLETED 和 COMPLETED。

**DBZ-6878** 扩展基本功能，以提供有关快照的增强详情。例如，IN\_PROGRESS 通知现在提供有关捕获的表以及当前正在进行中的表的更多详情，如下例所示：

```
{
  "id":"6d82a3ec-ba86-4b36-9168-7423b0dd5c1d",
  "aggregate_type":"Initial Snapshot",
  "type":"IN_PROGRESS",
  "additional_data":{
    "connector_name":"my-connector",
    "data_collections":"table1, table2",
    "current_collection_in_progress":"table1"
  },
  "timestamp": "1695817046353"
}
```

注意

上例中的一些字段（如 data\_collection）目前不适用于 MongoDB 快照，且仅适用于基于 SQL 的关联连接器。如需更多信息，请参阅 [Debezium 用户指南中的有关初始快照状态的 Debezium 通知](#)。

#### 2.4.3.6. INSERT/DELETE 用于增量快照水位线（所有连接器）(DBZ-6834)

此发行版本引入了属性 `incremental.snapshot.watermarking.strategy`，它允许您指定在增量快照期间使用的水位线策略。过去的版本使用 `insert_insert` 方法，Debezium 在各个块的快照期间在信号数据集中创建了两个条目。第一个条目指出了快照窗口打开的，下一条目会标记其 `closure`。

新的 `insert_delete` 选项会在快照窗口开始时将单个条目写入每个块的信号数据收集。快照完成后，条目会被删除，且不会添加对应的条目来表示快照窗口的冲突。此方法提供对信号数据收集的更有效的管理。

#### 2.4.3.7. 重新选择列后处理器，以填充缺少的大对象(LOB) (DBZ-7602)

由于某些源数据库功能的方式，当 Debezium 连接器发出更改事件时，事件可能会排除存储大量数据的列类型的值。例如，PostgreSQL 中的 TOAST 列的值、Oracle 中的 LOB 列或 Oracle Exadata 中的扩展 String 列可能不包括在更改事件中。

Debezium 2.5.4 在 *处理器后* 引入了 `Reselect` 列，它提供了一种机制来重新选择连接器从数据库表中捕获的一个或多个列，并获取这些列的当前状态。您可以将后处理器配置为重新选择以下列类型：

- null 列
- 包含 `unavailable.value.placeholder` sentinel 值的列。

配置 `PostProcessor` 与配置自定义转换器或单一消息转换类似，但它适用于 mutable 有效负载的 `Struct` 而不是 `SourceRecord`。如需更多信息，请参阅 Debezium 用户指南中的 [使用-the-reselect-columns-post-processor-to-add-source-fields-to-change-event-records](#)。

#### 2.4.3.8. MongoDB 中的后镜像支持([DBZ-7299](#),[DBZ-7647](#))

当 MongoDB 的 Debezium 连接器发出更改事件时，您可以配置连接器，以便事件有效负载包括更新中更改的完整源文档。在过去的版本中，连接器不会在事件有效负载中一致地流传输完整的文档。

要显式控制连接器如何在更改流中查找完整的文档，请设置配置选项 `capture.mode.full.update.type`。此选项的默认值会查找完整的文档，这需要数据库执行单独的查找来获取完整文档。如果您将连接器与 MongoDB 6 或更高版本搭配使用，您可以将连接器配置为使用 `post_image` 来依赖 MongoDB 的 `post-image` 支持。

如需更多信息，请参阅 Debezium 用户指南中的 MongoDB 连接器部分中的 [capture.mode.full.update.type](#)。

#### 2.4.4. 技术预览功能

Debezium 2.5.4 中提供了以下技术预览功能：

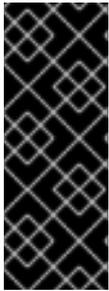
- [第 2.4.4.1 节 “使用带有 MariaDB 的 MySQL 的 Debezium 连接器”](#)
- [第 2.4.4.2 节 “从 PostgreSQL 16 待机流\(DBZ-7181\)”](#)

以前提供技术预览功能

之前版本中引入的以下功能仍为技术预览：

- [第 2.4.4.3 节 “CloudEvents converter”](#)

- [第 2.4.4.4 节 “自定义转换器”](#)
- [第 2.4.4.5 节 “使用带有 Oracle 连接器的 BLOB、CLOB 和 NCLOB 数据类型”](#)
- [第 2.4.4.6 节 “SQL 连接器的并行初始快照”](#)



#### 重要

技术预览功能不受红帽产品服务等级协议 (SLA) 支持，且功能可能并不完整。红帽不推荐在生产环境中实施任何技术预览功能。技术预览功能为用户提供了一个对最新的产品创新的试用机会，以使用户可以对其进行测试并提供反馈。如需有关支持范围的更多信息，请参阅 [技术预览功能支持范围](#)。

#### 2.4.4.1. 使用带有 MariaDB 的 MySQL 的 Debezium 连接器

从 Debezium 2.5.4 开始，您可以针对单个 MariaDB 数据库部署运行 MySQL 连接器。MySQL 和 MariaDB 都可以使用 [全局事务标识符\(GTID\)](#) 进行复制。GTID 有助于在集群中唯一标识事务。

要使 MySQL 连接器与 MariaDB GTID 一起使用，您必须将 supplemental 配置应用到连接器。如需更多信息，请参阅 [Debezium 用户指南](#)

#### 2.4.4.2. 从 PostgreSQL 16 待机流(DBZ-7181)

使用 PostgreSQL 16，您现在可以在独立实例上定义复制插槽。根据对数据库的这一更改，在 Debezium 中，您现在可以连接到备用 PostgreSQL 16 服务器，并从其中流传输更改。通过从副本而不是生产系统执行数据捕获，您可以更好地分发服务器负载，特别是在非常活跃的数据库中。

以前提供技术预览功能

之前版本中引入的以下功能仍为技术预览：

#### 2.4.4.3. CloudEvents converter

CloudEvents converter 会发出符合 CloudEvents 规格的更改事件记录。CloudEvents 更改事件 envelope 可以是 JSON 或 Avro，每个 envelope 类型都支持 JSON 或 Avro 作为数据格式。如需更多信息，请参阅 [CloudEvents 转换程序](#)。

#### 2.4.4.4. 自定义转换器

如果默认数据类型转换无法满足您的需要，您可以创建自定义转换器以用于连接器。如需更多信息，[请参阅自定义开发的转换器](#)。

#### 2.4.4.5. 使用带有 Oracle 连接器的 BLOB、CLOB 和 NCLOB 数据类型

如需更多信息，请参阅 Debezium 用户指南中的 [Oracle 二进制字符 LOB 类型](#)。

#### 2.4.4.6. SQL 连接器的并行初始快照

使用多个并行线程执行初始快照的功能作为 MySQL 以外的所有 Debezium SQL 连接器的技术预览功能提供。[Debezium MySQL 连接器的并行初始快照](#) 作为开发者预览功能提供。

要将连接器配置为使用并行初始快照，请将连接器配置中的 `snapshot.max.threads` 属性设置为大于 1 的值。

如需更多信息，请参阅 Debezium 用户指南文档中的 `snapshot.max.threads` 部分用于您的连接器。

#### 2.4.5. 开发人员预览功能

ProductName} 2.5.4 包括以下开发人员预览功能：name: value

- [第 2.4.5.1 节 “MySQL 并行模式快照\(DBZ-6472\)”](#)
- [第 2.4.5.2 节 “MySQL 并行初始快照”](#)
- [第 2.4.5.3 节 “从逻辑待机中获取更改”](#)
- [第 2.4.5.4 节 “PostgreSQL 流的一次发送”](#)



## 重要

红帽以任何方式支持开发人员预览功能，且功能不完整或生产就绪。对于生产环境或关键业务工作负载，不要使用开发人员预览软件。开发人员预览软件可提前访问即将发布的产品软件。客户可以使用此软件测试功能并在开发过程中提供反馈。此软件可能没有任何文档，可以随时更改或删除，并收到有限的测试。红帽可能会提供在没有关联的 SLA 的情况下提交开发人员预览软件反馈的方法。

有关 Red Hat Developer Preview 软件支持范围的更多信息，请参阅 [开发人员预览支持范围](#)。

### 2.4.5.1. MySQL 并行模式快照(DBZ-6472)

要提高 MySQL 连接器的快照性能，实现并行快照更改事件和表的模式事件

这可减少捕获数据库中多个表的架构时的快照时间。

配置此属性后，连接器使用收集指标时基于这些键值对的 `name` 模式。这会建立一个稳定的名称，以便连接器始终收集相同的指标集合，无论您是否修改 `topic.prefix`，还是 MBean 名称更改。自定义每个连接器实例的 MBean 对象名称。通过添加这些自定义指标标签，您可以确保每个连接器都有唯一的 MBean 名称，防止多个实例间的冲突。

### 2.4.5.2. MySQL 并行初始快照

MySQL 的 Debezium 初始快照始终是单线程。这种限制主要源自确保多个交易间的数据一致性的复杂性。

在这个发行版本中，您可以将 MySQL 连接器配置为使用多个线程并行执行表级快照。

为了利用这个新功能，请将 `snapshot.max.threads` 属性添加到连接器配置中，并将属性设置为大于 1 的值。

使用并行快照的配置示例

```
snapshot.max.threads=4
```

在上例中，如果连接器需要快照超过 4 个表，则最多 4 个表可以并行快照。当一个线程完成处理表时，它将在队列中找到下一个表并开始执行快照。此过程将继续，直到连接器完成所有指定表的快照。

如需更多信息，请参阅 Debezium 用户指南中的 [snapshot.max.threads](#)。

#### 2.4.5.3. 从逻辑待机中获取更改

当 Oracle 的 Debezium 连接器连接到生产环境或主数据库时，它使用内部清除表来管理 Oracle Log Writer Buffer (LGWR) 进程的清除周期。flush 过程要求连接器访问数据库的用户帐户具有创建和写入此 flush 表的权限。但是，逻辑代表数据库通常具有限制性的数据操作规则，甚至是只读的。因此，对数据库的写入可能并不可行。

为了支持 Oracle 只读逻辑独立数据库，Debezium 引入了一个属性，用于禁用 flush 表的创建和管理。您可以将此功能用于 Oracle Standalone 和 Oracle RAC 安装。

要启用 Oracle 连接器使用只读逻辑，请添加以下连接器选项：

```
internal.log.mining.read.only=true
```

如需更多信息，请参阅 Debezium 用户指南中的 [Oracle 连接器文档](#)。

#### 2.4.5.4. PostgreSQL 流的一次发送

在本发行版本中，Debebe 支持 PostgreSQL 连接器的只一次语义。PostgreSQL 的一次交付仅适用于 streaming 阶段；完全发送不适用于快照。

Debezium 旨在通过确保连接器捕获受监控源中的所有更改事件为目标，提供 at-least-once 交付。在 [KIP-618](#) 中，Apache Kafka 社区建议一种解决生成者重试消息时发生的问题的解决方案。源连接器有时会将事件批处理重新发送到 Kafka 代理，即使之前提交了批处理的代理也是如此。这种情况可能会导致重复的事件发送到消费者（接收器连接器），这可能会导致无法轻松处理重复的用户出现问题。

不需要更改连接器才能启用精确发送。但是，必须在 Kafka Connect worker 配置中配置准确的发送。有关设置所需的 Kafka Connect 配置属性的详情，请参考 [KIP-618](#)。



## 注意

要在 Kafka worker 配置中将 `exactly.once.support` 设置为必需，Kafka Connect 集群中的所有连接器都必须支持完全发送，如果您试图在集群中设置这个选项，则 worker 不支持完全发送的连接器，不支持此功能在启动时验证的连接器。

### 2.4.6. 本发行版本中的其他更新

此 Debezium 2.5.4 发行版本提供多个其他功能更新和修复，包括以下列表中的项目：

- **DBZ-3605** 将对 XML\_TYPE 列类型的支持添加到用于 Oracle 的 Debezium 连接器 (LogMiner)
- **DBZ-3642** Debezium outbox 不用于 CloudEventsConverter
- **DBZ-3925** MySQL 连接器无法解析声明 FLUSH FIREWALL\_RULES
- **DBZ-4321** Explore BLOB 通过 re-selection 支持
- **DBZ-5350** Oracle RAC 会抛出 ORA-00310: 归档日志序列
- **DBZ-5359** 添加 API 端点来公开正在运行的连接器指标
- 如果过滤了 LAST 记录，则 **DBZ-5464** Snapshot 结果不会被保存
- **DBZ-5518** 定义和文档模式历史记录主题消息 schema
- **DBZ-5656** Oracle 缺少 CDC 数据
- **DBZ-5676** Align query.fetch.size across connector

- [DBZ-5750](#) Missing Oracle CDC 记录
- [DBZ-6182](#) Don 在监控单个数据库/集合时不需要集群范围的特权
- 如果数据库无法访问，则 [DBZ-6236](#) PostgreSQL 连接器将无法正确重启
- [DBZ-6240](#) 提供的 DDL 类型 schema 事件过滤
- [DBZ-6317](#) JDBC Sink Connector - 支持批处理操作
- [DBZ-6416](#) Notify 关于初始快照进度
- [DBZ-6417](#) make signal 操作可扩展
- MongoDB 连接器中的 [DBZ-6434](#) NullPointerException
- [DBZ-6458](#) 只发布 deltas 而不是完整快照来减小同步事件消息的大小
- [DBZ-6468](#) Set Readpreference tags in the MongoDB client
- [DBZ-6472](#) MySqlSnapshotChangeEventSource parallel execute createSchemaEventsForTables
- [DBZ-6481](#) PostgreSQL 增量快照在主键中带有 enum 类型的表上失败
- [DBZ-6484](#) JDBC 模式历史记录 : 当表名称作为 dbName.tableName 传递时，连接器不会启动
- [DBZ-6518](#) 更新 MongoDB 增量快照，以允许多个线程读取块

- **DBZ-6521** MongoDB 更改流管道不遵循硬编码的 `readPreference=secondaryPreferred`
- **DBZ-6566** 支持阻止临时快照
- **DBZ-6567** SMT 用于处理时区转换
- 从 `all_tables` 中过滤时，在现有发布更新时 **DBZ-6577** Explain 失败
- **DBZ-6578** Debezium 应该遵循连接字符串中的读取首选项
- **DBZ-6595** Use source field in topic in `table.format.name`
- **DBZ-6602** 支持从标头获取主密钥
- **DBZ-6603** 支持连接器指标中的自定义标签
- **DBZ-6615** Max transaction duration for Oracle connector
- **DBZ-6617** 为事件添加分片字段
- **DBZ-6635** Debezium `heartbeat.action.query` 在写入 WAL 前不会启动
- **DBZ-6636** 只包括 JDBC sink 连接器中的某些列
- **DBZ-6637** Received an unexpected message type, 没有"after" Debezium 块
- **DBZ-6641** Schema 名称使用自定义主题命名策略

- [DBZ-6653](#) 为初始化过程添加可配置的超时
- 对于 JSON 消息 deserialization, [DBZ-6654](#) CloudEvents converter 无法正常工作
- [DBZ-6658](#) 在分区查询完成后发送 tombstone 事件
- 当没有 table.include.list 的 signal.data.collection 时, [DBZ-6669](#) 快照不会捕获数据
- 如果 LogMiner 查询没有返回任何行, 则 [DBZ-6679](#) Log Mining Processor advances SCN 不正确
- [DBZ-6682](#) Wrong behavior of quote.identifiers in JdbcSinkConnector
- [DBZ-6684](#) Propagate source 列名称并允许 sink 使用它
- 使用单个领导任务重新平衡后 [DBZ-6685](#) 分区重复
- [DBZ-6686](#) JDBC sink 连接器在加载包含 Kafka 中的 Struct 类型字段的扁平数据时失败
- [DBZ-6687](#) SQL syntaxErrorException 使用 Debezium JDBC sink 连接器
- [DBZ-6689](#) 为 PostgreSQL 连接器配置 Kafka 频道消费者组 ID
- 在对 Atlas 执行时, [DBZ-6700](#) Missing operationTime 字段
- [DBZ-6712](#) schema.history.internal.store.only.captured.databases.ddl 标志在快照模式到历史记录主题时不考虑
- [DBZ-6714](#) 以精确模式为 MySQL unsigned BIGINTs 指定模式中的十进制精度

- **DBZ-6720** Toasted UUID 数组没有被正确处理
- **DBZ-6726** Utilize \$changeStreamSplitLargeEvent 以处理带有 post- 和 pre-images 的大型更改事件
- **DBZ-6727** 支持 MySQL 连接器中的替代 JDBC 驱动程序
- **DBZ-6731** 阻塞快照必须从信号进行快照配置
- **DBZ-6741** 支持 MongoDB 连接器上的自定义身份验证
- **DBZ-6742** 使用 JMX 通知 userData 的 JSON 格式
- **DBZ-6745** SingleProcessor 删除冗余过滤器逻辑
- **DBZ-6748** Debezium 应该将 \_bin collate varchar 列转换为字符串 not byte 数组
- **DBZ-6775** Table 模式应该单独为每个分片更新
- **DBZ-6778** Refactor ElapsedTimeStrategy
- 在流中添加 XMLTYPE 列时, **DBZ-6782** Oracle XML 列类型不会被正确解决
- **DBZ-6786** 在 binlog 压缩的情况下使用自定义 RowDeserializers
- **DBZ-6787** Incremental snapshot data-collections not deduplicated
- **DBZ-6793** 为通知添加时间戳

- **DBZ-6801** JDBC sink 不支持 SQL Server 身份插入
- **DBZ-6811** SQL Server 连接器在数据库没有更改时发送心跳
- **DBZ-6814** Make finished partition deletion delay configurable
- **DBZ-6820** Fix bug with getsnapshottingtask
- **DBZ-6828** Ad hoc blocking snapshot trigger 会发出所有表的 schema 更改
- **DBZ-6831** Error with propagation source column name
- **DBZ-6834** 为增量快照水位线提供 INSERT/DELETE 语义
- **DBZ-6843** when using skip.messages.without.change=true 为每个记录报告 WARN 日志消息
- **DBZ-6844** Support truncating large column
- **DBZ-6853** Kafka 偏移存储失败，并显示 NPE
- **DBZ-6855** JDBC Offset 存储 - 表名称的配置无法正常工作
- **DBZ-6857** JDBC sink 插入 Oracle 目标数据库因为分号而失败
- **DBZ-6862** Tombstone 事件会导致 JDBC 连接器上的 NPE
- **DBZ-6864** MySQL 连接器不过滤 AWS RDS 内部事件

- [DBZ-6865](#) Avoid NPE 在 ExecuteSnapshot 中执行已到达的方法
- [DBZ-6869](#) On initial Oracle connector start, if a transaction 的 start\_scn 为 0, 日志从最旧的 scn 开始
- [DBZ-6870](#) 确保连接器可以更好地处理重新平衡事件
- [DBZ-6871](#) ChangeStream 聚合管道在应该排除的大型文档上失败
- [DBZ-6878](#) e 通知信息和初始快照的更多通知
- [DBZ-6893](#) 将所有示例从 mongodb.hosts 迁移到 mongodb.connection.string
- [DBZ-6899](#) Refactor Oracle streaming metrics beans
- [DBZ-6907](#) Setting none to delete.handle.mode
- [DBZ-6935](#) fix logger named
- [DBZ-6940](#) Timezone 转换无法正常工作
- [DBZ-6941](#) MySQL Kafka 信号文档不正确
- [DBZ-6945](#) Drop 事件具有错误的表更改信息
- 在 additional-condition 中使用 OR 条件时 [DBZ-6956](#) Infinite 循环
- [DBZ-6958](#) Wrong case-behavior for non-Avro 列名称在 sink 连接器中

- **DBZ-6966** Filter out 指定 DDL 事件逻辑已被恢复
- **DBZ-6967** Handle 正确对 JDBC sink 到 PostgreSQL 的 bytea 字段
- **DBZ-6968** 文档用于 cursor.oversize.skip.threshold 缺少单位
- **DBZ-6970** Debezium JDBC sink 进程 truncate 事件失败
- **DBZ-6971** DDL parser 不支持 NOCOPY 关键字
- **DBZ-6973** 添加 MongoDB 连接器支持 filtering.match.mode=regex|literal 属性
- **DBZ-6974** Decrease 时间用于处理重新平衡事件
- **DBZ-6975** Single quote 复制包括 N (CHAR/VARCHAR)列的转义引号
- **DBZ-6982** provide 配置选项，用于从 CloudEvent 中排除扩展属性
- **DBZ-6983** 添加在推断 JSON 模式时 sanitize 字段名称的功能
- **DBZ-6990** Debezium JDBC sink 应该抛出不支持 schema 更改主题异常
- **DBZ-7015** 启用复制插槽提前检查
- **DBZ-7016** 将配置选项添加到 CloudEventsConverter，以从标头检索 id 和 type
- **DBZ-7030** DDL 语句无法解析

- [DBZ-7035](#) 阻塞临时快照并不真正阻止 MySQL
- [DBZ-7037](#) Fake ROTATE 事件在连接重启时清理元数据
- [DBZ-7043](#) 在从捕获实例读取完成后提供通知
- [DBZ-7050](#) 支持快照，并自动重试
- [DBZ-7058](#) 字段排除不适用于已删除字段的事件
- [DBZ-7060](#) Rename metadata.location to metadata.source
- [DBZ-7065](#) JDBC sink 连接器无法使用 CloudEvent
- [DBZ-7066](#), DefaultDeleteHandlingStrategy 无法将 rewrite "\_\_deleted" 字段添加到非结构值
- [DBZ-7067](#) Improve logging at DEBUG level for Commit 事件
- 在处理过程中发生错误时 [DBZ-7069](#) JDBC 连接泄漏
- [DBZ-7071](#) 使用可配置的选项替换 SYS/SYSTEM 用户的模式跟踪限制
- [DBZ-7083](#) 为 MariaDB 和 MySQL 实施策略模式
- [DBZ-7093](#) Check schema length when create value to find missed DDL by SQL\_BIN\_LOG=OFF
- [DBZ-7095](#) MySQL 解析器不符合算术操作优先级

- **DBZ-7105** When `RelationalBaseSourceConnector#validateConnection` 被调用，并带有无效的配置 [`inside Connector#validate ()`] 可能会导致异常
- **DBZ-7108** 将默认连接模式切换到分片集群共享
- **DBZ-7119** Debezium 在解析 MySQL DDL 语句时崩溃 (特定的 INSERT)
- **DBZ-7132** 在快照后无法对 MySQL 数据库进行身份验证
- **DBZ-7139** MongoDB 数据收集过滤器需要在阻塞/初始快照执行时设置副本集规格
- **DBZ-7140** Debezium DDL parser 在解析 MySQL DDL 语句时崩溃 (特定的 UNION)
- **DBZ-7142** Outbox 事件路由器 SMT 在 `field. additional.placement` 值中有一个空格时抛出 `NullPointerException`
- **DBZ-7146** Inactivity pause in MongoDB 连接器应该可以被配置
- **DBZ-7152** Debezium DDL parser 在解析 MySQL DDL 语句 (特定的 UPDATE) 时崩溃
- **DBZ-7157** JsonSerialisation 无法从带有复合分片密钥的分片集合处理更改
- 如果值不是 `CloudEvent`，则 **DBZ-7159** 在 `deserialization` 过程中快速失败
- **DBZ-7162** 添加最后事件进程时间、事件数量、心跳事件指标数量到 MongoDB 连接器
- **DBZ-7164** 支持 kafka 信号主题的快照请求的持久性历史记录
- **DBZ-7177** Change metrics endpoint of Connect REST Extensions 以直接使用 `MBeanServerV`，而不是对 `Jolokia` 端点的 HTTP 调用

- **DBZ-7178 Metrics** 端点必须处理带有多个任务的连接器(SQL Server)
- **DBZ-7179** Fix DebeziumMySqlConnectorResource 不使用新的 MySQL 适配器结构来支持不同的 MySQL 类型
- **DBZ-7181** 支持逻辑解码来自 Postgres 16 支持
- **DBZ-7183** 支持 MySQL 8 高分辨率复制时间戳, 来自 GTID 事件
- 在读取 MongoDB 更改流事件时 **DBZ-7184** Use buffer queue
- **DBZ-7186** Cleanup event processing loop 在 MongoDB 连接器的流传输事件源中
- **DBZ-7189** Parsing MySQL index for JSON 字段会失败, 当将类型 double 和 float 一起使用
- **DBZ-7191** JDBC 连接器在可选字段中的值是 NULL 时错误地使用默认值
- **DBZ-7193** Unchanged toasted array 列被替换为 unavailable.value.placeholder, 即使配置了 REPLICA IDENTITY FULL
- **DBZ-7194** 启用针对 LogMiner 的 Oracle 23c 流更改的功能
- **DBZ-7196** Add modify range\_partitions 用于解析 PL/SQL 中的修改\_table\_partition 规则
- **DBZ-7197** Add 的功能, 以避免在缺少其他字段时抛出异常
- **DBZ-7206** MongoDB streaming pauses 仅在没有事件时阻止快照

- **DBZ-7208** Handle Drop Tablespace in PL/SQL
- **DBZ-7213** DDL GRANT 语句无法解析
- **DBZ-7216** 当元数据位于标头中时，CE data 字段中的结构名称不正确
- **DBZ-7217** 添加追踪日志来跟踪 Debezium JDBC 连接器的执行时间
- **DBZ-7218** Validate 和 clarify 对 Oracle 的多个归档日志目的地要求
- **DBZ-7230** MySQL BIT Type 应具有默认长度 1
- **DBZ-7235** 将配置选项添加到 CloudEventsConverter 以自定义模式类型名称
- **DBZ-7236** Oracle Abandoned 事务实现 bug 会导致内存错误
- **DBZ-7237** Oracle LOB，如果 lob.enabled=false 正确忽略
- **DBZ-7242** Add grammar Oracle truncate cluster
- 在更改列的类型时，**DBZ-7251** Length 值不会被删除
- **DBZ-7252** MongoDB 集合快照通知包含不正确的偏移
- **DBZ-7259** Debezium DDL parser 崩溃 when 解析 MySQL DDL 语句（带有 UNION 的子查询）
- **DBZ-7266** Oracle DDL 解析错误(PARTITION REFERENCE)

- [DBZ-7268](#) 为 Alter Table Memoptimize 添加 PL/SQL Parser
- 在带有多个快照线程的快照期间 [DBZ-7264](#) 错误不会正确中止快照
- [DBZ-7271](#) MySQL RDS UPDATE 查询没有忽略
- [DBZ-7272](#) Guard 针对由默认连接模式交换机导致的隐式偏移无效
- [DBZ-7275](#) Leaking JDBC 连接
- [DBZ-7277](#) Debezium MySQL 无法解析某些授予特权
- [DBZ-7279](#) 为 Create Table Memoptimize 添加 PL/SQL Parser
- [DBZ-7283](#) 支持创建 EDITIONABLE 或 NONEDITIONABLE 软件包
- [DBZ-7299](#) MongoDB 连接器不使用 post-images
- 当快照模式设置为 never 时, [DBZ-7311](#) PostgreSQL 临时阻止快照会失败
- [DBZ-7312](#) Ad-hoc Block dies with "invalid snapshot identifier", 在连接器创建后立即使用 "invalid 快照标识符"
- [DBZ-7315](#) 指定在元素之间带有空格的表列表会导致 LogMiner 查询丢失匹配项
- [DBZ-7316](#) Debezium heartbeat.action.query 在写入 WAL: part 2 前不会启动
- [DBZ-7347](#) Initial snapshot 通知应使用完整的标识符

- **DBZ-7358** Re-select 列应使用表的主列，而不是事件的密钥
- **SQL Server Table skip** 块 36 记录上的 **DBZ-7359** 全增量快照
- **DBZ-7360** Debezium 在表分割操作后失败
- **DBZ-7374** SQL Server 在 varchar, nvarchar, char 列的数据库 schema 中错误的默认值
- **DBZ-7379** 支持连接器范围为 MongoDB 的 trustore/keystore
- **DBZ-7420** ParsingException (MariaDB Only): changeSpec drop external key with 'tablename.' prefix
- **DBZ-7421** Poor performance with incremental snapshot with long tables list
- **DBZ-7425** Oracle 快照错误地使用 LogMiner Offset Loader
- 在不使用 event-key 源时，**DBZ-7429** Reselect 列应该从 Struct 后 source key 值
- **DBZ-7431** 允许通过配置自定义 C3P0ConnectionProvider
- **DBZ-7436** Stopwatch 在调用 toString 时抛出 NPE，且没有统计信息
- **DBZ-7437** ReselectColumnsPostProcessor 过滤器不使用 exclude predicate
- **DBZ-7441** 临时快照不会在应用程序开始前通过 File 频道信号触发
- **DBZ-7445** LogMiner 批处理大小不会自动增加

- **DBZ-7456** Oracle 连接器不会忽略排除的 CLOB/BLOB 列的 reselection
- **DBZ-7460** table.include.list 的预期值模式与文档不匹配
- **DBZ-7467** Signals 操作没有为 SQL Server 加载
- **DBZ-7468** MySQL 连接器无法使用 SYSTEM VERSIONING PARTITION BY SYSTEM\_TIME 解析表
- **DBZ-7479** 在 Oracle ReselectColumnsProcessor 中准备泄漏
- **DBZ-7488** Poor snapshot performance with new reselect post processor
- **DBZ-7489** Debezium Oracle Connector ParsingException on XMLTYPE with lob.enabled=true
- **DBZ-7500** Fix MySQL 8 事件时间戳解析逻辑错误, 如果非GTID 事件错误地出现回退到秒
- **DBZ-7562** Numeric 默认值十进制扩展不匹配
- **DBZ-7567** Fix null event timestamp from FORMAT\_DESCRIPTION 和 PREVIOUS\_GTIDS 事件在 MySqlStreamingChangeEventSource::setEventTimestamp 中
- **DBZ-7581** 在 PostgreSQL 失败时改进了日志
- **DBZ-7582** Unavailable toasted HSTORE Json Storage Mode 列会导致序列化失败
- **DBZ-7593** 第一次部署 Debezium, 无法捕获数据库中所有表的架构

- [DBZ-7594](#) Incorrect value of TIME (n)如果原始值为负值，则从 MySQL 复制
- [DBZ-7596](#) Re-select 后处理器不适用于复杂的类型
- [DBZ-7599](#) Null 而不是在"hex"模式配置时为二进制类型编写的占位符
- 在 schema 快照 DDL 处理期间 [DBZ-7608](#) Poor 快照性能
- [DBZ-7617](#) Incremental 快照查询没有遵循 message.key.columns 顺序
- [DBZ-7619](#) Metric ScnFreezeCount never increases
- 没有 mantissa 的 [DBZ-7643](#) Numeric 值无法解析
- [DBZ-7666](#) MySQL 连接器无法使用 RETURNING 关键字解析 DDL
- [DBZ-7690](#) Schema 历史比较器无法正确处理 SERVER\_ID\_KEY 和 TIMESTAMP\_KEY

## 2.5. 已弃用的功能

以下功能已弃用，并将在以后的发行版本中删除：

- Oracle 12 支持

## 2.6. 已知问题

以下已知问题会影响 Debezium 2.5.4：

- [Apicurio registry 2.4.3 和 2.4.4](#) 会导致 Kafka 上的无限重新平衡循环

