

Tencent Cloud DataBuddy

Data Integration

Product Documentation



Copyright Notice

©2013–2026 Tencent Cloud. All rights reserved.

Copyright in this document is exclusively owned by Tencent Cloud. You must not reproduce, modify, copy or distribute in any way, in whole or in part, the contents of this document without Tencent Cloud's the prior written consent.

Trademark Notice



All trademarks associated with Tencent Cloud and its services are owned by the Tencent corporate group, including its parent, subsidiaries and affiliated companies, as the case may be. Trademarks of third parties referred to in this document are owned by their respective proprietors.

Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

Contents

Data Integration

Supported Data Ingestion Solutions

Supported Data Sources and Read Capabilities

Data Source Management

List of Data Sources

Data Source Overview

Kafka Data Sources

MongoDB Data Source

MySQL Data Source

Oracle Data Source

PostgreSQL Data Source

Rest API Data Source

SQL Server Data Source

TDSQL-C MySQL Data Source

Offline Data Synchronization

Creating Single Table Synchronization Tasks

Table-to-Table Synchronization

Table-to-Volume Synchronization

File-to-Table Synchronization

File-to-Volume Synchronization

Batch Creating Single-Table Sync Tasks

Ops Monitoring for Offline Sync Tasks

Real-Time Data Synchronization

Real-Time Whole-Database Multi-Table Task Configuration

Real-Time Database and Table Sharding Task Configuration

Ops Monitoring for Real-Time Sync Tasks

Real-Time Sync Task Data Reconciliation

File Upload

Uploading Local Data to a Table

Uploading Local Files to a Table

Supported File Types and Constraints

File Parsing and Inference

Preview and Schema Editing

Uploading COS Files to a Table

Uploading Local Files to a Volume

Data Integration

Supported Data Ingestion Solutions

Last updated: 2026-10-09 16:09:03

Overview

DataBuddy Data Ingestion provides a unified entry point for importing data from external systems into the data lakehouse, covering three scenarios: **offline sync**, **real-time sync**, and **file upload**. It supports multiple granularities such as single table, batch single table, entire database with multiple tables, and sharded databases and tables. Regardless of the data source type and sync frequency, it consistently follows the main workflow of "Connect Data Source → Select Source Object → Configure Target → Set Run Policy → Publish and Run."

Ingestion Solutions Overview

DataBuddy provides the following five types of ingestion solutions. You can choose an appropriate solution based on the data source type, data scale, and timeliness requirements.

Access Method	Typical Scenario	Entries	Documentation
Offline Data Synchronization	Periodically synchronize data from relational databases, big data storage, and message queues to the data lakehouse in batches	Data ingestion → Task management → Create synchronization task → Offline single-table synchronization / Offline batch synchronization	Create single-table synchronization task / Batch create single-table synchronization tasks
Real-time Data Synchronization	Synchronize relational database CDC and message queue data in real time, with support for DDL change synchronization.	Data ingestion → Task management → Create synchronization task → Real-time whole-database multi-table synchronization / Real-time sharded database and table synchronization	Real-time whole-database multi-table task configuration / Real-time database/table sharding task configuration
File Upload to Table	Parse local or COS files and load them into structured tables.	Data ingestion → File upload → File to table / COS file to table	Local upload to table / COS file upload to table

File Upload to Volume	Upload files to a Volume in their original format while preserving the original file structure.	Data ingestion → File upload → File to volume	Upload local files to a volume
Batch File Sync to Volume	In an offline synchronization task, write structured data to a Volume in the specified format, or synchronize unstructured files to a Volume in batches.	Data ingestion → Task management → Create synchronization task (select a Volume type Catalog as the destination)	File-to-Volume sync

General Ingestion Process

Regardless of the ingestion solution you choose, you need to complete the following steps:

1. Prepare a data source connection (data source configuration).
↓
2. Prepare compute resources (use a real-time cluster for real-time ingestion and a platform cluster for offline ingestion).
↓
3. Create a sync task and select a sync method.
↓
4. Configure the source (select a database / table / file path, and configure filter conditions and filtering rules if necessary).
↓
5. Configure the target (Catalog / Schema / Table / Volume, and configure automatic or manual table creation if necessary).
↓
6. Configure field mappings and runtime parameters (such as dirty data threshold, concurrency, rate limit, and checkpoint interval).
↓
7. Configure alarms and run policies.
↓
8. Save / Publish / Start the task.

Comparison Between Offline and Real-Time Synchronization

Dimension	Offline Synchronization	Real-time Synchronization
-----------	-------------------------	---------------------------

Synchronization Granularity	Single table, batch single tables	Whole-database multi-table, sharded database and table
Triggering Method	Triggered by workflow scheduling or manual debugging	Runs continuously after startup and captures CDC changes from the source
Source Read Mode	Full read, incremental read (based on filter conditions)	Full + Incremental, Incremental Only
Write Mode	overwrite / append / upsert	append/upsert
Automatic table creation	Supports manual table creation	Supports automatic table creation (except for some sources such as Kafka)
Schema change synchronization	Not involved (based on the Schema at task execution time)	Supports DDL change synchronization (fully supported for MySQL / Tencent Cloud MySQL / TDSQL-C MySQL).
Data reconciliation	Supports full and incremental data reconciliation by the number of records	Supports full and incremental data reconciliation by the number of records / content.
Scenario	Scheduled data warehouse refresh, ODS layer loading, report data preparation	Real-time database ingestion into data lake, CDC streaming analysis

File Ingestion Capabilities

File ingestion provides an entry point for unstructured or semi-structured data to enter the lakehouse, covering the following three types of usage:

Entries	Description	Supported File Types
File to Table	After a file is uploaded, it is automatically parsed into a structured table and stored in the Schema under the specified Catalog.	.csv / .tsv / .json / .jsonl / .avro / .parquet / .txt / .xml
File to Volume	Files are stored in a Volume in their original format, preserving the binary / original structure.	Any type
COS File to Table	Select files from Tencent Cloud COS and parse them into structured tables.	.csv / .json / .xml / .avro / .parquet / .txt

File upload supports file-level parsing parameters (such as row and column delimiters, table headers, escape characters, JSON nesting, and XML row tags). After upload, the table is automatically registered in the

Catalog.

Suggestions for Selecting an Ingestion Method

Your Requirement	Recommended Solution
Synchronize data from business databases to the data lakehouse on a daily schedule.	Offline single-table synchronization + Workflow scheduling
Batch synchronization of multiple tables from the same source to the lakehouse	Offline Batch Synchronization Task
Real-time CDC ingestion from business databases into the data lake	Real-time whole-database multi-table synchronization task
Merge multiple sharded databases and tables into one target table.	Real-time sharded database and table synchronization task
One-time import of historical CSV/Parquet files for analysis	File to Table
Archive files to the lakehouse in their original format.	File to Volume

Prerequisites

Before using any ingestion solution, make sure you have completed the following preparations:

- You have activated the DataBuddy service and completed workspace initialization.
- The current account has **data access permissions** in the target Workspace. For details, see [Members and Permission Management](#).
- Compute resources for data ingestion have been created. For details, see [Creating Compute Resources](#).
- A corresponding data source connection has been created and tested based on the source type. For details, see [Add a Data Source Connection](#).
- The Catalog and Schema have been planned on the target side. For details, see [Unified Metadata Management Architecture](#).

Use Limits

Limit	Description
Computing Resource	Real-time ingestion tasks support selecting only real-time cluster compute resources; offline ingestion tasks support selecting only platform cluster compute resources.
Source Type	For the full range of supported data sources, see Supported Data Sources and Read/Write Capabilities

Target Type	The target of structured tasks is fixed as catalog tables (based on Iceberg native tables); unstructured data writes support catalog Volume volumes.
File Upload Size	Files to table: up to 10 files per upload, with a total size no greater than 2 GB; files to volume: up to 100 files per upload, with each file no greater than 5 GB.

References

- [Supported Data Sources and Read/Write Capabilities](#)
- [Real-Time Whole-Database Multi-Table Task Configuration](#)
- [Create Compute Resource](#)

Supported Data Sources and Read Capabilities

Last updated: 2026-10-09 16:09:07

Overview

DataBuddy data ingestion continuously expands the supported data source types, covering categories such as **relational databases**, **big data storage**, **semi-structured data**, **NoSQL databases**, and **message queues**. This document lists the currently supported data sources by category and their read capability support across different ingestion paths (offline synchronization, real-time full-database synchronization, and sharded database and table synchronization).

A data source is registered with DataBuddy as a data source connection. For details, see [Add a data source connection](#). The destination of a synchronization task is always a Catalog table (based on Iceberg native tables) or a Volume.

Read Capability Legend

Flag.	Description
✓	Supported
–	Currently not supported

Real-time synchronization paths typically have stricter requirements on source versions. For the detailed supported version range, refer to the Supported Version column.

Relational Database

Data Source	Offline read	Real-time full database read	Sharded database and table read	Supported Version
MySQL	✓	✓	✓	5.6 / 5.7 / 8.0.x
Tencent Cloud MySQL	✓	✓	✓	Same as MySQL
TDSQL-C MySQL	✓	✓	✓	5.7 / 8.0
TDSQL MySQL	✓	✓	–	InnoDB Type

PostgreSQL	✓	✓	✓	9.6 / 10 / 11 / 12
SQL Server	✓	✓	✓	2008 / 2008R2 / 2012 / 2014 / 2016 / 2017 / 2019
Oracle	✓	✓	✓	11 / 12 / 19
OceanBase	✓	✓	–	3.1.4 / 4.2.1.3
IBM DB2	✓	–	–	–
DM Database	✓	–	–	7 / 8
GBase	✓	–	–	–
GaussDB	✓	–	–	–
Greenplum	✓	–	–	4.x / 5.x / 6.x
TiDB	✓	–	–	–

Big Data Storage

Data Source	Offline read	Real-time full database read	Sharded database and table read	Supported Version
Hive	✓	–	–	1.x / 2.x / 3.x
HDFS	✓	–	–	2.x / 3.x
HBase	✓	–	–	2.2.x
TCHouse -D	✓	–	–	–
TCHouse -C	✓	–	–	–
TCHouse -X	✓	–	–	–
TCHouse -P	✓	–	–	–
Doris	✓	–	–	0.15 / 1.x
Iceberg	✓	–	–	0.13.1+

StarRocks	✓	–	–	–
ClickHouse	✓	–	–	20.7+
Kudu	✓	–	–	–
Hudi	✓	–	–	–
TDengine	✓	–	–	–
BigQuery	✓	–	–	–

Semi-Structured Data

Semi-structured data sources primarily handle unstructured file ingestion and offline reading scenarios. By combining file-to-table sync tasks with those targeting a Volume, you can achieve multiple usage modes such as archiving as-is, parsing into tables, and format conversion.

Data Source	Offline read	Real-time read	Description
COS	✓	–	Tencent Cloud COS; supports use as an offline sync source and as an entry for "COS file to table".
Amazon S3	✓	–	Overseas object storage; supports regex matching for file paths and full/incremental sync.
Azure Blob	✓	–	Access via container path
FTP	✓	–	Offline sync link only
SFTP	✓	–	Supports password and SSH private key authentication
File System	✓	–	Self-managed file system
Rest API	✓	–	Supports GET/POST, Header, and single/multiple requests; supports single JSON objects and JSON arrays.

NoSQL Database

Data Source	Offline read	Real-time full database read	Supported Version
-------------	--------------	------------------------------	-------------------

MongoDB	✓	✓	4.x
TencentDB for MongoDB	✓	✓	Same as MongoDB
Elasticsearch	✓	–	6.x / 7.x
Tencent Cloud Elasticsearch	✓	–	Same as Elasticsearch
CTSDB / InfluxDB	✓	–	–

Message queue

Data Source	Offline read	Real-time full database read	Supported Version
Kafka	✓	✓	2.4.1 / 2.7.1 / 2.8.1 / 2.8.2
CKafka	✓	✓	Same as Kafka

Viewing Available Data Source Types

The preceding list is for reference only. **When you go to Data Source Management or create a data source connection in a data ingestion task, you can view the complete list of types actually supported by the current version in the Data Source Type drop-down list.** For specific connection configuration items of each data source, see the corresponding section in [Data Source List](#).

References

- [Supported Access Solutions](#)
- [Add Data Source Connection](#)
- [Real-Time Whole-Database Multi-Table Task Configuration](#)

Data Source Management

Last updated: 2026-10-09 16:09:13

Overview

Data Source is the connection entry point for data ingestion tasks. In DataBuddy, data sources are centrally managed by the platform as **global objects** and are not bound to a specific ingestion task. A single data source definition can be reused by multiple offline synchronization, real-time synchronization, and file upload tasks, as well as external catalogs.

To avoid maintaining data source-related content separately in both the data ingestion module and the platform management module, DataBuddy consolidates common capabilities such as data source **creation, editing, connectivity testing, and permission management** into the Data Source Management section. This section serves as an entry point from the "Data Ingestion" perspective and guides you to complete specific operations in Platform Management.

How to Use Data Sources in Data Ingestion

Access Scenarios	Role of Data Source in It	Documentation
Offline data synchronization (single-table/batch/entire-database)	Source or target connection of a sync task	Offline data synchronization
Real-time data synchronization (entire-database/sharded-database-and-table)	Source connection of a sync task (the target is typically the built-in DataBuddy data source)	Real-time whole-database multi-table task configuration
Upload files to a table/volume	COS uploads require a COS data source. Local uploads do not require a pre-registered data source.	File upload
Data Source Types and Read/Write Capabilities	Before configuring a task, confirm whether the selected data source type supports being used as a source/target.	Supported data sources and read/write capabilities

Core Data Source Operations (Completed in Platform Management)

All management operations for data sources are performed in Platform Management > Data Source Management. The specific capabilities are as follows:

Capability	Entries
------------	---------

Create / view / edit / delete data sources; manage ACL permissions and associated tasks.	Global data source configuration
Configure data source connections (data source type, connection parameters, authentication method, and SSM credential hosting).	Add or Modify Data Source Connection
Run connectivity tests on data sources through specified compute resources to verify network reachability and credential validity.	Data source connectivity test
Built-in DataBuddy data source (default destination, non-editable)	Data Source Management Overview
Establish network connectivity for compute resources (when the data source is in a private VPC).	Compute resource network settings

Connection Fields for Each Data Source Type

When a data source is created, the **Data Source Type** drop-down list displays the complete list of types actually supported by the current version. For field descriptions and read/write capabilities of each data source type from the "Data Ingestion" perspective, see the following:

- [Supported Data Sources and Read/Write Capabilities](#)
- [Data Source List](#) and the documentation for each specific data source under it (MySQL / TDSQL-C MySQL / Oracle / PostgreSQL / SQL Server / MongoDB / Kafka, and so on)

Recommended Workflow

When connecting to an external data source for the first time, follow this order:

1. **Type Evaluation:** Check in [Supported Data Sources and Read/Write Capabilities](#) whether the required data source type supports your planned synchronization direction (as a source or as a destination).
2. **Network Connectivity** (if the data source is in your private VPC): Complete [Compute Resource Network Settings](#) first.
3. **Create a Data Source:** Create a data source connection in [Global Data Source Configuration](#).
4. **Connectivity Test:** Specify a compute resource and [complete the connectivity test](#) on the data source creation or editing page.
5. **Reference in Data Ingestion Tasks:** Select the corresponding task type under [Offline Data Synchronization](#) or [Real-Time Data Synchronization](#) based on your ingestion requirements.

References

- Platform Management: [Data Source Management Overview](#)
- Data Ingestion Overview: [Supported Access Solutions](#)
- Data Source Types and Read/Write Capabilities: [Supported Data Sources and Read/Write Capabilities](#)

- Compute Resource Network Settings: [Compute Resource Network Settings](#)
- Member and Permission Management: [Member and Permission Management](#)

List of Data Sources

Data Source Overview

Last updated: 2026-10-09 16:09:17

Overview

DataBuddy data ingestion continuously expands the supported data source types, covering five major categories: **relational databases**, **big data storage**, **semi-structured data**, **NoSQL databases**, and **message queues**. This document summarizes the currently supported data source types and the support status of each data source across different data ingestion paths, helping you quickly determine whether a target data source is available and choose an appropriate ingestion method.

Data sources are registered with DataBuddy as data source connections. For details, see [Data Source Management](#). For specific connection configuration items of each data source, refer to the corresponding section in the data source list.

DataBuddy data sources provide four types of ingestion scenarios: **offline synchronization**, **real-time synchronization**, **file upload**, and **direct analysis**. Offline synchronization and real-time synchronization are distinguished by two directions: read (source-side ingestion) and write (destination-side writing). DataBuddy focuses primarily on read, with the destination fixed as the built-in DataBuddy TCLake.

Data Source Type Overview

The following table summarizes the ingestion capabilities of each data source by data source type. **Offline integration** and **real-time integration** are distinguished by two directions: read and write. Read indicates that the data source can be ingested as a source, and write indicates that it can be written to as a destination. DataBuddy focuses primarily on read, with the destination fixed as the built-in DataBuddy TCLake. **Direct analysis** indicates that the data source supports direct queries through Linked Catalog, which analyzes source data directly without landing it in the lake.

Data Source Type	Data Source	Offline Integrati on Read	Offline Integrati on Write	Real-Time Integrati on Read	Real-Time Integrati on Write	Direct Query Analysis
Relational databases	MySQL	✓	✓	✓	–	✓
Relational databases	Tencent Cloud MySQL	✓	–	✓	–	✓
Relational databases	TDSQL-C MySQL	✓	–	✓	–	✓

Relational databases	PostgreSQL	✓	–	✓	–	✓
Relational databases	Oracle	✓	–	✓	–	–
Relational databases	SQL Server	✓	–	✓	–	–
Relational databases	TDSQL MySQL	✓	–	–	–	✓
Relational databases	TDSQL PostgreSQL	✓	–	–	–	–
Relational databases	IBM DB2	✓	–	–	–	–
Relational databases	DM Database	✓	–	–	–	–
Relational databases	Greenplum	✓	–	–	–	–
Relational databases	GBase	✓	–	–	–	–
Relational databases	GaussDB	✓	–	–	–	✓
Relational databases	TiDB	✓	–	–	–	–
Relational databases	OceanBase	✓	–	–	–	–
Big data storage	HDFS	✓	–	–	–	–
Big data storage	HBase	✓	✓	–	–	–
Big data storage	Hive	✓	–	–	–	–
Big data storage	Kudu	✓	–	–	–	–

Big data storage	ClickHouse	✓	✓	–	–	–
Big data storage	TCHouse–C	✓	–	–	–	–
Big data storage	TCHouse–D	✓	–	–	–	✓
Big data storage	TCHouse–X	✓	–	–	–	–
Big data storage	TCHouse–P	✓	–	–	–	–
Big data storage	Doris	✓	–	–	–	✓
Big data storage	StarRocks	✓	–	–	–	✓
Big data storage	Iceberg	✓	–	–	–	–
Big data storage	Hudi	✓	–	–	–	–
Big data storage	TDengine	✓	–	–	–	–
Big data storage	BigQuery (Overseas)	✓	–	–	–	–
Semi-structured data	FTP	✓	–	–	–	–
Semi-structured data	SFTP	✓	–	–	–	–
Semi-structured data	COS	✓	✓	–	–	–
Semi-structured	REST API	✓	–	–	–	–

data						
Semi-structured data	File System	✓	–	–	–	–
Semi-structured data	Amazon S3	✓	–	–	–	–
NoSQL databases	MongoDB	✓	–	✓	–	–
NoSQL databases	TencentDB for MongoDB	✓	–	✓	–	–
NoSQL databases	Elasticsearch	✓	–	–	–	–
NoSQL databases	Tencent Cloud Elasticsearch	✓	–	–	–	–
NoSQL databases	CTSDB / InfluxDB	✓	–	–	–	–
Message queue	Kafka	✓	–	✓	–	–
Message queue	CKafka	✓	–	✓	–	–

References

- [Supported Access Solutions](#)
- [Supported Data Sources and Read Capabilities](#)
- [Data Source Management](#)
- [Real-Time Whole-Database Multi-Table Task Configuration](#)
- [Real-Time Database/Table Sharding Task Configuration](#)

Kafka Data Sources

Last updated: 2026-10-09 16:09:22

Overview

Kafka is a distributed stream processing platform. DataBuddy data ingestion supports reading messages from Kafka (including Tencent Cloud CKafka) as a source, covering both offline reading and real-time whole-database scenarios. This document describes how to use it in the order of "Create Data Source → Configure Sync Task → Troubleshoot Common Issues".

Supported Versions

Type	Supported Version
Self-Built Kafka	2.x / 3.x
CKafka	2.4.1 / 2.8.1 / 3.2.3

Read Capability

Capability	Offline read	Real-time full database read
Supported	✓	✓

For a complete comparison of data source capabilities, see [Supported Data Sources and Read Capabilities](#).

Use Limits

Serialization Format Support

Scenario	Read Format
Full database synchronization	<code>canal-JSON</code> / <code>debezium</code> / <code>ogg-json</code> / <code>csv</code> / <code>json</code> / <code>raw</code>

For whole-database synchronization, DDL changes support only **adding columns, dropping columns, and creating tables**.

Creating a Data Source

Operation Steps

1. Log in to the [DataBuddy console](#).

2. Switch to the target region and Workspace at the top.
3. Go to **Data Integration > Task Management > Data Source Management** (or **Platform Management > Data Source Management**).
4. Click **Add Data Source** and select **Kafka**.
5. Enter the connection configuration and authentication information. For details, see [Parameter Description](#).
6. Click **Test Connection** (see [Data Source Connectivity Test](#)).
7. After the test passes, click **Save** or **Save & Create Task**.

Parameter Description

Parameter	Description	Required
Data Source Name	Unique identifier of the data source within the workspace	Yes
Kafka Type	Self-built open source Kafka / Tencent Cloud CKafka	Yes
Bootstrap Servers	<code>host1:port1,host2:port2</code> (for example, <code>10.0.0.1:9092,10.0.0.2:9092</code>)	Yes
Authentication Method	See Authentication Method	No
Username / Password	Required when SASL authentication is selected (passwords support SSM credential hosting).	Depends on Authentication

Authentication Method

Method	Description	Required field
No Authentication (PLAINTEXT)	Authentication Disabled	–
SASL_PLAINTEXT	Plaintext SASL Authentication	Username and password
SASL_GSSAPI(Kerberos)	Kerberos Authentication	keytab file, conf file, principal, and Kafka service name
SASL_SSL	SASL + SSL Dual Authentication	Truststore certificate file, Keystore password, username, and password

SSL	Mutual TLS Authentication	Truststore certificate file, Truststore password, Keystore certificate file, Keystore password, and private key password
-----	---------------------------	--

Network and Server Preparation

- The integration resource group must have **bidirectional connectivity** to all brokers.
- The `listeners` / `advertised.listeners` in the server-side `server.properties` must be set to **externally reachable** addresses, which is a common root cause of connection failures.

Using in Data Ingestion Tasks

Real-Time Sync: Source

Parameter	Description
Data Source	Created Kafka data source that supports connectivity tests.
Topic	Topic to consume, supporting multi-select from the dropdown list.
Serialization Format	See Serialization Format Support
Read Position	Earliest (earliest) / Latest (latest) / Specified Time
Subscription Type	You can select INSERT / UPDATE / DELETE, and at least one must be selected.

Nested JSON

Nested JSON is not supported by default. To enable it, add the following to the advanced parameters:

```
source.nested.json.enabled=true
```

Tip

This parameter applies to both the Kafka **key** and **value**. If the key is not JSON (such as a plain string or numeric primary key), append `source.key.nested.json.enabled=false`. Otherwise, an error will be reported during key parsing.

To reference nested fields, declare them in the `parent_field->child_field` format. Use multiple `->` to connect multiple levels of nesting. When referencing them in a function, enclose the reference in backticks:

```
TO_TIMESTAMP(`book->published`, 'yyyyMMdd')
```

Advanced Parameters

Except for the parameters listed in the following table, all native Kafka client parameters can be passed through by adding the `properties.` prefix, for example, `properties.fetch.max.wait.ms=500`.

Parameter	Description	Default Value
<code>source.nested.json.enabled</code>	Enable nested JSON (applies to both keys and values).	<code>false</code>
<code>source.key.nested.json.enabled</code>	Whether to enable nested JSON for keys	Follows <code>source.nested.json.enabled</code>
<code>source.ignore-parse-errors</code>	Skip error records when parsing fails.	<code>false</code>
<code>scan.topic-partition-discovery.interval</code>	Interval for periodically discovering new partitions / topics; set to <code>0</code> to disable.	<code>5min</code>
<code>timestamp.format</code>	Time field parsing format	<code>ISO8601</code>

Data Format Example

canal-json

```
{
  "data": [{ "id": "1", "name": "Zhang San", "age": "25" }],
  "database": "test_db",
  "table": "user_info",
  "type": "INSERT",
  "es": 1589373560000,
  "ts": 1589373560798,
  "isDdl": false,
  "mysqlType": { "id": "int", "name": "varchar(50)", "age": "int" },
  "sqlType": { "id": 4, "name": 12, "age": 4 },
  "pkNames": ["id"],
  "old": null,
  "sql": ""
}
```

debezium

```
{
  "before": null,
  "after": { "id": 1, "name": "Zhang San", "age": 25 },
  "source": {
    "version": "1.5.0.Final",
    "connector": "mysql",
    "name": "mysql_server",
    "ts_ms": 1589373560000,
    "db": "test_db",
    "table": "user_info"
  },
  "op": "c",
  "ts_ms": 1589373560798
}
```

FAQs

Q: Connection Failure or Test Connection Timeout?

A: Troubleshoot in the following order:

1. Check whether the integration resource group has bidirectional connectivity to the broker network through the VPC, allowlist, and security group.
2. Check whether the address returned by `advertised.listeners` is reachable from the network where the resource group resides.
3. Check whether the username, password, and SASL mechanism used for SASL authentication are consistent with those on the server.

Q: No Data Consumed?

Possible cause	How to Confirm
The read position is set to latest, but the Topic has no new messages.	Change to earliest or a specified time point and restart.
The Topic actually has no data.	consume once directly from the Kafka console.
The Consumer group already has a persisted offset externally.	Change to a different group ID and restart.

Data is completely filtered out by the filter operation or database/table scope.

Check the corresponding configuration of the sync task.

Q: Serialization Format Parsing Failure or All Fields null?

A: Troubleshoot in the following order:

1. Use a consumer tool to pull a raw message and compare it with the configured serialization format.
2. The message is nested JSON, but **is not enabled** `source.nested.json.enabled=true`.
3. The key is not JSON, but nested JSON is enabled. Add `source.key.nested.json.enabled=false`.

Q: Missing mysqlType / sqlType / pkNames in the Entire Database?

A: For whole-database synchronization with `canal-json` / `debezium`, messages must carry these metadata fields. Confirm that the upstream Canal / Debezium output includes them. You can use a tool to pull a message and verify.

Q: SQL Syntax Error When Nested Field Functions Are Processed?

A: `-` and `>` are not valid SQL identifiers. When referencing nested fields in a function, enclose them in backticks and confirm that the fields have been declared through "Add Field":

```
TO_TIMESTAMP(`book->published`, 'yyyyMMdd')
```

References

- [Add Data Source Connection](#)
- [Data Source Connectivity Test](#)
- [Supported Data Sources and Read Capabilities](#)
- [Create a Single-Table Sync Task](#)
- [Real-Time Whole-Database Multi-Table Task Configuration](#)

MongoDB Data Source

Last updated: 2026-10-09 16:09:26

Overview

MongoDB is a high-performance, open-source NoSQL document database known for its flexible document model and strong horizontal scalability. It is widely used in Web applications, mobile applications, and big data analytics scenarios. DataBuddy data ingestion supports using MongoDB as a source for real-time full-database synchronization and offline synchronization. This document describes its connection configuration, supported capabilities, prerequisites, and common issues.

Supported Versions

Type	Supported Version
Self-Built MongoDB	3.6 / 4.0 / 4.2 / 4.4 / 5.0 / 6.0 / 7.0 / 8.0
TencentDB for MongoDB	3.6 / 4.0 / 4.2 / 4.4 / 5.0 / 6.0 / 7.0 / 8.0

Real-time synchronization (Change Streams) requires MongoDB version 3.6 or later.

Read Capability

Capability	Offline read	Real-time full database read
Supported	✓	✓

For a complete comparison of data source capabilities, see [Supported Data Sources and Read Capabilities](#).

Use Limits

Real-Time Sync Limitations

Limit	Description
Version Requirement	MongoDB version must be 3.6 or later.
Replica set	Must be configured in Replica Set mode to use Change Streams.
oplog Retention	Sufficient oplog retention time is required to support incremental sync.
System Collections	Synchronization of system collections such as <code>system.*</code> is not supported.

Offline Sync Limitations

Limit	Description
Single document size	No more than 16 MB (MongoDB limit)
Nesting depth	Control document nesting depth. Excessive nesting may affect performance.

Supported Field Types

For the complete list of BSON types, see the [MongoDB official documentation](#).

Field Type	Offline read	Real-time read
Int32 / Int64 / Long / Double / Decimal128	✓	✓
Boolean / String / ObjectId / UUID / Symbol / Regex	✓	✓
JavaScript / BinData	✓	✓
Date / Timestamp	✓	✓
Object / Array / Null / Undefined / MinKey / MaxKey / DBPointer	✓	✓

MongoDB is a Schema-free database, and the same field may contain multiple types. It is recommended to unify field types at the source to avoid synchronization exceptions.

Supported DML/DDL for Real-Time Sync

DML

Operation	Whether Supported	Description
INSERT / UPDATE / REPLACE / DELETE	✓	Document-level change operation

DDL

Operation	Whether Supported
DROP COLLECTION / DROP DATABASE / RENAME COLLECTION	✗

MongoDB captures changes based on Change Streams and does not support automatic synchronization of DDL changes. Operations such as collection deletion and renaming must be handled manually at the destination.

Prerequisites

1. Confirming Replica Set Configuration (Real-Time Sync Only)

```
rs.status()
```

2. Configuring Account Permissions

Offline synchronization:

```
use admin
db.createUser({
  user: "wedata_user",
  pwd: "your_password",
  roles: [{ role: "read", db: "your_database" }]
})
```

Real-time synchronization:

```
use admin
db.createUser({
  user: "wedata_cdc_user",
  pwd: "your_password",
  roles: [
    { role: "read", db: "your_database" },
    { role: "read", db: "local" } // Required for reading the oplog.
  ]
})
```

3. Configuring oplog Size (Real-Time Sync Only)

```
use local
db.oplog.rs.stats() // View the current
oplog configuration.
db.adminCommand({ replSetResizeOplog: 1, size: 10240 }) // Unit: MB
```

To modify the oplog size, you must perform the operation on each node in the replica set.

Creating a Data Source

Operation Steps

1. Log in to the [DataBuddy console](#).
2. Switch to the target region and Workspace at the top.
3. Go to **Data Integration > Task Management > Data Source Management** (or **Platform Management > Data Source Management**).
4. Click **Add Data Source** and select **MongoDB**.
5. Enter the connection configuration and authentication information. For details, see [Parameter Description](#).
6. Click **Test Connection** (see [Data Source Connectivity Test](#)).
7. After the test passes, click **Save** or **Save & Create Task**.

Parameter Description

Parameter	Description	Required
Data Source Name	Unique identifier of the data source within the workspace	Yes
Connection Address	MongoDB connection address (see the format below)	Yes
Database	Default database to connect	Yes
Username.	Database connection account (if authentication is enabled)	Depends on Authentication
Password	Database connection password (if authentication is enabled; SSM credential hosting supported)	Depends on Authentication
Data Source Version	See Supported Versions	Yes

Connection Address Format

Deployment Mode	Format example
Single-node	<code>mongodb://host:port</code>
Replica set	<code>mongodb://host1:port1,host2:port2,host3:port3/?replicaSet=rsName</code>
Sharded cluster	<code>mongodb://mongos1:port1,mongos2:port2</code>

Complete example:

```
mongodb://10.0.0.1:27017,10.0.0.2:27017,10.0.0.3:27017/?replicaSet=rs0
mongodb://10.0.0.1:27017/mydb?authSource=admin
```

Using in Data Ingestion Tasks

Real-Time Full Database Sync

Configuration Item	Description
Source Table Scope	Specified collections (only the selected collections are synchronized; new collections require a restart) / regex matching (specify database name + collection name regex; new collections that match the rule are automatically added)
Read Mode	Full + incremental / Incremental only
Subscription Type	You can select INSERT / UPDATE / DELETE, and at least one must be selected.

Offline Single Table Sync

Parameter	Description
Database / Collection	Required; supports selection or manual input.
Filtering Condition	MongoDB query condition (JSON format)

Filter condition examples:

```
{"age": {"$gte": 18}, "status": "active"}
{"create_time": {"$gte": {"$date": "2024-01-01T00:00:00Z"}}}
```

For detailed task configuration, see [Create a Single-Table Synchronization Task](#) and [Real-time Whole-Database Multi-Table Task Configuration](#).

Best Practices

- **Account Permission Minimization:** Use a dedicated account and avoid root privileges.

- **oplog Retention Period:** Set it to at least 72 hours to prevent data loss after task restarts.
- **Index Optimization:** Create indexes on query fields during offline synchronization to improve read efficiency.
- **Network Connectivity:** Use a private network connection whenever possible.

FAQs

Q: Connection Failed?

A: Check network connectivity, confirm that the username and password are correct, and check the `authSource` configuration.

Q: Change Streams Unavailable?

A: MongoDB is not configured in replica set mode, or its version is earlier than 3.6. Configure MongoDB as a replica set and upgrade it to version 3.6 or later.

Q: oplog Not Found Error During Incremental Sync?

A: The oplog retention period is too short, and the oplog has been overwritten. Increase the oplog size:

```
db.adminCommand({ replSetResizeOplog: 1, size: 20480 })
```

Q: Type Mismatch Error During Sync?

A: MongoDB is Schema-free, and the same field may contain different types. It is recommended to unify data types at the source or configure type conversion at the write end.

Q: Some Documents Failed to Sync?

A: A single document exceeds the 16 MB limit. Split large documents, or use GridFS to store large files.

References

- [Add Data Source Connection](#)
- [Data Source Connectivity Test](#)
- [Supported Data Sources and Read Capabilities](#)
- [Create a Single-Table Sync Task](#)
- [Real-Time Whole-Database Multi-Table Task Configuration](#)

MySQL Data Source

Last updated: 2026-10-09 16:09:30

Overview

MySQL is a widely used open-source relational database. DataBuddy data ingestion supports using MySQL as a source, covering various scenarios such as offline synchronization, real-time full-database synchronization, and sharded database and table synchronization. This document describes the connection configuration, supported capabilities, prerequisites, and common issues of MySQL data sources in DataBuddy.

Supported Versions

Type	Supported Version
Self-Built MySQL	5.6.x / 5.7.x / 8.0.x / 8.1.x / 8.2.x / 8.3.x / 8.4.x
TencentDB for MySQL (CDB)	5.6 / 5.7 / 8.0
JDBC Driver	8.0.21

Read Capability

Capability	Offline read	Real-time full database read	Sharded database and table read
Supported	✓	✓	✓

For a complete comparison of data source capabilities, see [Supported Data Sources and Read Capabilities](#).

Use Limits

Real-Time Sync Limitations

Limit	Description
Binlog format	Only ROW format is supported.
<code>binlog_row_image</code>	Must be set to FULL
Binlog retention period	Retain for at least 72 hours; the larger the full data volume, the longer the retention period required.
Table without primary key	Tables without primary keys cannot guarantee exactly-once semantics, and data duplication may occur.

XA transaction	XA ROLLBACK is not supported, and related tables must be processed manually.
Read-only database	Read-only database instances earlier than version 5.6.x are not supported.
Functional Index	Tables containing Functional indexes are not supported (a new feature in MySQL 8.0).
Cascading deletion	Synchronizing cascading deletion of associated table records is not supported.
Stored procedures and materialized views	Reading is not supported.
SET data type	Real-time synchronization is not supported.
Online DDL tool	Only gh-ost is supported, and other online DDL change tools are not supported.

Offline Sync Limitations

Limit	Description
Concurrency for Sharded Table Synchronization	When multiple tables are split into a single table during synchronization, the task concurrency must be greater than the number of tables.
Stored Procedure	Reading is not supported.

Supported Field Types

For complete field types, see [MySQL official documentation](#). The following table shows DataBuddy support across different links (excerpted from MySQL 8.0.x).

Field Type	Offline read	Real-time read
TINYINT / TINYINT UNSIGNED	✓	✓
SMALLINT / MEDIUMINT / INT / BIGINT (including UNSIGNED)	✓	✓
FLOAT / DOUBLE / DECIMAL / NUMERIC	✓	✓
REAL	✗	✓
CHAR / VARCHAR / TEXT Series	✓	✓
JSON	✓	✓

BINARY / VARBINARY / BLOB Series	✓	✓
DATE / TIME / DATETIME / TIMESTAMP / YEAR	✓	✓
BIT / BOOL / BOOLEAN / ENUM	✓	✓
SET	✓	✗
GEOMETRY / POINT	✗	✓
Spatial types such as LINESTRING / POLYGON / MULTIPOINT	✗	✗

Spatial types (GEOMETRY series) have limited support. Convert them to string types before synchronization.

Supported DML/DDL for Real-Time Sync

DML

Operation	Whether Supported
INSERT / DELETE / UPDATE	✓

DDL

Operation	Whether Supported	Supported syntax examples
Add column (ADD COLUMN)	✓	<code>ALTER TABLE t ADD col TYPE [FIRST AFTER col]</code>
Drop column (DROP COLUMN)	✓	<code>ALTER TABLE t DROP [COLUMN] col</code>
Rename column (RENAME / CHANGE COLUMN)	✓	<code>ALTER TABLE t CHANGE old new TYPE</code>
Modify column type (MODIFY COLUMN)	✓	<code>ALTER TABLE t MODIFY col TYPE</code>
Add table (CREATE TABLE)	✓	CHECK and Temporary tables are not supported.
Rename table / Delete table / Clear table	✗	

DDL change response **takes effect only during the incremental phase**. For high-risk operations such as **dropping or renaming tables**, the destination typically does not support automatic response. For detailed DDL change response policies (automatic change, ignore, and log alarm), see [Real-time Full-Database Multi-Table Task Configuration](#).

Prerequisites

1. Configuring Account Permissions

Offline synchronization:

```
CREATE USER 'wedata_user'@'%' IDENTIFIED BY 'your_password';
GRANT SELECT ON your_database.* TO 'wedata_user'@'%';
FLUSH PRIVILEGES;
```

Real-time synchronization:

```
CREATE USER 'wedata_cdc_user'@'%' IDENTIFIED BY 'your_password';
GRANT SELECT, SHOW DATABASES, REPLICATION SLAVE, REPLICATION CLIENT
    ON *.* TO 'wedata_cdc_user'@'%';
FLUSH PRIVILEGES;
```

We recommend creating a dedicated account for sync tasks to avoid mixing it with business accounts.

2. Enabling Binlog (Real-Time Sync Only)

```
SHOW VARIABLES LIKE 'log_bin';    -- ON indicates that it is enabled.
```

Binlog is enabled by default for Tencent Cloud MySQL. For self-managed instances, configure it in `my.cnf`:

```
[mysqld]
log-bin=mysql-bin
server-id=1
binlog_format=ROW
binlog_row_image=FULL
```

Restart the database after the modification for the changes to take effect.

3. Setting Session Timeout (Recommended)

For full synchronization of large databases, we recommend increasing the session timeout:

```
SET GLOBAL interactive_timeout = 28800;  
SET GLOBAL wait_timeout = 28800;
```

Creating a Data Source

Operation Steps

1. Log in to the [DataBuddy console](#).
2. Switch to the target region and Workspace at the top.
3. Go to **Data Integration > Task Management > Data Source Management** (or **Platform Management > Data Source Management**).
4. Click **Add Data Source** and select **MySQL**.
5. Enter the connection configuration and authentication information. For details, see [Parameter Description](#).
6. Click **Test Connection** (see [Data Source Connectivity Test](#)).
7. After the test passes, click **Save** or **Save & Create Task**.

Parameter Description

Parameter	Description	Required	Default Value
Data Source Name	Unique identifier of the data source within the workspace	Yes	—
Connection Method	String connection (JDBC URL) / Tencent Cloud instance	Yes	String concatenation
JDBC URL	Connection string in string connection mode, in the format <code>jdbc:mysql://host:port/database</code>	Required for string connection	—
Region and Instance	Select an instance in Tencent Cloud instance mode	Required for Tencent Cloud instance	—

Username.	Database connection account	Yes	—
Password	Database connection password (SSM credential hosting supported)	Yes	—
Data Source Version	5.6 / 5.7 / 8.0.x	Yes	—
Advanced Parameters	Connection-level extended parameters, such as <code>useUnicode</code> , <code>characterEncoding</code> , <code>serverTimezone</code> , and <code>connectTimeout</code>	No	—

Example JDBC URL:

```
jdbc:mysql://10.0.0.1:3306/mydb?
useUnicode=true&characterEncoding=utf8&serverTimezone=Asia/Shanghai
```

Using in Data Ingestion Tasks

Real-Time Full-Database and Sharded Database/Table Sync

Configuration Item	Description
Source Table Scope	Specified tables (only selected tables are synchronized; new tables require a restart) / regex matching (specify database name + table name regex; new tables that match the rule are automatically added)
Read Mode	Full + incremental / Incremental only
Subscription Type	You can select INSERT / UPDATE / DELETE, and at least one must be retained.
gh-ost Synchronization	Whether to synchronize temporary tables generated by gh-ost online DDL.

Offline Single Table Sync

Configuration	Description
---------------	-------------

Item	
Data source.	Configured MySQL data source
Database / Table	Supports exact specification and regex matching. Table names support ranges such as <code>table_[0-99]</code>
Split Key	Supports integer fields only; primary key or indexed columns recommended; avoid COLLATE columns.
Filtering Condition	Standard WHERE syntax, supports time parameters (such as <code>gmt_create > \$bizdate</code>).

For detailed task configuration, see [Create a Single-Table Synchronization Task](#) and [Real-time Whole-Database Multi-Table Task Configuration](#).

Best Practices

Practice	Recommendation
Account Permission Minimization	Create a dedicated account for the synchronization task, grant only necessary permissions, and do not use root.
Binlog retention period	At least 72 hours; longer retention is recommended for full synchronization of large tables.
Split key selection	Primary key or index columns; avoid COLLATE columns to prevent data duplication.
Network connectivity establishment	Prioritize private network connections to improve transfer performance.

Handling Primary/Secondary Switchover (Real-Time Sync)

Whether a real-time synchronization task can recover smoothly during a MySQL primary/secondary switchover depends on the address type and whether GTID is enabled:

Scenario	Solution
TencentDB for MySQL (unchanged instance address)	The interruption lasts only seconds. The task automatically reconnects to the new primary and resumes from the latest snapshot position. failover is completed automatically without manual intervention.
Self-Built MySQL + VIP/DNS	After VIP/DNS drifts to the new primary, the task automatically reconnects. It is recommended to set the number of task retries to at least 3.

Self-Built MySQL +
Fixed IP address

Pause the task before the switch, modify the data source IP address after the switch, and then resume running.

It is strongly recommended to enable GTID: In a primary/secondary switchover scenario, recovery based on GTID ensures no data duplication or loss.

With asynchronous replication, a primary database failure may lose the last few unsynchronized transactions, which is a limitation of the MySQL replication layer itself. For zero data loss, use strong synchronous or semi-synchronous replication.

FAQs

Q: Why does the error `A slave with the same server_uuid/server_id as this slave has connected to the master` occur?

Multiple synchronization tasks used the same server-id. The system now generates server-id randomly. If an old task has explicitly configured `server-id` in its advanced parameters, delete this parameter.

Q: Why does the error `Cannot replicate because the master purged required binary logs` occur?

The Binlog files on MySQL have been cleaned up. Increase the Binlog retention period or optimize job concurrency to speed up consumption.

Q: Why does the error `Connection reset by peer` or a connection timeout occur?

A: Network issues or job backpressure cause idle timeout. You can increase `slave_net_timeout` (for example, 120) or optimize job parallelism and memory.

Q: JobManager OOM?

A: Large data volume or sparse primary key range distribution causes excessive splitting. You can:

- Increase the JobManager CU.
- Adjust the advanced parameter `scan.incremental.snapshot.chunk.size` (default: 8096).
- Adjust `split-key.even-distribution.factor.upper-bound`.

Q: What happens if `binlog_format` is changed from ROW to STATEMENT for MySQL during real-time task execution?

A: A real-time task only validates `binlog_format=ROW` at startup and does not check it again during runtime. After `binlog_format` is switched to STATEMENT, all DML statements are treated as DDL by the

parser, so the downstream cannot detect data changes, resulting in silent data loss. Do not modify `binlog_format` while the task is running.

Q: Why is some data duplicated when a COLLATE column is used as the split key?

A: COLLATE affects sorting, filtering, and grouping results. When setting a split key, select a non-COLLATE column.

Q: Is gh-ost Online DDL Supported?

A: Yes. After you enable the gh-ost synchronization switch, the system monitors and synchronizes changes to `*_gho` temporary tables. Prerequisites: gh-ost can access MySQL normally. For Tencent Cloud CDB, append the `--aliyun-rds` parameter to the gh-ost command.

References

- [Add Data Source Connection](#)
- [Data Source Connectivity Test](#)
- [Supported Data Sources and Read Capabilities](#)
- [Create a Single-Table Sync Task](#)
- [Real-Time Whole-Database Multi-Table Task Configuration](#)

Oracle Data Source

Last updated: 2026-10-09 16:09:35

Overview

Oracle is an enterprise-grade relational database that is widely used in core business systems across industries such as finance, telecommunications, and government. DataBuddy data ingestion supports using Oracle as a source, covering scenarios such as offline synchronization, real-time full-database synchronization, and sharded database and table synchronization. This document describes the connection configuration, supported capabilities, prerequisites, and common issues of Oracle data sources in DataBuddy.

Supported Versions

Type	Supported Version
Oracle	11g / 12c / 19c
JDBC Driver	ojdbc6

New features of Oracle 19c are not supported.

Read Capability

Capability	Offline read	Real-time full database read	Sharded database and table read
Supported	✓	✓	✓

For a complete comparison of data source capabilities, see [Supported Data Sources and Read Capabilities](#).

Use Limits

Real-Time Sync Limitations (Based on LogMiner)

Limit	Description
Log Retention Period	Incremental logs are retained for at least 72 hours.
Standby Database Support	Only logical standby is supported, and physical standby is not supported.

Table without primary key	Tables without primary keys cannot guarantee exactly-once semantics.
Account restrictions	Do not use the system or sys account to configure data sources.
DDL risk	Adding a column with a default value may cause parsing exceptions; LogMiner may cause DML and DDL order disorder.
Performance bottleneck	When LogMiner reads at a rate exceeding 5k/s, significant latency may occur.
Risk in RAC mode	Log file switching may cause LogMiner to fail to detect changes, resulting in data loss.

If the preceding limitations are unacceptable, consider using the OGG (Oracle GoldenGate) solution.

Offline Sync Limitations

Limit	Description
View	Supports view reading
Encoding format.	Only <code>UTF8</code> / <code>AL32UTF8</code> / <code>AL16UTF16</code> / <code>ZHS16GBK</code> are supported.

Supported Field Types

For complete field types, see the [Oracle official documentation](#).

Field Type	Offline read	Real-time read
NUMBER / INTEGER / INT / SMALLINT / NUMERIC / DECIMAL	✓	✓
FLOAT / BINARY_FLOAT / DOUBLE PRECISION / BINARY_DOUBLE / REAL	✓	✓
CHAR / VARCHAR / VARCHAR2 / NCHAR / NVARCHAR2	✓	✓
CLOB / NCLOB	✓	✓
DATE / TIMESTAMP / TIMESTAMP WITH TIME ZONE / TIMESTAMP WITH LOCAL TIME ZONE	✓	✓
BLOB / RAW / LONG RAW	✓	✓

ROWID	×	✓
INTERVAL YEAR TO MONTH / INTERVAL DAY TO SECOND	×	✓
XMLTYPE / BFILE	×	×

The NUMBER type is automatically mapped to TINYINT / SMALLINT / INT / BIGINT or DECIMAL based on its precision. If the precision is not specified, it is mapped to DECIMAL(38, 18) by default.

Supported DML/DDL for Real-Time Sync

DML

Operation	Whether Supported
INSERT / DELETE / UPDATE	✓

DDL

Operation	Whether Supported
Add table / Add column / Delete column / Rename column / Modify column type	✓
Delete table / Clear table / Rename table (RENAME TABLE)	×

DDL change response **takes effect only during the incremental phase**. Adding a column with a default value may cause parsing exceptions, and LogMiner may cause DML and DDL operations to be out of order.

Prerequisites

1. Configuring Account Permissions

Offline synchronization:

```
CREATE USER wedata_user IDENTIFIED BY your_password;
GRANT CREATE SESSION TO wedata_user;
GRANT SELECT ANY TABLE TO wedata_user;
```

Real-time synchronization:

```
CREATE USER wedata_cdc_user IDENTIFIED BY your_password DEFAULT
TABLESPACE USERS QUOTA UNLIMITED ON USERS;

GRANT CREATE SESSION, SELECT ANY TABLE, SELECT_CATALOG_ROLE TO
wedata_cdc_user;

GRANT ANALYZE ANY TO wedata_cdc_user;

GRANT EXECUTE_CATALOG_ROLE TO wedata_cdc_user;

GRANT LOGMINING TO wedata_cdc_user;           -- Required for 12c and
later versions, not required for 11g.

GRANT EXECUTE ON DBMS_LOGMNR TO wedata_cdc_user;

GRANT EXECUTE ON DBMS_LOGMNR_D TO wedata_cdc_user;

-- System view query permission
GRANT SELECT ON V_$DATABASE TO wedata_cdc_user;
GRANT SELECT ON V_$LOG TO wedata_cdc_user;
GRANT SELECT ON V_$LOG_HISTORY TO wedata_cdc_user;
GRANT SELECT ON V_$LOGMNR_CONTENTS TO wedata_cdc_user;
GRANT SELECT ON V_$LOGMNR_LOGS TO wedata_cdc_user;
GRANT SELECT ON V_$ARCHIVED_LOG TO wedata_cdc_user;
GRANT SELECT ON V_$ARCHIVE_DEST_STATUS TO wedata_cdc_user;
```

 Note:

When table field changes are monitored, **do not use** the `system` or `sys` account to configure data sources.

2. Enabling Log Archiving (Real-Time Sync Only)

A database restart is required, and disk space will be consumed.

```
sqlplus / as sysdba

ALTER SYSTEM SET db_recovery_file_dest_size = 10G;
ALTER SYSTEM SET db_recovery_file_dest =
'/opt/oracle/oradata/recovery_area' scope=spfile;
SHUTDOWN IMMEDIATE;
STARTUP MOUNT;
ALTER DATABASE ARCHIVELOG;
ALTER DATABASE OPEN;
ARCHIVE LOG LIST;           -- Verify the archiving mode.
```

3. Enabling Supplemental Logging (Real-Time Sync Only)

If it is not enabled, other columns downstream of the UPDATE operation will be empty.

```
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;  
ALTER TABLE schema_name.table_name ADD SUPPLEMENTAL LOG DATA (ALL)  
COLUMNS;
```

Creating a Data Source

Operation Steps

1. Log in to the [DataBuddy console](#).
2. Switch to the target region and Workspace at the top.
3. Go to **Data Integration > Task Management > Data Source Management** (or **Platform Management > Data Source Management**).
4. Click **Add Data Source** and select **Oracle**.
5. Enter the connection configuration and authentication information. For details, see [Parameter Description](#).
6. Click **Test Connection** (see [Data Source Connectivity Test](#)).
7. After the test passes, click **Save** or **Save & Create Task**.

Parameter Description

Parameter	Description	Required
Data Source Name	Unique identifier of the data source within the workspace	Yes
JDBC URL	<code>jdbc:oracle:thin:@host:port:sid</code> or <code>jdbc:oracle:thin:@//host:port/service_name</code>	Yes
Username.	Database connection account (system / sys not allowed)	Yes
Password	Database connection password (SSM credential hosting supported)	Yes
Data Source Version	11g / 12c / 19c	Yes

Example JDBC URL:

```
jdbc:oracle:thin:@10.0.0.1:1521:ORCL  
jdbc:oracle:thin:@//10.0.0.1:1521/ORCLPDB
```

Using in Data Ingestion Tasks

Real-Time Full Database Sync

Configuration Item	Description
Source Table Scope	Specified tables (only selected tables are synchronized; new tables require a restart) / regex matching (specify database name + table name regex; new tables that match the rule are automatically added)
Read Mode	Full + incremental / Incremental only

Offline Single Table Sync

Parameter	Description
Database / Schema / Table	Required; supports specification and regex matching.
Split Key	Primary key or index columns recommended; used for parallel synchronization.
Filtering Condition	Standard WHERE syntax

For detailed task configuration, see [Create a Single-Table Synchronization Task](#) and [Real-time Whole-Database Multi-Table Task Configuration](#).

Best Practices

- **Account Permission Minimization:** Use a dedicated account and avoid system / sys.
- **Log Retention Time:** Archived logs must be retained for at least 72 hours.
- **Split Key Selection:** Primary key or index column.
- **Performance Considerations:** When LogMiner exceeds 5k/s, it is recommended to use the OGG solution instead.

FAQs

Q: High Latency in Real-Time Sync?

A: LogMiner has performance bottlenecks when the read rate exceeds 5k/s. You can optimize job parallelism and reduce the number of synchronized tables. For high-throughput scenarios, OGG is recommended.

Q: Task Exception After a Column with a Default Value Is Added?

A: LogMiner has issues parsing added columns with default values. It is recommended to add the column first and then UPDATE the default value.

Q: Some Data Not Synced in a RAC Environment?

A: Log file switching may cause LogMiner to fail to detect changes. Increase the log retention time and optimize the archived log configuration.

Q: Other Columns Empty on the Destination After an UPDATE?

A: Supplemental logging is not enabled. Perform the following:

```
ALTER TABLE schema_name.table_name ADD SUPPLEMENTAL LOG DATA (ALL)
COLUMNS;
```

Q: Connection Timeout or Refused?

A: Check network connectivity, listener status, and firewall ports.

References

- [Add Data Source Connection](#)
- [Data Source Connectivity Test](#)
- [Supported Data Sources and Read Capabilities](#)
- [Create a Single-Table Sync Task](#)
- [Real-Time Whole-Database Multi-Table Task Configuration](#)

PostgreSQL Data Source

Last updated: 2026-10-09 16:09:39

Overview

PostgreSQL is a powerful open-source object-relational database system. DataBuddy data ingestion supports using PostgreSQL as a source, covering scenarios such as offline synchronization, real-time full-database synchronization, and sharded database and table synchronization. This document describes its connection configuration, supported capabilities, prerequisites, and common issues.

Supported Versions

Type	Supported Version
Self-Built PostgreSQL	9.6 / 10.x / 11.x / 12.x / 13.x / 14.x / 15.x / 16.x
Tencent Cloud PostgreSQL	11.x / 12.x / 13.x / 14.x / 15.x / 16.x / 17.x / 18.x
JDBC Driver	postgresql-42.2.26

Read Capability

Capability	Offline read	Real-time full database read	Sharded database and table read
Supported	✓	✓	✓

For a complete comparison of data source capabilities, see [Supported Data Sources and Read Capabilities](#).

Use Limits

Real-Time Sync Limitations

Limit	Description
Logical Replication Version Requirement	Native logical replication is supported only in PostgreSQL 10+.
<code>wal_level</code>	Must be set to logical
Replication Slot	Create a Replication Slot.
Table without primary key	Set <code>REPLICA IDENTITY FULL</code> .

Publication	Create a publication to specify the tables to be synchronized.
-------------	--

Offline Sync Limitations

Limit	Description
View	Supports view reading
Split Key	Only integer fields are supported
Table Name with Special Characters	Table names / field names cannot start with a digit or contain uppercase letters, lowercase letters, or hyphens.

Supported Field Types

For complete field types, see [PostgreSQL official documentation](#).

Field Type	Offline read	Real-time read
SMALLINT / SMALLSERIAL / INTEGER / SERIAL / BIGINT / BIGSERIAL	✓	✓
REAL / DOUBLE PRECISION / NUMERIC / DECIMAL / MONEY	✓	✓
BOOLEAN	✓	✓
CHAR / VARCHAR / TEXT / UUID	✓	✓
JSON / JSONB / ARRAY	✓	✓
DATE / TIME / TIMETZ / TIMESTAMP / TIMESTAMPTZ	✓	✓
INTERVAL	✗	✓
BYTEA / BIT / BIT VARYING	✓	✓
CIDR / INET / MACADDR	✗	✗
POINT / LINE / LSEG / BOX / PATH / POLYGON / CIRCLE	✗	✗

Supported DML/DDL for Real-Time Sync

DML

Operation	Whether Supported
INSERT / DELETE / UPDATE	✓

DDL

Operation	Whether Supported
Add table / Add column / Delete column / Modify column type	✓
Delete table / Clear table / Rename table (RENAME TABLE)	✗

PostgreSQL logical replication has limited support for DDL changes. After Schema changes, you may need to reconfigure the synchronization task and publication.

Prerequisites

1. Configuring Account Permissions

Offline synchronization:

```
CREATE USER wedata_user WITH PASSWORD 'your_password';
GRANT CONNECT ON DATABASE your_database TO wedata_user;
GRANT USAGE ON SCHEMA public TO wedata_user;
GRANT SELECT ON ALL TABLES IN SCHEMA public TO wedata_user;
```

Real-time synchronization:

```
CREATE USER wedata_cdc_user WITH REPLICATION PASSWORD 'your_password';
GRANT CONNECT ON DATABASE your_database TO wedata_cdc_user;
GRANT USAGE ON SCHEMA public TO wedata_cdc_user;
GRANT SELECT ON ALL TABLES IN SCHEMA public TO wedata_cdc_user;
```

2. Configuring the WAL Level (Real-Time Sync Only)

```
SHOW wal_level;
```

For TencentDB for PostgreSQL, set `wal_level` to `logical` in Parameter Settings in the console. For self-managed PostgreSQL, modify `postgresql.conf` :

```
wal_level = logical
max_replication_slots = 10
max_wal_senders = 10
```

After the modification, restart the database for the changes to take effect.

3. Configuring pg_hba.conf (Real-Time Sync Only)

```
# TYPE      DATABASE          USER            ADDRESS          METHOD
host        replication      wedata_cdc_user 10.0.0.0/8       md5
```

4. Creating a Publication (Real-Time Sync Only)

```
CREATE PUBLICATION wedata_pub FOR TABLE table1, table2;
-- Or: CREATE PUBLICATION wedata_pub FOR ALL TABLES;
```

5. Setting REPLICA IDENTITY (Tables Without Primary Keys)

```
ALTER TABLE your_table REPLICA IDENTITY FULL;
```

Creating a Data Source

Operation Steps

1. Log in to the [DataBuddy console](#).
2. Switch to the target region and Workspace at the top.
3. Go to **Data Integration > Task Management > Data Source Management** (or **Platform Management > Data Source Management**).
4. Click **Add Data Source** and select **PostgreSQL**.
5. Enter the connection configuration and authentication information. For details, see [Parameter Description](#).
6. Click **Test Connection** (see [Data Source Connectivity Test](#)).
7. After the test passes, click **Save** or **Save & Create Task**.

Parameter Description

Parameter	Description	Required
-----------	-------------	----------

Data Source Name	Unique identifier of the data source within the workspace	Yes
JDBC URL	<code>jdbc:postgresql://host:port/database</code>	Yes
Username.	Database connection account	Yes
Password	Database connection password (SSM credential hosting supported)	Yes
Data Source Version	12 / 13 / 14 / 15 / 16	Yes

Example JDBC URL:

```
jdbc:postgresql://10.0.0.1:5432/mydb
jdbc:postgresql://10.0.0.1:5432/mydb?rewriteBatchedInserts=true
```

Common URL parameters:

Parameter	Description
<code>rewriteBatchedInserts</code>	Optimize batch insert performance (recommended: true)
<code>ssl</code> / <code>sslmode</code>	SSL connection configuration
<code>connectTimeout</code>	Connection timeout (seconds)

Using in Data Ingestion Tasks

Real-Time Full Database Sync

Configuration Item	Description
Source Table Scope	Specified tables (only selected tables are synchronized; new tables require a restart) / regex matching (specify database name + table name regex; new tables that match the rule are automatically added).
Read Mode	Full + incremental / Incremental only

Offline Single Table Sync

Parameter	Description
Database / Schema / Table	Required. Schema defaults to <code>public</code> . Table supports the range <code>table_[0-99]</code>
Split Key	Supports integer fields only; primary key or index columns recommended.
Filtering Condition	Standard WHERE syntax

For detailed task configuration, see [Create a Single-Table Synchronization Task](#) and [Real-time Whole-Database Multi-Table Task Configuration](#).

Best Practices

- **Account Permission Minimization:** Use a dedicated account and grant only the necessary permissions.
- **WAL Log Retention:** `ALTER SYSTEM SET wal_keep_size = '1GB';`
- **Split Key Selection:** Primary key or index column.
- **Network Connectivity:** Use a private network connection whenever possible.

FAQs

Q: Real-time sync reports an error that `wal_level` is not supported?

A: `wal_level` is not set to `logical`.

```
ALTER SYSTEM SET wal_level = 'logical';  
-- Restart the database after the modification.
```

Q: Error indicating the replication slot cannot be created?

A: `max_replication_slots` is too small:

```
ALTER SYSTEM SET max_replication_slots = 20;
```

Q: UPDATE/DELETE cannot be correctly synchronized for tables without primary keys?

A: `REPLICA IDENTITY` is not set:

```
ALTER TABLE your_table REPLICA IDENTITY FULL;
```

Q: Connection refused?

A: `pg_hba.conf` does not allow connections from the DataBuddy network segment. Add the following to the file:

```
host    all             wedata_user      10.0.0.0/8       md5
host    replication     wedata_cdc_user  10.0.0.0/8       md5
```

Q: Tables with special characters in their names cannot be synchronized?

A: Table names and field names cannot start with a digit or contain uppercase letters, lowercase letters, or hyphens.

References

- [Add Data Source Connection](#)
- [Data Source Connectivity Test](#)
- [Supported Data Sources and Read Capabilities](#)
- [Create a Single-Table Sync Task](#)
- [Real-Time Whole-Database Multi-Table Task Configuration](#)

Rest API Data Source

Last updated: 2026-10-09 16:09:43

Overview

A REST API data source can read data exposed by third-party systems through HTTP/HTTPS interfaces, making it suitable for scenarios where data is obtained from the open APIs of business systems. DataBuddy data ingestion supports using REST API as a source, covering offline synchronization scenarios. This document describes its connection configuration, supported capabilities, prerequisites, and common issues.

Supported Versions

Type	Supported Version
REST API	All Versions (HTTP/HTTPS)

Read Capability

Capability	System Support	Description
Offline sync (read/write)	✓ / –	Supports reading data as a source for offline synchronization tasks. The DataBuddy data source supports only read access and does not support writes.

For a complete comparison of data source capabilities, see [Supported Data Sources and Read Capabilities](#).

Use Limits

Limit	Description
Response Format	JSON format data must be returned.
Authentication Method	Supports no authentication / Basic / Token / OAuth2 password mode / OAuth2 client mode

Supported Authentication Methods

Authentication Method	Description
No Authentication	The API does not require authentication.

Basic	Username + Password
Token	Fixed Token
OAuth2 Password Mode	Username + Password for Token Exchange
OAuth2 Client Mode	ClientId + ClientSecret for Token Exchange

Prerequisites

Confirm the API address, request method, and authentication information with the API provider, and make sure that the API is accessible from the DataBuddy runtime environment. The API response must be in JSON format and have a stable field structure.

Creating a Data Source

Operation Steps

1. Log in to the [DataBuddy console](#).
2. Switch to the target region and Workspace at the top.
3. Go to **Data Integration > Access Method > Data Source Management** (or **Platform Management > Data Source Management**).
4. Click **Add Data Source** and select **REST API**.
5. Enter the connection configuration and authentication information. For details, see **Parameter Description**.
6. Click **Test Connectivity** (see [Data Source Connectivity Test](#)).
7. After the test passes, click **Create Only** or **Create and Connect Task**.

Parameter Description

Parameter	Description	Required	Default Value
Data Source Name	Unique identifier of the data source within the workspace	Yes	–
Request URL	Complete API URL	Yes	–
Authentication Method	No authentication / Basic / Token / OAuth2 password mode / OAuth2 client mode	Yes	No Authenti

			cati on
Username / Password	Account for Basic or OAuth2 password mode	Required for the corresponding authentication method	–
Token	Fixed token for Token authentication	Required when the authentication method is Token	–
ClientId / ClientSecret	Credentials for OAuth2 client mode	Required when the authentication method is OAuth2 client mode	–
Token Endpoint / Scope	Token endpoint and scope in OAuth2 mode	Required for OAuth2 authentication	–

Using in Data Ingestion Tasks

Offline Data Read

Configuration Item	Description
Source Connection	A configured REST API data source. Required. Click Test to verify connectivity.
Read Method	Parse data: Parse the JSON content returned by the API and write it to the target table based on field mappings. Raw file: Do not parse the response content. Write the raw data returned by the API to the target as a whole. For regular structured data ingestion, select Parse data.
Request method	HTTP method of the API, POST / GET
Returned data type	Data format of the API response. Currently, JSON is supported.
JSON data path	Use JSONPath to locate the level where records reside in the response, for example, <code>\$.data.list</code> . It must point to an array of records or a single record object. After configuration, the source field JSONPath in field mapping uses a single record as the root node.
Returned data structure	Single record: The response is a single JSON object.

	Array: The response is an array of JSON objects, with each element representing a record. Note that in array mode, each record must be a JSON object. A two-dimensional array will cause parsing to fail.
Request header.	API request headers. Fill them in as prompted on the page, for example, <code>{"Accept-Language": "zh-cn"}</code> . If authentication information is required, such as Authorization, enter it here as well.
Request parameters.	Optional. For GET requests, enter the query parameters appended to the URL, for example, <code>name=wedata&age=10</code> . For POST requests, enter the request body parameters.
Advanced Settings	Click Settings to configure advanced parameters. The default settings meet general scenarios. For large data volumes, split the data into multiple tasks based on the API query parameters and read them separately.

Best Practices

Practice	Recommendation
Authentication Security	Prioritize OAuth2 and avoid plaintext passwords.
Pagination Handling	Confirm the API pagination mechanism to avoid data omission.
Rate Limiting Control	Set request frequency appropriately to avoid triggering API rate limiting.

FAQs

Q: Authentication Failed?

A: Check whether the authentication method and credentials are correct, and confirm that the token has not expired.

Q: Incomplete Data?

A: Confirm that the API pagination parameters are set correctly and that the response is a stable JSON structure.

References

- [Add Data Source Connection](#)

- [Data Source Connectivity Test](#)
- [Supported Data Sources and Read Capabilities](#)

SQL Server Data Source

Last updated: 2026-10-09 16:09:48

Overview

SQL Server is a relational database management system developed by Microsoft, widely used in enterprise applications and data management. DataBuddy data ingestion supports using SQL Server as a source, covering scenarios such as offline synchronization, real-time full-database synchronization, and sharded database and table synchronization. This document describes its connection configuration, supported capabilities, prerequisites, and common issues.

Supported Versions

Type	Supported Version
Self-Built SQL Server	2012 / 2014 / 2016 / 2017 / 2019 / 2022
TencentDB for SQL Server	2012 / 2014 / 2016 / 2017 / 2019 / 2022
JDBC Driver	mssql-jdbc-7.4.1.jre8

Read Capability

Capability	Offline read	Real-time full database read	Sharded database and table read
Supported	✓	✓	✓

For a complete comparison of data source capabilities, see [Supported Data Sources and Read Capabilities](#).

Use Limits

Real-Time Sync Limitations

Limit	Description
CDC Feature	Enable CDC (Change Data Capture) on the database and tables.
Log Retention Period	CDC logs are retained for at least 72 hours.
Table without primary key	Tables without primary keys cannot guarantee exactly-once semantics.
Version Requirement	CDC is available only in Enterprise / Developer / Standard editions.
Lock Contention	Enable snapshot isolation to avoid lock contention.

Offline Sync Limitations

Limit	Description
View	Supports view reading
Split Key	Only integer fields are supported.

Supported Field Types

For complete field types, see [SQL Server official documentation](#).

Field Type	Offline read	Real-time read
BIT / TINYINT / SMALLINT / INT / INTEGER / BIGINT	✓	✓
FLOAT / REAL / DECIMAL / NUMERIC / MONEY / SMALLMONEY	✓	✓
CHAR / NCHAR / VARCHAR / NVARCHAR / TEXT / NTEXT / XML	✓	✓
DATE / TIME / DATETIME / DATETIME2 / SMALLDATETIME / DATETIMEOFFSET	✓	✓
BINARY / VARBINARY / IMAGE / UNIQUEIDENTIFIER	✓	✓
GEOGRAPHY / GEOMETRY / HIERARCHYID / SQL_VARIANT	✗	✗

Supported DML/DDL for Real-Time Sync

DML

Operation	Whether Supported
INSERT / DELETE / UPDATE	✓

DDL

Operation	Whether Supported
Add column / Delete column / Rename column / Modify column type	✗
Add table / Rename table / Delete table / Clear table	✗

SQL Server captures changes based on the CDC mechanism and does not support automatic synchronization of DDL changes. After a Schema change, you need to re-enable CDC and may need to

reconfigure the synchronization task. It is recommended to perform Schema changes during off-peak hours.

Prerequisites

1. Configuring Account Permissions

Offline synchronization:

```
CREATE LOGIN wedata_user WITH PASSWORD = 'your_password';
USE your_database;
CREATE USER wedata_user FOR LOGIN wedata_user;
GRANT SELECT ON SCHEMA::dbo TO wedata_user;
```

Real-time synchronization:

```
CREATE LOGIN wedata_cdc_user WITH PASSWORD = 'your_password';
USE your_database;
CREATE USER wedata_cdc_user FOR LOGIN wedata_cdc_user;
GRANT SELECT ON SCHEMA::dbo TO wedata_cdc_user;
GRANT SELECT ON SCHEMA::cdc TO wedata_cdc_user;
ALTER ROLE db_datareader ADD MEMBER wedata_cdc_user;
```

2. Enabling Database and Table CDC (Real-Time Sync Only)

```
USE your_database;
EXEC sys.sp_cdc_enable_db;

EXEC sys.sp_cdc_enable_table
    @source_schema = N'dbo',
    @source_name = N'your_table',
    @role_name = NULL,
    @supports_net_changes = 1;
```

3. Ensuring the SQL Server Agent Service Is Running

CDC relies on SQL Server Agent Jobs to capture changes:

```
EXEC sys.sp_cdc_help_jobs;
```

4. Enabling Snapshot Isolation (Recommended)

```
ALTER DATABASE your_database SET ALLOW_SNAPSHOT_ISOLATION ON;  
ALTER DATABASE your_database SET READ_COMMITTED_SNAPSHOT ON;
```

Creating a Data Source

Operation Steps

1. Log in to the [DataBuddy console](#).
2. Switch to the target region and Workspace at the top.
3. Go to **Data Integration > Task Management > Data Source Management** (or **Platform Management > Data Source Management**).
4. Click **Add Data Source** and select **SQL Server**.
5. Enter the connection configuration and authentication information. For details, see [Parameter Description](#).
6. Click **Test Connection** (see [Data Source Connectivity Test](#)).
7. After the test passes, click **Save** or **Save & Create Task**.

Parameter Description

Parameter	Description	Required
Data Source Name	Unique identifier of the data source within the workspace	Yes
JDBC URL	<code>jdbc:sqlserver://host:port;DatabaseName=database</code>	Yes
Username.	Database connection account	Yes
Password	Database connection password (SSM credential hosting supported)	Yes
Data Source Version	See Supported Versions	Yes

Example JDBC URL:

```
jdbc:sqlserver://10.0.0.1:1433;databaseName=mydb  
jdbc:sqlserver://10.0.0.1:1433;databaseName=mydb;encrypt=true;trustServerCertificate=true
```

Common URL parameters:

Parameter	Description
<code>databaseName</code>	Database name.
<code>encrypt</code> / <code>trustServerCertificate</code>	Related to encrypted connections
<code>loginTimeout</code>	Login timeout (seconds)

Using in Data Ingestion Tasks

Real-Time Full Database Sync

Configuration Item	Description
Source Table Scope	Specified tables (only selected tables are synchronized; new tables require a restart) / regex matching (specify database name + table name regex; new tables that match the rule are automatically added).
Read Mode	Full + incremental / Incremental only

Offline Single Table Sync

Parameter	Description
Database / Schema / Table	Required. Schema defaults to <code>dbo</code>
Split Key	Only integer fields are supported.
Filtering Condition	Standard WHERE syntax

For detailed task configuration, see [Create a Single-Table Synchronization Task](#) and [Real-time Whole-Database Multi-Table Task Configuration](#).

Best Practices

- **Account Permission Minimization:** Use a dedicated account and avoid `sa`.
- **CDC Log Retention:** Set it to at least 72 hours.

```
EXEC sys.sp_cdc_change_job
    @job_type = 'cleanup',
```

```
@retention = 4320; -- Unit: minutes (72 hours)
```

- **Split Key Selection:** Primary key or index column.
- **Network Connectivity:** Use a private network connection whenever possible.

FAQs

Q: Error Indicating CDC Is Not Enabled or CDC Table Is Not Found?

A: Enable CDC on the database and tables:

```
USE your_database;  
EXEC sys.sp_cdc_enable_db;  
EXEC sys.sp_cdc_enable_table  
    @source_schema = N'dbo',  
    @source_name = N'your_table',  
    @role_name = NULL;
```

Q: CDC Changes Not Captured?

A: The SQL Server Agent service is not started. Start the Agent service and check the CDC job status by running `EXEC sys.sp_cdc_help_jobs;`.

Q: Error Indicating CDC Change Data Not Found?

A: The CDC log retention period is too short, and the logs have been cleaned up. Increase the CDC retention period (see [Best Practices](#)).

Q: Connection Timeout or Refused?

A: Check network connectivity, confirm that SQL Server allows remote connections, and check the firewall port (1433 by default).

Q: Lock Wait or Deadlock During Synchronization?

A: Snapshot isolation is not enabled:

```
ALTER DATABASE your_database SET ALLOW_SNAPSHOT_ISOLATION ON;  
ALTER DATABASE your_database SET READ_COMMITTED_SNAPSHOT ON;
```

References

- [Add Data Source Connection](#)

- [Data Source Connectivity Test](#)
- [Supported Data Sources and Read Capabilities](#)
- [Create a Single-Table Sync Task](#)
- [Real-Time Whole-Database Multi-Table Task Configuration](#)

TDSQL-C MySQL Data Source

Last updated: 2026-10-09 16:09:52

Overview

TDSQL-C MySQL is a cloud-native relational database independently developed by Tencent Cloud. It is 100% compatible with MySQL and provides high throughput and massive distributed storage. DataBuddy data ingestion supports using TDSQL-C MySQL as a source for real-time and offline synchronization. This document describes its connection configuration, supported capabilities, and common issues.

Supported Versions

Type	Supported Version
TDSQL-C MySQL	5.7,8.0
JDBC Driver	8.0.21

Read Capability

Capability	Offline read	Real-time full database read	Sharded database and table read
Supported	✓	✓	✓

For a complete comparison of data source capabilities, see [Supported Data Sources and Read Capabilities](#).

Use Limits

Real-Time Sync Limitations

Limit	Description
Binlog format	Only ROW is supported.
<code>binlog_row_image</code>	Must be set to FULL
Binlog retention period	Retain for at least 72 hours.
Table without primary key	Tables without primary keys cannot guarantee exactly-once semantics.
XA transaction	XA ROLLBACK is not supported.

Read-only standby database	Reading is not supported due to stability and compatibility issues.
Stored procedures and materialized views	Reading is not supported.
SET type	Real-time synchronization is not supported.

Offline Sync Limitations

Limit	Description
Concurrency for Sharded Table Synchronization	When multiple tables are split into a single table during synchronization, the task concurrency must be greater than the number of tables.
Stored Procedure	Reading is not supported.

Supported Field Types

For complete field types, see [MySQL official documentation](#).

Field Type	Offline read	Real-time read
TINYINT / TINYINT UNSIGNED	✓	✓
SMALLINT / MEDIUMINT / INT / BIGINT (including UNSIGNED)	✓	✓
FLOAT / DOUBLE / DECIMAL / NUMERIC	✓	✓
REAL	✗	✓
CHAR / VARCHAR / TEXT Series	✓	✓
JSON	✓	✓
BINARY / VARBINARY / BLOB Series	✓	✓
DATE / TIME / DATETIME / TIMESTAMP / YEAR	✓	✓
BIT / BOOL / BOOLEAN / ENUM	✓	✓
SET	✓	✗
GEOMETRY / POINT	✗	✓
Spatial types such as LINESTRING / POLYGON / MULTIPOINT	✗	✗

`TINYINT (1)` is mapped to `BOOLEAN` by default. You can change it to `TINYINT` in advanced settings.

Supported DML/DDL for Real-Time Sync

DML

Operation	Whether Supported
INSERT / DELETE / UPDATE	✓

DDL

Operation	Whether Supported	Supported syntax examples
Add column (ADD COLUMN)	✓	<code>ALTER TABLE t ADD col TYPE [FIRST AFTER col]</code>
Drop column (DROP COLUMN)	✓	<code>ALTER TABLE t DROP [COLUMN] col</code>
Rename column (RENAME / CHANGE COLUMN)	✓	<code>ALTER TABLE t CHANGE old new TYPE</code>
Modify column type (MODIFY COLUMN)	✓	<code>ALTER TABLE t MODIFY col TYPE</code>
Add table (CREATE TABLE)	✓	CHECK and Temporary tables are not supported.
Rename table / Delete table / Clear table	✗	

DDL change response **takes effect only during the incremental phase**. For high-risk operations such as dropping or renaming tables, the destination typically does not support automatic response.

Prerequisites

1. Configuring Account Permissions

Offline synchronization:

```
CREATE USER 'wedata_user'@'%' IDENTIFIED BY 'your_password';
GRANT SELECT ON your_database.* TO 'wedata_user'@'%';
FLUSH PRIVILEGES;
```

Real-time synchronization:

```
CREATE USER 'wedata_cdc_user'@'%' IDENTIFIED BY 'your_password';
GRANT SELECT, SHOW DATABASES, REPLICATION SLAVE, REPLICATION CLIENT
    ON *.* TO 'wedata_cdc_user'@'%';
FLUSH PRIVILEGES;
```

After `scan.incremental.snapshot.enabled` is enabled, the RELOAD privilege is no longer required.

2. Binlog Configuration (Real-Time Sync Only)

Binlog is enabled by default for TDSQL-C MySQL. In the console, go to Parameter Settings and confirm:

- `binlog_format` = `ROW`
- `binlog_row_image` = `FULL`
- GTID is enabled by default and cannot be disabled.

After modifying the parameters, restart the instance as prompted in the console.

Creating a Data Source

Operation Steps

1. Log in to the [DataBuddy console](#).
2. Switch to the target region and Workspace at the top.
3. Go to **Data Integration > Task Management > Data Source Management** (or **Platform Management > Data Source Management**).
4. Click **Add Data Source** and select **TDSQL-C MySQL**.
5. Enter the connection configuration and authentication information. For details, see [Parameter Description](#).
6. Click **Test Connection** (see [Data Source Connectivity Test](#)).
7. After the test passes, click **Save** or **Save & Create Task**.

Parameter Description

Parameter	Description	Required	Default Value
Data Source Name	Unique identifier of the data source within the workspace	Yes	–

Connection Method	String connection (JDBC URL) / Tencent Cloud instance	Yes	Tencent Cloud Instance
JDBC URL	Connection string in string connection mode, in the format <code>jdbc:mysql://host:port/database</code>	Required for string connection	—
Region and Instance	Required for Tencent Cloud instance mode	Required for Tencent Cloud instance	—
Username.	Database connection account	Yes	—
Password	Database connection password (SSM credential hosting supported)	Yes	—
Data Source Version	5.7	Yes	—
Advanced Parameters	Connection-level extended parameters	No	—

Using in Data Ingestion Tasks

Real-Time Full-Database and Sharded Database/Table Sync

Configuration Item	Description
Source Table Scope	Specified tables (only selected tables are synchronized; new tables require a restart) / regex matching (specify database name + table name regex; new tables that match the rule are automatically added).
Read Mode	Full + incremental / Incremental only
Subscription Type	You can select INSERT / UPDATE / DELETE, and at least one must be retained.
gh-ost Synchronization	Whether to synchronize temporary tables generated by gh-ost online DDL

Offline Single Table Sync

Configuration Item	Description
Data source.	Configured MySQL data source
Database / Table	Supports exact specification and regex matching. Table names support ranges such as <code>table_[0-99]</code>
Split Key	Supports integer fields only; primary key or indexed columns recommended; avoid COLLATE columns.
Filtering Condition	Standard WHERE syntax, supports time parameters (such as <code>gmt_create > \$bizdate</code>).

For detailed task configuration, see [Create a Single-Table Synchronization Task](#) and [Real-time Whole-Database Multi-Table Task Configuration](#).

Best Practices

- **Account Permission Minimization:** Use a dedicated account and grant only the necessary permissions.
- **Binlog Retention Period:** at least 72 hours. A longer period is recommended for full synchronization of large tables.
- **Split Key Selection:** Prefer the primary key or an index column.
- **Network Connectivity:** Use a private network connection whenever possible.

FAQs

Q: server-id Conflict Error?

Multiple synchronization tasks used the same server-id. The system now generates server-id randomly. If an old task has explicitly configured `server-id` in its advanced parameters, delete it.

Q: Binlog File Purged Error?

Increase the Binlog retention period, or optimize job performance to accelerate consumption.

Q: Connection reset Error?

A: Network issue or job backpressure. You can increase `slave_net_timeout = 120` and optimize parallelism and memory.

Q:JobManager OOM?

A: Increase the JobManager CU and adjust the advanced parameter `scan.incremental.snapshot.chunk.size` (default: 8096).

Q: Is gh-ost Online DDL Supported?

A: Yes. After you enable the gh-ost synchronization switch, the system monitors changes to `*_gho` temporary tables. You can also configure a regular expression for shadow tables.

References

- [MySQL Data Source](#)
- [Add Data Source Connection](#)
- [Supported Data Sources and Read Capabilities](#)
- [Create a Single-Table Sync Task](#)
- [Real-Time Whole-Database Multi-Table Task Configuration](#)

Offline Data Synchronization

Creating Single Table Synchronization Tasks

Table-to-Table Synchronization

Last updated: 2026-10-09 16:09:59

Overview

Table-to-table sync is the most common offline sync pipeline: it syncs table data from structured data sources (such as MySQL, PostgreSQL, Oracle, and SQL Server) to target tables in the DataBuddy data lakehouse on a schedule. This document uses MySQL-to-DataBuddy Catalog table sync as an example to describe the configuration process.

Prerequisites

Before you start this operation, make sure the following prerequisites are met:

- You have activated the DataBuddy service and completed workspace initialization.
- The current account has a role related to **data access** in the target Workspace. For details, see [Members and Permission Management](#).
- Compute resources for data ingestion (data ingestion type) have been created.
- A source data source connection has been created and tested. For details, see [Add a Data Source Connection](#).
- The target Catalog has been planned. If the target Schema or table does not exist, it can be automatically created through the one-click table creation feature of this task, which requires the Catalog creation permission.

Use Limits

Limit	Description
Computing Resource	Only Platform Cluster computing resources are supported.
Source Type	See Supported Data Sources and Read/Write Capabilities
Target Type	Tables under the DataBuddy data lakehouse Catalog (Iceberg native tables)
Split Key	Integer fields only; primary key or index columns recommended.
Concurrency Number	When the value is greater than 1, a split key must be configured.

Operation Steps

Step 1: Entering the Task Creation Page

1. Log in to the [DataBuddy console](#).
2. Go to **Data Integration > Task Management** and click **Create Sync Task**.

Step 2: Filling in Basic Information

Enter the task name, sync method, owner, and description. For details, see the general configuration in [Create a Single-Table Sync Task](#). Click **Next**.

Step 3: Configuring the Source

Parameter	Description	Required
Data Source Type	Select the source database type, such as MySQL.	Yes
Data Source Connection	Select a registered data source. In the drop-down list, click Add Data Connection to go to the creation page.	Yes
Source Database	Supports two modes: Select Database and Enter Database Name . Parameter variables can be inserted.	Yes
Source Data Table	Supports two modes: Select Data Table and Enter Data Table Name . Parameter variables can be inserted.	Yes
Add Sharded Databases and Tables	Multiple data sources + tables can be added for the same source type.	No
Split Key	Integer field. Required when concurrency > 1; otherwise, a prompt to complete it will appear upon saving.	Depends on Concurrency Number
Filtering Condition	Standard WHERE condition (without the <code>where</code> keyword); supports time parameters and scheduling parameters.	No
Advanced Settings	Data source-specific parameters (see the corresponding data source section for details)	No

Tip

Data Source Connectivity Test: After you select a data source connection, "To be tested" is displayed. You can click **Test** to initiate a connectivity test. For details, see [Data Source Connectivity Test](#).

Database and Table Selection Mode

Mode	Action	Scenario
Select Database/Table.	Literal exact selection	Synchronize fixed databases and tables.
Enter Database Name/Table Name.	Can be matched through rule expressions or use parameter variables.	Scenarios with multiple sharded tables (such as <code>table_[0-99]</code>) or synchronization based on reading the current day's partition data

Sharding Configuration

Click **Add Sharding** to add multiple data sources of the same source type. Each data source is configured independently with the data source connection, source database, and source table. Added sources can be deleted.

Requirement: The table schemas of all sources must be consistent.

Step 4: Configuring the Target

Parameter	Description	Required	Default Value
Target Catalog	Select Target Catalog.	Yes	–
Target Schema	Specify Schema under the selected Catalog.	Yes	–
Target Table	Specify the target table. If it does not exist, click Create Target Table to create it immediately.	Yes	–
Write Mode	For details, see Write Mode .	Yes	append
Advanced Settings	Target-side extended parameters	No	–

Creating the Target Table

Click the drop-down list, and then click **Create Target Table** to open the table creation dialog box.

1. Select the compute resource for executing the table creation statement (interactive cluster type). If the resource is not started, it will be started automatically after submission.
2. The platform generates a default table creation SQL statement based on the source Schema, which can be adjusted in the editor.
3. Click **Submit**:
 - On success, the dialog box closes and a message appears indicating that the target table \${TargetTableName} was created successfully.
 - On failure, the dialog box remains open and displays the specific failure cause. You can adjust the SQL statement and retry.

Note:

- Some data sources do not support automatic generation of table creation DDL. You can manually write the table creation SQL statement.

Write Mode

Mode	Description
append	Insert mode; write fails when a primary key conflict occurs.
overwrite	Overwrite mode; when a primary key conflict occurs, delete the original row and then insert a new row.
upsert	Update mode; when a primary key conflict occurs, update the specified fields.

Step 5: Configuring Field Mapping

Field mapping is populated by default based on the same-name and same-row rules. The following operations are supported:

Operation	Description
Same Name Mapping	Automatically align fields with the same name in the source/target.
Same Row Mapping	Align in order of row numbers
Clear	Clear all mappings
Pin Mapped	Mapped fields are listed first.
Moving	Drag the move icon before a field to move the row.

Mapping Switch	Click the arrow switch in the Mapping column to toggle the mapping relationship of the row.
----------------	---

Field Operations:

Operation	Description
Add Field	Manually add a field mapping row.
Delete Field	Manually delete a field mapping row.
Adjust Field Type	Adjust the field type. Supported types: field, constant, function, and parameter. When field is selected, you can select the source field type, such as string.

Settings for various field types:

- Field: You can select a source field type, such as "string" or "int".
- Constant: You can enter a constant value as the field name, such as "1" or "true".
- Function: You can enter a database function as the field name, such as "concat(first_name,last_name)" or "sum(amount)".
- Parameter: You can enter a parameter variable as the field name, such as "\${plan_date}". Parameter definitions can be configured in Create a Single-Table Sync Task > Task Parameters.

Step 6: Configuring Runtime Settings

For details, see Create a Single-Table Sync Task > Run Settings.

Step 7: Debugging, Saving, and Publishing

For details, see Create a Single-Table Sync Task > Debug, Save, and Publish.

FAQs

Q: Why does it say "Set a split key first" when a task is saved?

A: When the concurrency is greater than 1, a split key must be configured. Add the split key in the **Source & Target** step, or set the concurrency to 1.

Q: Why does table creation fail with a permission denied error?

A: The current account does not have table creation permission on the target Catalog. Contact the workspace administrator or Catalog owner to grant the **Manageable** permission or the permission required for table creation. For details, see [Member and Permission Management](#).

Q: How do I ensure all source connections pass the connectivity test in a sharding scenario?

A: When multiple data sources are selected, test each connection separately.

Q: Does a debug run write actual data?

A: **Yes**. Debugging and publishing use the same data transfer link and write data to the target based on the current configuration. Before debugging, confirm the target table and write mode to avoid overwriting business data.

Q: How long does it take for task changes to take effect?

A: Saving only generates a new version and does not affect scheduling. You need to click **Publish** to set the current version as the published version. The new version will be used in the next workflow scheduling.

References

- [Table-to-Volume synchronization](#)
- [File-to-table synchronization](#)
- [Create Single-Table Sync Tasks in Batches](#)
- [Offline Synchronization Task Ops Monitoring](#)
- [Workflow Overview](#)

Table-to-Volume Synchronization

Last updated: 2026-10-09 16:10:05

Overview

Table-to-Volume sync writes table data from structured data sources (such as MySQL, PostgreSQL, Oracle, and SQL Server) to DataBuddy Volume in a specified file format. It is commonly used to export processed data as .ok / .ctl files for downstream systems to consume. This document describes the specific configuration for this pipeline. For general steps, see [Create a Single-Table Sync Task](#).

Applicable Scenarios

- Periodically export processed result files to downstream businesses such as SMS, marketing, and external systems (for example, .txt / .csv / .json).
- Data delivery: Export lakehouse data to a specified Volume path in the agreed format.
- File archiving: writes business database data to a Volume partitioned by time for long-term storage.

Prerequisites

Before you start this operation, make sure the following prerequisites are met:

- You have activated the DataBuddy service and completed workspace initialization.
- The current account has a role related to **data access** in the target Workspace. For details, see [Members and Permission Management](#).
- Compute resources for data ingestion (platform cluster type) have been created.
- A source data source connection has been created and tested. For details, see [Add a Data Source Connection](#).
- A Schema of the Volume type already exists in the target Catalog, or the current account has the permission to create one.

Use Limits

Limit	Description
Source Type	Only structured data sources are supported (see Supported Data Sources and Read/Write Capabilities).
Target Type	Volume type Catalog under DataBuddy Catalog
Field mapping	Source can be configured; target cannot be configured (Volume does not perceive field structure).

File Type	Only txt / csv / json are supported.
.ok and .ctl	Marker files supported vary by file type.

Operation Steps

The general process is the same as [Table-to-Table Sync](#). This section only describes the differences in **Target** and **Field Mapping**.

Step 1: Entering the Task Creation Page

See the entry for [Table-to-Table Sync](#).

Step 2: Filling in Basic Information and Source

The source fields (data source type, connection, source database, data table, split key, filter conditions, sharding, and advanced settings) are the same as those in Table-to-Table Sync. See [Table-to-Table Sync > Step 3](#).

Step 3: Configuring the Target

When the target is selected as a **Volume-type Catalog**, the configuration items are as follows:

Parameter	Description	Required	Default Value
Catalog	Select a Volume type Catalog for which you have permission.	Yes	—
Schema	Schema under Catalog	Yes	—
Volume	Select an existing Volume; if it does not exist, create it in the Catalog.	Yes	—
Volume Directory	Subpath within the Volume for file writing; selectable, enterable, supports inserting regular expressions and parameters; using the root path "/" as the subpath is not recommended.	Yes	—
File name.	Name of the file to be written; enterable, supports inserting regular expressions and parameters.	Yes	—
File format.	txt / csv / json; different types have different parsing configurations.	Yes	—
Format Settings	See File Parsing and Inference	Yes	—

Write Mode	Append / Overwrite / Truncate / NonConflict	Yes	Append
Mark Completion File	Generate the .ok file after the file is written.	Yes	Not generated
ctl File	Generate the .ctl file after the file is written.	Depends on file type.	Not generated

Step 4: Format Settings

Click Format Settings to open a dialog box, and configure the write rules based on the selected file type:

TXT / CSV

Configuration Item	Description
Compression Format	gzip / No compression, and so on.
Column delimiter	Field delimiter (such as <code>,</code> <code>;</code> <code>`</code>)
Row delimiter	Row delimiter (such as <code>\n</code> <code>\r\n</code> custom)
Encoding	UTF-8 / GBK, and so on
Control escaping	Escape special characters or not
Quote Character	String delimiter (such as <code>"</code> <code>'</code>)
Include Header	Write field names in the first row or not

JSON

Configuration Item	Description
Compression Format	gzip / No compression, and so on
Encoding	UTF-8 / GBK, and so on
Null Conversion	Whether to convert null to a specified string
Include Header	Write field names in the first row or not

Step 5: Field Mapping

Link	Source field mapping	Target field mapping
Structured data source → Volume	Configurable	Not configurable

- If you do not want a field to be written to a Volume, you can delete the corresponding field mapping row.
- When a field mapping row is added, the field type must be Field. Otherwise, an error will be reported.

Step 6: Running Settings, Debugging, Saving, and Publishing

See [Table-to-Table Sync > Step 4 / 5](#).

Supported File Types and Marker Files

File Type	.ok file	ctl File
Text	Yes	Yes
CSV	Yes	Yes
JSON	Yes	None

Tip

The `.ok` file is used to notify downstream systems that data has been written. The `.ctl` file is used to record data descriptions, such as the number of rows and file list.

Verification Result

After the task is saved and debugged, you can confirm it in the following ways:

1. In [Data Catalog](#), go to the target Volume and view the written files, such as `output_20260101.txt` and `output_20260101.ok`.
2. Download or preview the file content to confirm that the field order, delimiter, and encoding are consistent with the configuration.
3. In [Offline Sync Task Ops Monitoring](#), view the task logs to confirm that no errors exist.

FAQs

Q: What Parameters Are Supported for the Target Volume Path or Filename?

A: System time parameters (such as `${yyyyMMdd}`) and workflow scheduling parameters are supported. During manual debugging runs, if no parameter values are set, the system uses default values. Define them in Run Parameters in advance to ensure that debugging is repeatable.

Q: Will Existing Files Be Deleted from the Target Volume Path When the Write Mode Is Set to Overwrite?

A: Overwrite only overwrites files with the same name and does not delete other files in the path. Append appends content to the end of a file with the same name, depending on the file type.

Q: How Do I Configure Null Value Conversion for JSON Files?

A: When this feature is enabled, all NULL fields are converted to a specified string, such as an empty string `""` or the literal `NULL`. When this feature is disabled, the literal `null` is output according to the JSON standard.

Q: Are Columnar Formats Such as .parquet and .avro Supported?

A: Currently, only TEXT / CSV / JSON formats are supported.

References

- [Table-to-table synchronization](#)
- [File-to-Volume synchronization](#)
- [Create Single-Table Sync Tasks in Batches](#)
- [Offline Synchronization Task Ops Monitoring](#)
- [Unstructured Data Catalog \(File and Volume Management\)](#)

File-to-Table Synchronization

Last updated: 2026-10-09 16:10:10

Overview

File-to-table sync reads semi-structured files (FTP / SFTP / COS / HDFS / REST API, and so on) according to parsing rules and writes them to the target table under DataBuddy Catalog. This document describes the specific configurations for this pipeline. For general steps, see [batch creation of single-table sync tasks](#).

Applicable Scenarios

- Periodically ingest batch files uploaded by business partners to FTP / SFTP / COS, and load them into the data lakehouse for downstream analysis.
- Ingest data provided by third-party SaaS systems through REST API.
- Incrementally sync files partitioned by date in COS.

Prerequisites

Before you start this operation, make sure the following prerequisites are met:

- You have activated the DataBuddy service and completed workspace initialization.
- The current account has a role related to **data access** in the target Workspace. For details, see [Members and Permission Management](#).
- Compute resources for data ingestion (data ingestion type) have been created.
- A source data source connection has been created and tested. For details, see [Add a Data Source Connection](#).
- The target Catalog and Schema have been planned. If the target table does not exist, it can be automatically created by this task with one click, which requires the table creation permission.

Use Limits

Limit	Description
Source Type	Semi-structured data sources (FTP / SFTP / COS / HDFS / REST API, and so on)
Target Type	Tables under the DataBuddy Catalog (Iceberg native tables)
Read Method	Only parse data (write to tables after parsing); to retain the original file structure, use file-to-Volume sync

File path	When no files or only empty files exist in the path, the task runs successfully, but no new data is added to the destination.
Field mapping	The source does not read field metadata. Users need to manually add fields and specify their types.

Operation Steps

The general process is the same as [Table-to-Table Sync](#). This section only describes the differences in **Source** and **Field Mapping**.

Step 1: Entering the Task Creation Page and Filling in Basic Information

See [Table-to-Table Sync](#).

Step 2: Configuring the Source

Source configuration items vary slightly depending on the selected data source type.

Common Fields (FTP / SFTP / COS / HDFS)

Parameter	Description	Required
Data Source Type	Select Source Data Source Type.	Yes
Source Connection	Select a registered data source connection.	Yes
File path	File or folder path. Click the folder icon to open the path selector and drill down level by level. Parameters and regular expressions are supported.	Yes
Read Method	Parse data: parses files into tables; Raw file: writes files to Volume as-is (belongs to file-to-Volume sync).	Yes
File format.	Select Text/CSV/JSON/ORC/Parquet in data synchronization mode.	Yes
Format Settings	Display corresponding fields by file type.	Depends on file type.

Advanced Settings	Extended parameters specific to the data source	No
-------------------	---	----

REST API

Parameter	Description	Required
Data Source Type	REST API	Yes
Data Source Connection	Select a registered API data source.	Yes
Read Method	Parse data: parses structured data content and maps and syncs data content based on field relationships; Raw file: transfers the entire file without content parsing, applicable to unstructured data sync.	Yes
Request method	GET / POST	Yes
Data path	JSON path where the data resides in the API response (for example, <code>data.list</code>)	No
Returned structure	Single JSON object / JSON array	Yes
Request Header	Custom request header	No
Request parameters.	GET query parameters or POST request body parameters	No
Use First Row as Header	Use first row as field names when the returned data is in a table structure.	No
Advanced Settings	Request method: single request / multiple requests; Parameters: pagination parameters and others	No

Step 3: Parsing Configurations (by File Type)

File Type	Key configurations
Text	Column delimiter (custom / <code>;</code> / <code> </code> / Tab), row delimiter (custom / carriage return / line break), first row as field row, escape character (default <code>\</code>)
CSV	Column delimiter (custom / <code>,</code> / <code>;</code> / Tab), escape character (<code>"</code> / <code>\</code>), first row as field name, rows across lines
JSON	Convert time text to time type, allow single quotes, filter comments, rows across lines When JSON nesting exists, nested values are parsed as a text object (not automatically expanded into multiple columns).
ORC / Parquet	No additional configuration required (self-describing format)

Step 4: Configuring the Target

The target-side configuration items are exactly the same as those in [Table-to-Table Sync](#).

- Target Catalog / Schema / Table.
- Write mode: append / overwrite / upsert.

Step 5: Configuring Field Mapping

Link	Source field mapping	Target field mapping
File (parsed) → Table	Configurable (Requires Manual Field Addition)	Configurable

For file-type sources, **field metadata is not read**, so users need to add fields manually.

1. Click **Add Field** and enter the field index value. For example, entering 0 indicates the first element read, and note that the search starts from 0.
2. Select a field type. Supported types include STRING / INT / BIGINT / FLOAT / DOUBLE / DECIMAL / DATE / TIMESTAMP / BOOLEAN.
3. Establish mappings with target fields.

Supported operations: Clear, Pin mapped, Refresh fields.

Step 6: Running Settings, Debugging, Saving, and Publishing

See [Table-to-Table Sync](#).

Verification Result

After the task is saved and debugged, you can confirm it in the following ways:

1. You can see the newly created task in the data integration task list.

2. After the debug run succeeds, the corresponding table under the target Catalog already contains the parsed data. You can verify this by querying in [Data Catalog](#) or SQL Explore.
3. For more fine-grained field type validation, you can run `DESCRIBE` or a simple query in SQL Explore to observe the actually written fields.

FAQs

Q: The uploaded CSV file contains a header, but the mapping result treats the header as a data row?

A: In the CSV/TXT parsing configuration of the source, select **First Row as Field Name**.

Q: How do I expand nested objects in a JSON file into multiple columns?

A: Currently, nested JSON fields are parsed as text objects as a whole and are not automatically expanded. To expand them, you are advised to use functions such as `get_json_object` / `from_json` in an SQL task for post-processing.

Q: How do I manually specify the row tag if XML row tag auto-detection fails?

A: In the parsing settings, enter the correct row tag name in the row tag field. For example, in `<root><record>...</record></root>`, the row tag should be `record`.

Q: Will the task still run if there are no files in the path?

A: Yes, the task runs successfully but no new data is added to the target.

References

- [Table-to-table synchronization](#)
- [File-to-Volume synchronization](#)
- [Create Single-Table Sync Tasks in Batches](#)
- [Offline Synchronization Task Ops Monitoring](#)
- [Upload Local Data to Table](#) (one-time manual upload scenario)

File-to-Volume Synchronization

Last updated: 2026-10-09 16:10:16

Overview

File-to-Volume sync writes semi-structured files (FTP / SFTP / COS / S3 / Azure Blob / HDFS, and so on) to a DataBuddy Volume, supporting two usage modes:

- **Original file transfer:** preserves the original file format and writes files directly to a Volume, commonly used for archiving and backup.
- **Parse and convert format:** reads the source file and re-outputs it according to the target file type, commonly used for format conversion (such as TXT to JSON).

This document describes the specific configurations for this sync link. For general steps, see [Create a Single-Table Sync Task](#).

Applicable Scenarios

- File archiving: writes raw files uploaded by business partners to SFTP into a Volume by date.
- Format conversion: reads the upstream TXT file and writes it back to a Volume in JSON format for downstream consumption.
- Multi-file merge: reads files from multiple source paths and outputs them centrally in a unified format.

Prerequisites

Before you start this operation, make sure the following prerequisites are met:

- You have activated the DataBuddy service and completed workspace initialization.
- The current account has a role related to **data access** in the target Workspace. For details, see [Members and Permission Management](#).
- Compute resources for data ingestion (data ingestion type) have been created.
- A source data source connection has been created and tested. For details, see [Add a Data Source Connection](#).
- A Schema of the Volume type already exists in the target Catalog, or the current account has the permission to create one.

Use Limits

Limit	Description
Source Type	Semi-structured data sources (FTP / SFTP / COS / S3 / Azure Blob / HDFS, and so on)

Target Type	Volume type Catalog under DataBuddy Catalog
Read Method	Original file (Write to Volume as is) / Parse data (Parse and output again based on the target file type)
Field mapping	Original file mode: Neither source / target can be configured; parse mode: Source can be configured (manually add fields), target cannot be configured.
Output File Type	txt / csv / dat / json only

Operation Steps

Step 1: Entering the Task Creation Page and Filling in Basic Information

See [Create a Single-Table Synchronization Task](#).

Step 2: Configuring the Source

Parameter	Description	Required
Data Source Type	Select the source data source type, such as SFTP.	Yes
Data Source Connection	Select a registered data source connection.	Yes
File path	File or folder path. Click the folder icon to open the path selector. Parameters and regular expressions are supported. For S3, include the bucket name. For Azure Blob, include the container name.	Yes
Read Method	Original file or Parsed data (Determines whether to parse)	Yes
File format.	Required for data synchronization mode: .csv / .avro / .json / .parquet / .txt.	Depends on the mode
Parsing Configuration	Required for data synchronization mode. Configure by file type.	Depends on the mode
Synchronization Scope (S3 Only)	Full / Incremental	Depends on the scenario

Advanced Settings	Extended parameters specific to the data source	No
-------------------	---	----

For specific source field differences, see [File-to-Table Sync](#). The source configuration of this pipeline is exactly the same as that of File-to-Table.

Step 3: Configuring the Target

When the target is selected as a **Volume-type Catalog**:

Parameter	Description	Required
Target Catalog	Select a Volume type Catalog for which you have permission. If none exists, click Create New to create one.	Yes
Target Schema	Schema under Catalog	Yes
Volume	Select an existing Volume.	Yes
Volume File Directory	Subpath within the Volume; if left empty, files are written to the Volume root directory.	No (root directory by default)
File name.	Output file name (required only in data synchronization mode)	Depends on the mode
File format.	Required for data synchronization mode: txt / csv / dat / json.	Depends on the mode
Format Settings	Required for data synchronization mode. Configure by target type.	Depends on the mode
Write Mode	overwrite / append / upsert	Yes

Output Format Configuration in Data Sync Mode

File Type	Configuration Item
TXT / CSV / DAT	Compression format, column delimiter, row delimiter, encoding, control conversion, Quote Character, and header included
JSON	Compression format, encoding, null value conversion, and header included

Marker File Support

File Type	.ok file	ctl File
txt	Yes	Yes
csv	Yes	Yes
dat	Yes	None
json	Yes	None

Step 4: Field Mapping

Mode	Source field mapping	Target field mapping
File sync (raw)	✗ Not configurable	✗ Not configurable
Data synchronization (parsed)	✓ Configurable (Add Fields Manually)	✗ Not configurable

Step 5: Running Settings, Debugging, Saving, and Publishing

See [Create a Single-Table Synchronization Task](#).

Verification Result

After the task is saved and debugged, you can confirm it in the following ways:

1. Go to the target Volume in [Data Catalog](#) and view the written files.
2. In file sync mode, the file name is the same as the original file name. In data sync mode, the file name is determined by the target configuration.
3. Download or preview the file to confirm its content. If there is an .ok file, confirm that the content marker complies with downstream conventions.

FAQs

Q: Can I Change the Read Mode If I Selected the Wrong One?

A: Yes. On the task editing page, switch the Read Mode from Raw File to Parsed Data, fill in the required configuration items, and save. Note: Switching resets the existing parsing configuration and field mappings.

Q: Will Multiple Source Files Be Merged into One When the Read Mode Is Set to "Original File"?

A: No. In file sync mode, source files are written to the Volume in a 1:1 mapping with the original structure, retaining the original file names without merging.

Q: Will Files with the Same Name Be Overwritten in Data Sync Mode?

A: It depends on the write mode:

- **overwrite**: Overwrites files with the same name.
- **append**: Appends content to the end of a file with the same name.
- **upsert**: The behavior is similar to overwrite, as the Volume does not perceive primary keys.

Q: What Happens If Files with the Same Name Are Selected from Different Paths on the Source?

A: An error will be reported. Logically, during reading, all files are collected into a list, and an error is reported if duplicate file names exist.

Q: Can a Task Retain Original Files and Generate Parsed Files at the Same Time?

A: No. A task can only use one sync mode. To archive and transform data at the same time, create two tasks and orchestrate them through [Workflow](#).

References

- [Create a Single-Table Sync Task](#)
- [File-to-table synchronization](#)
- [Table-to-Volume synchronization](#)
- [Create Single-Table Sync Tasks in Batches](#)
- [Upload Local Files to a Volume](#)

Batch Creating Single-Table Sync Tasks

Last updated: 2026-10-09 16:10:20

Overview

Batch creation of single-table sync tasks allows you to generate multiple independent single-table sync tasks for multiple source tables at once, avoiding the repetitive effort of creating them one by one. It is commonly used to ingest a large number of structured tables into the lake at once, or to periodically load multiple unstructured file paths into the target Catalog / Volume.

The structure of each generated single-table task is exactly the same as that of [Create Single-Table Sync Task](#), and each task can be independently edited, debugged, published, and scheduled. The essence of a batch task is to configure once and generate N single-table tasks in batches.

Applicable Scenarios

- Batch ingestion from business databases: 100 business tables in MySQL are synced to the lakehouse at once.
- Multi-shard same-source sync: tables with the same name from multiple shards (such as db_0 / db_1 / ... / db_n) are merged into the target table.
- Batch sync from multiple file paths: files with the same structure in multiple directories on FTP are loaded into Volume in batches.

Prerequisites

Before you start this operation, make sure the following prerequisites are met:

- You have activated the DataBuddy service and completed workspace initialization.
- The current account has a role related to **data access** in the target Workspace. For details, see [Members and Permission Management](#).
- Compute resources for data ingestion (platform cluster type) have been created.
- A source data source connection has been created and tested. For details, see [Add a Data Source Connection](#).
- The target Catalog has been planned. If the target Schema or table does not exist, the batch task can automatically create the table during the creation process, which requires the Catalog creation permission.

Use Limits

Limit	Description
-------	-------------

Source Type	Structured data sources (same as single-table tasks) + unstructured files (FTP / SFTP / COS / S3 / Azure Blob / HDFS / Rest API)
Target Type	Table (structured scenario) / Volume (file scenario)
Disabled Target Configurations	After the target Catalog is selected, the target Schema, Table (or Volume / Path / File name) are specified item by item through table mapping, and the target form displays "Set in mapping".
Single-task Concurrency	Channel settings apply to each single-table task, equivalent to batch setting the properties of each single-table task.

Operation Steps

Step 1: Entering the Task Creation Page

1. Log in to the [DataBuddy console](#).
2. Switch to the target region and Workspace at the top.
3. Go to **Data Integration > Task Management** and click **Create Sync Task**.
4. Select **Sync Method: Offline Batch Sync**.

Step 2: Filling in Basic Information

Parameter	Description	Required	Default Value
Task Prefix	Prefix used to generate single-table task names. Final single-table task name format: <code>\${Task Prefix}_\${Target Table Name}</code>	Yes	–
Access Mode	Batch offline single-table ingestion	Yes	–
Owner	Applied to all generated single-table tasks by default	Yes	Current Creating User
Task Description	Applied to all generated single-table tasks by default	No	–

Step 3: Configuring the Source

Structured Scenarios

Parameter	Description	Required
Source Connection	Same as single-table tasks	Yes
Select Source Table.	Supports selecting multiple tables; supports search by database/table; displays the structure of selected database tables; can be deleted item by item.	Yes
Split Key	Applied to each single-table task by default	No
Filtering Condition	Applied to each single-table task by default	No
Advanced Settings	Applied to each single-table task by default	No

Unstructured File Scenarios

Parameter	Description
Source Connection	Same as single-table tasks
Read Method	Data sync / file sync (same as single-table)
File format.	Files will be parsed in the specified file format.
Parsing Configuration	Same as single-table tasks; applied to each single-table task by default.
Advanced Settings	Applied to each single-table task by default.

Step 4: Configuring the Target

The target Catalog can be a table type or a Volume type. Configure it based on the target Catalog type.

Select the target Catalog as the table type

Parameter	Description	Required
Schema matching policy	Select the Schema matching policy for the sync target. Options: Same as source database / Custom. For Custom, configure it through matching rules.	Yes
Table matching	Select the table matching policy for the sync target. Options: Same as source table / Custom. For Custom, configure it through matching rules.	Yes

policy		
Write Mode	append / overwrite / upsert	Yes
Advanced Settings	Applied to each single-table task by default	No

Select the target Catalog as the volume type

Parameter	Description	Required
Write Mode	append / overwrite / upsert	Yes
File format.	Files will be written in the specified file format.	Yes
Format Settings	The written file will be assembled in the specified format.	Yes
Mark Completion File	Generate the .ok file after the file is written.	Yes
CTL File	Generate the .ctl file after the file is written.	Depends on file type.
Advanced Settings	Applied to each single-table task by default	No

Step 5: Configuring Table Mapping

Click "Refresh Source and Target Table Mapping" to refresh the mapping between source and target tables. Single-table tasks will be generated based on this mapping later.

The table mapping area is dynamically displayed based on the target Catalog type.

Target as a Table-Type Catalog

Field	Description
Task name / source database / source table	Read-only, follows the source configuration
Target catalog	Specify under Catalog
Target Schema	Display the target Schema name based on the schema matching rule configured on the target. If the Schema does not exist, a red "!" prompt is displayed, and you can choose to replace it or create one.

Target Table	Display the target table name based on the table matching rule configured on the target. If the table does not exist, a red "!" prompt is displayed, and you can choose to replace it or create one.
Field mapping rule	Same Name / Same Row Mapping
Computing Resource	Default unified resource for tasks

Target as a Volume-Type Catalog

Field	Description
Source	You can click Add a Row to generate a single-table task.
Target Schema	Specify under Catalog. You can select an existing one or create a new one.
Target Volume	Specify under Schema. You can select an existing one or create a new one.
Volume Path	Subpath within the Volume
File name.	When the path endpoint is a file, the file name variable is automatically displayed by default.

Table Mapping Operations

Operation	Description
Batch Operations	Configure field mapping rules, configure compute resources, and remove selected rows.
Show unconfigured only.	Filter out rows that have not been configured with targets.
Configure target schema in batches.	Configure target schema in batches.
Configure target volume in batches.	Configure target volume in batches.
List operation – Edit	Open the single-table task editor (for details, see Single-Table Overwrite).
List operation – Remove	Remove this mapping entry.

Step 6: Overwriting a Single Table

The settings are the same as those for a single-table task. For details, see [Create a Single-Table Task](#). Some settings cannot be modified.

Step 7: Configuring Runtime Settings

For details, see [Create a Single-Table Synchronization Task – Step 4](#).

Step 8: Saving and Creating a Single-Table Task

Click **Start Batch Creation**. The system returns to the task list page, and a **progress pop-up** appears in the lower-right corner of the page:

Field	Description
Batch Task Status	Creating / Completed / Terminating / Terminated
Batch Task Name	Display batch task prefix.
Progress Metric	Number of successful tasks / Number of exceptions / Total number of tasks
Deleting an alarm policy	Delete a batch task to remove the batch task, but generated single-table tasks are not affected.

Click the "Task List" area to view the creation details of a specific single-table task:

Field	Description
Task Name	If the task is created successfully, the single-table task name is displayed; otherwise, "-" is displayed.
Source / Target	Source table / target table
Status	Displayed under the All tab; switch by All / Succeeded / Failed.

Verification Result

After the batch task is completed, you can confirm in the following ways:

1. The progress pop-up shows **Completed** and the number of successful tasks = the total number of tasks.
2. Go to **Data Ingestion > Task Management** task list, search by task prefix, and view the generated single-table tasks.
3. Open any generated single-table task. Its configuration items are the same as those of the batch task, and it can be debugged, published, and scheduled independently.

FAQs

Q: Can a single-table task still be edited independently after a batch task is created?

A: Yes. Generated single-table tasks have the same status as manually created single-table tasks and can be independently edited, debugged, published, and deleted. Modifying a single-table task does not affect the batch task record.

Q: After a single table is overwritten, will modifying global settings in the batch task overwrite it again?

A: A batch task takes effect immediately after it is saved. Generated single-table tasks will not be overwritten by subsequent configuration changes to the batch task. To update them uniformly, edit each single-table task individually.

Q: What data sources are supported by batch tasks?

A: Same as single-table tasks. Structured data sources support batch table synchronization, while unstructured data sources support batch synchronization of multiple paths (see [Limitations](#)).

Q: How does field mapping in a batch task handle multiple heterogeneous tables?

A: Field mappings are not configured at the batch level. Each single-table task is configured independently. If table structures differ significantly, split them into multiple batch tasks to process groups of tables with the same structure separately.

References

- [Table-to-table synchronization](#)
- [Table-to-Volume synchronization](#)
- [File-to-table synchronization](#)
- [File-to-Volume synchronization](#)
- [Offline Synchronization Task Ops Monitoring](#)

Ops Monitoring for Offline Sync Tasks

Last updated: 2026-10-09 16:10:25

Overview

Ops monitoring for offline synchronization tasks is organized around the following workflow: Task List → Version Management → Publishing → Permission Authorization → Running Instance Viewing. This document describes the entry points and procedures for various Ops operations, helping you view task status, roll back versions, control publishing pace, and diagnose runtime issues in daily use.

Tip

The actual **running instances** of offline sync tasks are generated through [Workflow](#) scheduling. Running details and logs can be viewed in the instance view of the Workflow task node.

Entries

Entries	Path
Task list	Data Integration > Ingestion task > Offline
Task editing page	Click Edit in the Operation column on the task list page, or click the task name.
Task Instance (Run Details)	Workflow > Workflow details > Task node

Task List

The task list displays all offline sync tasks that the current user **has permission to access**.

Default Display Fields

- Task name (click to go to the edit page).
- Publish version.
- Creator.
- Last modified by.
- Last modification time.

Complete Optional Fields

You can customize the displayed fields by clicking the settings icon on the right side of the list header to open the field selection dialog. The complete list of fields is shown in the following table:

Field	Description
Task Name	Single-table synchronization task name
Task ID	Unique task ID
Published Version	Currently published version number
Release time.	Last published time
Creator.	Creator username
Creation Time	Task creation time.
Owner	Actual task running identity
Last Modified By	User who last saved the task
Last modification time.	Last saved time

i Tip

The customized display field settings are saved locally in the browser and remain effective the next time you access the page.

List Operations

Operation	Description
Publish	Trigger task release check
Editing an alarm policy	Go to the task editing page.
Permission	Configure task-level ACL permissions. For details, see Members and Permission Management .
Deleting an alarm policy	Delete Task (Irreversible)

Batch Operations

The task list supports batch publishing after multiple selection:

1. Select one or more tasks.
2. Click **Batch Publish**, and a publish check dialog will pop up.
3. Switch to view the check results of each task.

4. Select **Recheck** / **Continue Publishing** / **Ignore Exceptions and Publish**.

Version Management

Historical Versions

Each time a user clicks **Save**, a historical version is generated. A maximum of 200 versions are retained. If this limit is exceeded, the earliest unpublished versions are automatically deleted.

Go to the task editing page and click the **Version Management** icon in the upper-right corner to view the version list:

Field	Description
Version name	Consists of the save time in the format "month day hour minute second".
Creator.	User who saved
Label	Published version displays "Published".

Creating a Version

Operation	Action
Save	Generate a new historical version; prompt "Task saved successfully"/"Task save failed: \${failure reason}".
Save Only	Same as above.
Create and Publish / Save and Publish	Same as above.

Deleting a Version

Operation options:

Operation	Description
Delete a single version	Hover over a historical version and click Delete to remove it. At least one version must be retained.
Clear	After confirmation, clear all versions (retain the released version and the latest version).

 **Warning**

Delete Published Version triggers an additional confirmation: "You are deleting a published version. This operation may cause exceptions in modules associated with this task." If you continue, the task publishing status changes to **Unpublished**, and workflows that reference this task will fail with the message "Task not published."

Version Rollback

Version rollback is the process of copying a version, editing it, and publishing it:

1. Click a version in the historical version list to return to that version. The editing page then switches to the configuration of that version, and a temporary editing version is generated based on it.
2. Click **Save** to generate the latest saved version.
3. Click **Publish** to save and publish, generate a new version, and point the published version to that version.

Release Process

Single-Table Task Publishing

Entry: Click **Publish** in the operation area of the task list or task editing page.

Publishing process:

Click Publish to open the publish check dialog.

└ All passed: Select "Recheck" or "Continue to Publish".

└ Partial exceptions: Select "Recheck" or "Ignore Exceptions and Publish".

Publish check items are consistent with debugging. For details, see [Create a Single-Table Sync Task – Debug Run](#):

Phase	Check Item
Task Saving	Serialization and Persistence
Configuration Check	Source / Target / Mapping
Data Source Check	Source/Target Connectivity, Source Permissions
Resource Check	Resource Status

Publish completion message:

- Success: The message "Task published successfully" is displayed.
- Failure: The message "Task publishing failed: \${failure reason}" is displayed.

Publishing version rules:

- The publish operation sets the current edited version as the published version.
- A task **retains only one published version**: A subsequent publish replaces the old published version with the new version.
- The "Published" indicator in the list moves with the current published version.

Batch Task Publishing

After selecting multiple tasks in the task list, click **Batch Publish**:

1. A publish check dialog pops up, displaying the check status of each task in the list.
2. You can view by status: Passed, Abnormal, In Progress.
3. The progress bar displays the number of passed and abnormal items.
4. Select **Recheck** / **Continue Publishing** / **Ignore Exceptions and Publish**.

Permission Management

Task-level ACL permissions are based on the platform's unified permission model. For details, see [Members and Permission Management](#).

Permission Items

Permission	Operation scope
Manageable	View / Edit / Run / Delete / Grant permissions
Runnable	View / Edit / Run
Editable	View / Edit
Viewable	You can only view items.

Tip

The **Runnable** permission also controls debug runs: Only users with the **Runnable** permission or higher can debug tasks.

Default Permissions

- Workspace administrators have the **Manage** permission by default, and this permission cannot be revoked.
- Task owners have all permissions by default, and these permissions cannot be revoked.

Operation Entry

Entries	Path
---------	------

Task list

In the operation column, choose **More > Permissions**

Viewing Running Instances

Tip

Offline sync tasks are triggered to run through [Workflow](#). The entry for viewing task instances is: Workflow > Workflow Details > Task Node.

Data overview

The instance run details provide the following metrics:

Type	Field
Read	Read Size, Record Quantity
Write	Write Size, Number of Records
Test Duration	Execution Duration, Speed, Dirty Data

Reading Node Details

Field	Description
Data source.	<code>Database.table</code>
The total number of entries	Total records read
Total bytes	Total bytes read
Speed	Rows/s
Throughput	Bytes/s
Waiting time for read	Blocked waiting time
Number of dirty data records	Number of dirty data records that failed to parse or exceeded the threshold

Writing Node Details

The fields are the same as those for the read node, but the described object is the target table: Target Table, Total Records, Total Bytes, Speed, Throughput, Read Wait Time, and Dirty Data Records.

Instance Operations

The following operations are supported in the workflow task instance details:

Operation	Trigger Conditions
Running	The instance has not started.
Terminate	The instance is running. Stop the instance after secondary confirmation.
Retry	The instance has failed.

Run Logs

The entry is the same as that for run instances: In the Workflow task instance details, switch to the **Run Log** tab.

Function	Description
Switch Logs	Multiple executions of an instance (for example, retries) generate multiple log records. The records are named with the start and end time, and can be viewed one by one.
Log Content	Display complete running logs in text format
Download	Click Download in the upper-right corner to export the logs as a .log file.

Deleting a Task

Entries

Entries	Path
Task list	In the operation column, choose More > Delete

Prerequisites for Deletion

The current account has the **Manage** permission on the task or is the task owner.

Deletion Behavior

1. After Delete is clicked, a confirmation dialog appears.
2. After confirmation, the task is deleted (actually a soft delete, which facilitates exception recovery).
3. Assets already completed through this task (such as synced data and created target tables) are **not affected**.

4. When an entity that references this task (such as a workflow) is executed, the error "Task deleted" is reported.
5. Deletion operations are recorded in CloudAudit logs (to be supplemented) for traceability.

FAQs

Q: Why Can't I See the Task I Created in the List?

A: Check the following:

1. Whether the currently selected Workspace is the Workspace where the task was created.
2. Whether the filter conditions (such as creator and last modified time) limit the display range.
3. Does your account have the View permission on this task? For details, see [Members and Management Permissions](#).

Q: When Will Scheduled Workflows Use the New Version After a Version Is Released?

A: The latest published version is used in the next workflow scheduling. Running instances do not switch to the new version.

Q: Can a Task Be Recovered After Deletion?

A: The platform uses soft deletion to facilitate exception recovery, but **does not provide an external recovery entry**. To restore data, submit a ticket to contact technical support.

Q: If Some Tasks Fail During a Batch Release, Will Other Tasks Be Affected?

A: No impact. Each task is detected and published independently. In the progress dialog, you can click **Re-detect** or **Ignore Exception and Publish** for failed tasks.

Q: How to Quickly Identify the Cause of an Instance Failure?

A: In the Workflow task instance details:

1. In Data Overview, check whether the "Dirty Data Volume" exceeds the threshold.
2. Switch to the **Run Log** tab to view detailed errors.
3. Click **Download Logs** to export the .log file for further analysis or feedback to technical support.

References

- [Create a Single-Table Sync Task](#)
- [Create Single-Table Sync Tasks in Batches](#)
- [Workflow Overview](#)
- [Workflow and Task Run Monitoring](#)

- [Member and Permission Management](#)

Real-Time Data Synchronization

Real-Time Whole-Database Multi-Table Task Configuration

Last updated: 2026-10-09 16:10:32

Overview

A real-time whole-database multi-table task synchronizes **one or more tables** from a source database to the DataBuddy data lakehouse in real time through CDC links. A single task configuration can cover the entire database or multiple specified tables, supporting automatic table creation, DDL synchronization, and automatic table addition or removal during runtime. This document describes the complete configuration process of the task.

Prerequisites

Before you start this operation, make sure the following prerequisites are met:

- You have activated the DataBuddy service and completed workspace initialization.
- The current account has data access permissions in the target Workspace. For details, see [Members and Permission Management](#).
- A source **data source connection** has been created and passed the connectivity test. For details, see [Add a Data Source Connection](#).
- A **real-time cluster** compute resource for real-time synchronization has been created. For details, see [Create a Compute Resource](#).
- The source has enabled CDC capabilities, such as setting MySQL Binlog to ROW, Oracle LogMiner, and SQL Server CDC. For specific requirements, see the sections for each data source.
- The target Catalog has been planned. To enable automatic table creation, the current account must have write and table creation permissions on the target Catalog.

Use Limits

Limit	Description
Source Type	MySQL, Tencent Cloud MySQL, TDSQL-C MySQL, PostgreSQL, Oracle, SQL Server, Kafka, CKafka, MongoDB, Tencent Cloud MongoDB
Target Type	Iceberg native tables under the DataBuddy data lakehouse Catalog
Computing Resource	Only real-time cluster computing resources are supported.

Resource consumption	Single-task range: 1 to 256 CUs, with a step size of 1 CU
Maximum Number of Source Tables	A single task can select up to 5,000 tables.
Automatic table creation	When Kafka is used as the source, automatic table creation is not supported. Tables must be created in the destination in advance.
Mapping Match Preview	Kafka and MongoDB do not support mapping match preview.
Automatic table addition and removal	Only the regex matching mode supports automatically adding newly matched tables during running. The specified tables mode requires pausing the task before manually adding or removing tables.

Operation Steps

A real-time whole-database multi-table task uses a wizard-based configuration, which includes five steps: **Basic Information** → **Source** → **Target** → **Data Verification** → **Runtime Settings**.

Step 1: Entering the Task Creation Page

1. Log in to the [DataBuddy console](#).
2. From the workspace list, go to the target workspace.
3. Go to **Data Integration > Ingestion Tasks**, and click **Create Task**.
4. After selecting the data source type, select **Real-time Whole-Database Multi-Table Ingestion** as the ingestion method.

Step 2: Configuring Basic Information

After filling in the basic task information, click **Next**. For parameter details, see [Basic Settings](#).

Step 3: Configuring the Source

1. Select a data source connection to run a connectivity test. If the test fails, click **Retry** to view the specific cause.
2. Select the **source table**:
 - **Specified Tables**: Filter by database name and table name, and then select the specific tables.
 - **Regex-Matched Tables**: Enter regular expressions for database names and table names. New tables that match the expressions during runtime are automatically ingested.
1. Select the **read mode**: **Full and Incremental** (default) or **Incremental Only**.
2. Keep at least one of the following in **Subscription Type**: Insert, Update, or Delete.

3. To configure advanced behaviors such as setting sharding parameters and controlling metadata field writes, expand **Advanced Settings**.

For parameter details, see [Source Settings](#).

Tip

Configurations vary by data source type. For details, see the documentation for each data source type listed under Data Sources, such as [Kafka Data Source](#).

Step 4: Configuring the Target

1. **Basic Configuration:** Select the target Catalog and system fields.
2. **Table Creation Configuration:** Currently, only Iceberg native tables are supported. For Kafka sources, this section is disabled and tables must be created manually.
3. **Mapping Match Preview:** Check the mapping relationship between each source table and its target table. You can click **Edit** to adjust field mappings, target Schema/table, and synchronization primary keys.

For parameter details, see [Target Settings](#).

Note:

After field mappings are modified, DDL changes no longer support adding columns, dropping columns, renaming columns, or modifying column types. If such changes occur in the source table, the task may report an error. Use this only when differentiated mappings are truly needed.

Step 5: Data Reconciliation (Optional)

Configure data reconciliation rules as needed. For detailed configuration items and limitations, see [Real-Time Synchronization Task Data Reconciliation](#).

Step 6: Configuring Runtime Settings

1. Configure the **DDL Message Processing Policy**: determines whether to automatically synchronize, ignore, or stop the task when a DDL message is received from the source. (Some data sources do not have this configuration. The actual page prevails.)
2. Configure the **Task Running Policy**: Checkpoint interval, write exception policy, and maximum restart attempts.
3. Configure **Alarm Notification**: Select an existing alarm channel and enable monitoring metrics. For detailed rules, see [Alarm Notification](#).
4. To adjust pass-through parameters, expand **Advanced Settings**.

Step 7: Saving or Publishing

Operation	Action	Task Status
-----------	--------	-------------

Save only	Save configuration without publishing	Unpublished
Save and publish	Save and submit for publishing	Pending start

After the operation is completed, the page returns to the task list. For subsequent Ops operations such as start, restart, and stop, see [Real-Time Synchronization Task Ops Monitoring](#).

Parameter Description

Basic Settings

Parameter	Description	Required	Default Value
Access Type	Select Source Type	Yes	–
Access Mode	Real-time whole-database multi-table ingestion / Real-time sharded database and table ingestion	Yes	Real-time whole-database multi-table ingestion
Task Name	Uppercase and lowercase letters, Chinese characters, digits, and underscores, with a length not exceeding 256.	Yes	<code>\${Data Source Type}_YYYYMMDD_HHmms</code>
Computing Resource	Real-time cluster computing resources used for real-time synchronization	Yes	–
Resource consumption	Number of CUs, ranging from 1 to 256, with a step size of 1.	Yes	–
Owner	Account used to run the task; must have access permissions on the target end.	Yes	Current Creating User
Task Description	No more than 200 characters	No	–

Source Settings

Parameter	Description	Required
-----------	-------------	----------

Data source.	Data source connection of a matching type in the current Workspace	Yes
Source Table	See Two Selection Modes for Source Table	Yes
Read Mode	Full + incremental / Incremental only	Yes
Subscription Type	Insert / Update / Delete , keep at least one.	Yes
Advanced Settings	Advanced parameters specific to the data source	No

Two Selection Modes for the Source Table

Mode	Action	Automatic table addition and removal during running
Specified Tables	Filter by database name and table name, then select; up to 5,000 tables.	✗ Pause the task before manual editing.
Regex-matched Tables	Enter the database and table regex; new tables that match during the running period are automatically added.	✓ Automatic addition and removal; added or removed tables must be within the regular expression scope.

Target Settings

Complete the basic configuration

Parameter	Description	Required
Data Location	Select the target Catalog. Local creation is supported (requires Catalog creation permission).	Yes
Schema matching policy	Control the naming rule of the target Schema.	Yes
Table matching policy	Control the naming rule of the target table.	Yes
System Fields	See System Fields (Metadata Fields)	No

System Fields (Metadata Fields)

Built-in fields that can be selected and written to the target table:

Field Name	Type	Description
<code>datasource_wedata_di</code>	string	Data Source Name
<code>database_wedata_di</code>	string	Database name.
<code>table_wedata_di</code>	string	Data Table Name
<code>sequence_id_wedata_di</code>	string	Incremental event record ID, unique and incrementing.
<code>type_wedata_di</code>	string	Operation Type
<code>ts_wedata_di</code>	timestamp	Log Time
<code>processing_time_wedata_di</code>	timestamp	Processing time (machine local)
<code>before_image_wedata_di</code>	string	Whether it is a pre-update record, with values <code>Y</code> / <code>N</code>
<code>after_image_wedata_di</code>	string	Whether it is a post-update record, with values <code>Y</code> / <code>N</code>

 **Note:**

If you deselect a system field after the task has been published and is running, **the field in the target table will not be deleted**, but subsequent data will be written as null. Confirm this before saving.

Table Creation Configuration

Parameter	Description
Table creation type	Currently, only native tables (Iceberg) are supported.
Field mapping	The platform populates field type mappings based on default rules, and supports modifications as needed.

Mapping Match Preview

Preview the mapping relationship between each source table and its target table. The columns include Sequence Number, Source Database, Source Schema (if any), Source Table, Target Schema, Target Table, Synchronization Primary Key, Table Creation Method, Field Mapping, and Operation.

Item	Description
Synchronization primary key	Auto-fill: use the primary key of an existing table to fill the primary key of this table; for auto-created tables, fill the primary key of the table to be created. If no primary key exists, select 1+ columns as a joint substitute primary key.
Field mapping	Default: Default mapping ; after Edit is clicked and modifications are made, it changes to Custom mapping (DDL change capability is limited).
Operation (remove)	Available only in specified tables mode; the remove button is grayed out in regex matching mode.
Operation (edit)	Open the right drawer to display and allow adjustment of field mappings based on the target DataBuddy type.

Preview is not supported when the data volume is too large. You can view the actual link details on the Ops details page after the task is published and running.

Running Settings

DDL Message Processing Policy

Processing method when a DDL change is received from the source (for the DDL types supported by different links, see the corresponding data source sections, such as [MySQL Data Source](#)):

Policy	Action
Automatic synchronization	Automatically execute the corresponding DDL on the destination.
Ignored.	Skip DDL; subsequent data may fail due to Schema mismatch.

Task Execution Policy

Parameter	Description
Checkpoint Interval	State snapshot write interval
Write Exception Policy	Handling method for a single write failure
Maximum Restart Attempts	Maximum number of automatic restarts after task exceptions

Alarm Notification

Integration with [notification channels](#) is supported. After a channel is added in the basic platform, it automatically appears in the options.

Alarm Type	Description	Key threshold
Task failed.	Trigger an alarm when a task enters the failed state	–
Service Latency	Synchronization link latency exceeds the threshold.	Latency threshold (seconds), duration (minutes)
Number of Failovers	The number of restarts within the observation window exceeds the threshold.	Observation window (minutes), number of restarts (times)
DDL Notification	Notify when the selected DDL type occurs.	Multiple types: Add column, Delete column, Rename column, Modify column type, Delete table, Truncate table, Rename table

Parameter	Description	Value Range	Default Value	Note.
Latency threshold (seconds)	Lower latency threshold for service latency alarms	1 to 86400, with a step size of 1	120	Recommended to be at least 2 times the Checkpoint interval.
Duration (minutes)	Alarm after the latency threshold is continuously exceeded for a specified duration	1 to 60, with a step size of 1	5	A value that is too short may cause false alarms due to network jitter.
Observation window (minutes)	Failover Statistics Window	5 to 60, with a step size of 1	10	Recommended range: 10 to 30 minutes.
Number of restarts (times)	Alarm threshold for the number of restarts within the observation window	1 to 20, with a step size of 1	3	Recommended to be less than the upper limit of the number of automatic restarts.

For alarms related to data reconciliation, see [Real-Time Synchronization Task Data Reconciliation](#).

Task Version Management

You can view the version in the upper-right corner of the task editing page:

- **Save** to generate a version immediately, recording the save time and the user who saved it.

- Click a specific version to enter read-only mode. Click **Return to This Version** to switch to edit mode, and a new version is generated after you edit and save.
- The version of a running task is marked with a **Production** label.
- Changes made through the Modify Owner and Modify Alarm entries in the task list do not generate a new version.

Verification Result

After the operation is completed, you can confirm in the following ways:

1. You can see the task in the **Data Integration > Access Tasks > Real-Time** list, and its status is as expected:
 - **Unpublished:** Save only.
 - **Pending Start:** Published. You can click **Start** to enter the running state.
1. After the task is started, its status changes from **In Progress** to **Running**. Click the task name to go to the Ops details page and view the running progress. For more information, see [Real-Time Synchronization Task Ops Monitoring](#).
2. View the automatically created target table under the target Catalog. Data is written after the full phase is completed, and changes are continuously appended during the incremental phase.

FAQs

Q: A New Table Was Added at the Source, but the Running Task Did Not Automatically Ingest It?

A: Only in **Regex-Matched Tables** mode are new tables that fall within the regex scope automatically ingested. In **Specified Tables** mode, you need to pause the task, edit and add new tables, and then republish and restart it.

Q: Automatic Table Creation Failed?

A: Automatic table creation fails in the following cases:

- The current account does not have table creation permission on the target Catalog. Contact the administrator for authorization.
- If a source table field type has no corresponding type on the target side, click **Edit** in **Mapping Match Preview** to manually adjust the field type and retry.
- When Kafka is used as the source, automatic table creation is not supported. You need to create the table on the target side in advance and then select **Use Existing Table**.

Q: After System Fields Are Deselected, Do the Corresponding Columns Still Exist in the Destination Table?

A: Yes. After a system field is created in the target table, it will not be deleted if you deselect it. Instead, new data will no longer be written to it. To remove it completely, manually adjust the target table Schema outside the data ingestion task.

Q: The Task Reports an Error When a DDL Change Occurs After Field Mapping Is Edited?

A: In custom field mapping mode, DDL changes (adding columns, dropping columns, renaming columns, and modifying column types) are no longer automatically synchronized. Recommendations:

- Evaluate whether it can be restored to **Default Mapping**.
- If you must keep the custom mapping, manually adjust the target table Schema and task field mappings when changes occur on the source side.

References

- [Real-Time Database/Table Sharding Task Configuration](#)
- [Real-Time Synchronization Task Ops Monitoring](#)
- [Real-Time Synchronization Task Data Reconciliation](#)
- [Supported Access Solutions](#)
- [Supported Data Sources and Read/Write Capabilities](#)
- [Kafka Data Source](#)
- [MySQL Data Source](#)
- [Configure Notification Channels](#)
- [Create Compute Resource](#)

Real-Time Database and Table Sharding Task Configuration

Last updated: 2026-10-09 16:10:38

Overview

A real-time sharded database and table task aggregates real-time changes from **multiple data source connections, multiple databases, and multiple physical sharded tables** into the same target logical table in the DataBuddy data lakehouse. A common scenario is as follows: a business adopts horizontal database and table sharding (distributed across 8 databases and 32 tables by user_id hash) and wants to merge them into a single large table in the data lakehouse for analysis. This document describes the complete configuration process of the task.

Prerequisites

Before you start this operation, make sure the following prerequisites are met:

- You have activated the DataBuddy service and completed workspace initialization.
- The current account has data access permissions in the target Workspace. For details, see [Members and Permission Management](#).
- Data source connections have been created for **all** physical databases to be connected and have passed connectivity tests. For details, see [Add a Data Source Connection](#).
- A **real-time cluster** compute resource has been created. For details, see [Create a Compute Resource](#).
- CDC has been enabled in the source database.
- The schemas of all physical sharded tables involved in the merge must be **consistent**, with identical field names, types, and primary key constraints.

Use Limits

Limit	Description
Source Type	MySQL, Tencent Cloud MySQL, TDSQL-C MySQL, PostgreSQL, Oracle, SQL Server
Target Type	Iceberg native tables under the DataBuddy data lakehouse Catalog
Computing Resource	Only real-time cluster computing resources are supported.
Resource consumption	Single-task range: 1 to 256 CUs, with a step size of 1 CU

Physical table Schema	Schemas of all physical shards corresponding to the same logical table must be consistent.
Unsupported source type	Kafka, CKafka, MongoDB (to use them as real-time sources, use real-time whole-database multi-table tasks).

Operation Steps

A real-time sharded database and table task uses a wizard-based configuration, which includes five steps: **Basic Information** → **Source** → **Target** → **Data Verification** → **Runtime Settings**. The process is **consistent with the whole-database task**. This document focuses on the steps that differ, while the parts that are consistent with the whole-database task are reused through links to [Real-time Whole-Database Multi-Table Task Configuration](#).

Step 1: Entering the Task Creation Page

1. Log in to the [DataBuddy console](#).
2. Go to the workspace.
3. Go to **Data Integration** > Ingestion Tasks, and click **Create Task**.
4. After selecting the data source type, select **Real-time Sharded Database and Table Ingestion** as the ingestion method.

Step 2: Configuring Basic Information

After filling in the basic task information, click **Next**. For parameter details, see [Parameter Description](#) > [Basic Information Settings](#).

Step 3: Configuring the Source

1. **Data Source:** Select **multiple** data source connections of the same type from the drop-down list.
2. Configure the **logical table** mapping rule: Declare multiple physical databases and physical tables as the same logical table, which serves as the merge dimension on the target side. For specific rules, see the on-page prompts.
3. Select the **read mode:** **Full + Incremental** or **Incremental Only**.
4. Select the **subscription type:** Insert, Update, or Delete (at least one).
5. To configure advanced parameters, expand **Advanced Settings**.

Step 4: Configuring the Target

The target-side configuration items and runtime rules are exactly the same as those in [Target Settings for Real-time Whole-Database Multi-Table Tasks](#).

- **Basic Configuration:** Select the target Catalog, Schema matching policy, table matching policy, and system fields.

- **Table Creation Configuration:** Currently, only Iceberg native tables are supported. The platform generates table creation statements based on default field mapping rules, and users can modify the field mapping as needed.
- **Mapping Match Preview:** The source database and source table of each record are displayed as the logical database and logical table, while the remaining columns are consistent with those in whole–database tasks.

Step 5: Data Reconciliation

Configure data reconciliation rules as needed. For details, see [Real–Time Synchronization Task Data Reconciliation](#).

Tip

In a sharded database and table scenario, the reconciliation process **aggregates the row counts and primary keys of all source physical shards** and compares them with the target logical table. For example, if 8 databases and 32 tables on the source are merged into 1 table on the target, the total row counts and primary keys of the 8×32 physical tables are used to compare against the 1 target table.

Step 6: Configuring Runtime Settings

The DDL processing policy, task running policy, alarm notifications, and advanced settings are consistent with those of whole–database tasks. For details, see [Run Settings](#).

Step 7: Saving or Publishing

Operation	Action	Task Status
Create only	Save configuration without publishing.	Unpublished
Create and publish	Save and submit for publishing.	Pending start

For subsequent Ops operations such as start, restart, and stop, see [Real–Time Synchronization Task Ops Monitoring](#).

Parameter Description

Basic Information Settings

Parameter	Description	Required	Default Value
-----------	-------------	----------	---------------

Access Type	Select Source Type.	Yes	The destination is fixed as DataBuddy.
Access Mode	Real-time sharded database and table ingestion	Yes	Real-time whole-database multi-table ingestion
Task Name	Uppercase and lowercase letters, Chinese characters, digits, and underscores, with a length not exceeding 256	Yes	<code>\${Data Source Type}_YYYYMMDD_HHmms</code>
Computing Resource	Real-time cluster computing resources used for real-time synchronization	Yes	–
Resource consumption	Number of CUs, ranging from 1 to 256, with a step size of 1	Yes	–
Owner	Account used to run the task; must have access permissions on the target end.	Yes	Current Creating User
Task Description	No more than 200 characters	No	–

Source Settings

Parameter	Description	Required
Data source.	Data source connection of a matching type in the current Workspace, supporting multiple selection	Yes
Logical Tables	Merge multiple physical databases + physical tables into one logical table.	Yes
Read Mode	Full + incremental / Incremental only	Yes
Subscription Type	Insert, update, and delete, with at least one retained.	Yes
Advanced Settings	Pass-through parameters specific to the data source	No

 Tip

The **Run Identity** of a sharded database and table task must have access permissions to all selected data source connections. Failure to access any data source will cause the task to fail.

Target Settings

The parameters fully inherit the [Target Settings for Real-time Whole-Database Multi-Table Tasks](#).

Running Settings

The parameters are fully inherited from the [Run Settings for Real-time Whole-Database Multi-Table Tasks](#), including the DDL processing policy, Checkpoint interval, maximum number of restarts, alarm channels, and alarm types.

Verification Result

After the operation is completed, you can confirm in the following ways:

1. You can see the task in the **Data Integration > Access Tasks > Real-Time Synchronization** list, and the **Task Type** column displays **Database/Table Sharding**.
2. After the task is started, go to the Ops details page. In the link details list, each **physical database and physical table** corresponds to a mapping record, and the target columns all point to the same logical table.
3. In the data reconciliation overview, the **number of tables with consistent data** is counted by target logical table. In the details, you can view the differences in total row counts and primary keys between the source and target.

FAQs

Q: Can Multiple Target Tables Be Merged in a Sharding Scenario?

A: The current task uses **logical tables** as the merge unit. A single task can define multiple logical tables, and each logical table corresponds to a group of sharded databases and tables → one target table. If you need to split different business modules into multiple target tables, you can define multiple logical tables in the same task or split them into multiple tasks.

Q: What If the Schemas of Source Physical Sharded Tables Are Inconsistent? (Some Tables Have Additional Fields)

A: The current task recommends that the schemas of all physical sharded tables involved in the merge be **completely consistent**. If minor differences exist in individual tables, such as a table having extra fields, the target table will be created based on the union of fields from the source tables. If the schemas of the source tables differ significantly, the following is recommended:

- Fill in the differences on the source side to align all sharded tables.
- Alternatively, you can split out the differing tables and configure them as a separate task.

Q: What Happens If a Primary/Secondary Switchover Is Performed for the Instance of a Source Data Source?

A: Refer to the "Primary/Secondary Switchover" description in the corresponding data source section (such as [MySQL Data Source](#)). General rule: CDC links recover by GTID first. If GTID is not supported, you may need to restart the task and specify a recovery position.

Q: Can Tables Be Added to a Running Sharding Task?

A: The "logical table" configuration of a real-time database/table sharding task is fixed once saved. To change the scope, you need to pause the task, edit it, and then republish and restart it.

References

- [Real-Time Whole-Database Multi-Table Task Configuration](#)
- [Real-Time Synchronization Task Ops Monitoring](#)
- [Real-Time Synchronization Task Data Reconciliation](#)
- [Supported Access Solutions](#)
- [Supported Data Sources and Read/Write Capabilities](#)
- [MySQL Data Source](#)
- [Oracle Data Source](#)
- [Configure Notification Channels](#)

Ops Monitoring for Real-Time Sync Tasks

Last updated: 2026-10-09 16:10:43

Overview

Ops monitoring for real-time synchronization tasks covers the complete task lifecycle from publishing, starting, and running to stopping. This document describes how to view the task list, Ops details page, run logs, link details, and DDL change records, as well as common Ops operations such as starting, restarting, stopping, modifying the owner, and modifying alarms.

Prerequisites

- A real-time synchronization task has been created. For details, see [Real-time Whole-Database Multi-Table Task Configuration](#) or [Real-time Database/Table Sharding Task Configuration](#).
- The current account has the corresponding permissions on the task in the target Workspace: **View** / **Edit** / **Run** / **Manage** (for details, see [Members and Permission Management](#)).

Task Status

Status	Description	Available Operations (Primary Operation + More)
Unpublished	Task created but not published	Publish, Edit, Permission, Delete
Pending start	Task published but not started.	Start, Edit, Permission, Delete (After edits are saved, the primary operation changes to Publish)
In progress	Intermediate state of start, stop, or restart	–
Running	Stable running state after successful startup	Stop, Force Stop, Restart, Edit, Permission
Running (with updates)	A running task is edited and published again.	Restart (Primary Operation), Stop, Force Stop, Edit, Permission
Running (abnorm)	The number of restarts is greater than 1 within 10 minutes; the status is followed by the exception	Stop, Force Stop, Restart, Edit, Permission

al)	reason.	
Stopped	Stop operation completed.	Start, Edit, Permission, Delete
Failed	Startup failed (for example, startup timed out) or running failed (for example, the maximum number of restarts was reached); the status is followed by the exception reason.	Start, Edit, Permission, Delete

Operation Steps

Step 1: Opening the Task List

1. Log in to the [DataBuddy console](#).
2. Switch to the target region and Workspace at the top.
3. Go to **Data Integration > Task Management > Real-Time Synchronization**.

The task list displays the task name/ID, task type, status, data source, owner, creator, and update time. It supports fuzzy search by task name/ID and filtering by source type, data source connection, target Catalog, and compute resource.

Step 2: Publishing the Task

1. Select an **Unpublished** task and click **Publish**.
2. In the publish dialog, select:
 - **Publish only**: After publishing, the task enters the **Pending Start** state.
 - **Publish and start**: After publishing, the task directly enters the startup process (for details, see [Step 3](#)).

Step 3: Starting the Task

Starting involves three steps: detection → selecting a startup policy → viewing progress.

1. **Task Startup Detection**: The platform checks the completeness of the task configuration.
2. **Select a startup policy**: Different policies are available based on the current task state. For details, see [Startup and Restart Policies](#).
3. **View Progress and Results**: You can click **View Run Details** to go to the task Ops details page, or click **Close** to return to the list.

Start and Restart Policies

Current status	Optional startup policy	Remarks
Pending start	Start immediately and sync from the default position; start immediately and sync from a	Only MySQL / Tencent Cloud MySQL / TDSQL-C MySQL data sources support

	specified position; start immediately and sync from a specified point in time	specified positions and specified points in time.
Running / Running (with updates) / Running (abnormal)	Continue running and resume from the last position while retaining the job state; restart and resume from a specified point in time / a specified position; restart and start from the default position while discarding the job state.	"Continue running" applies to scenarios where the state does not need to be discarded after configuration changes.
Stopped	Same as the above four options	–
Failed	Resume from the Checkpoint where the last run failed; restart and start from the default position.	–

Step 4: Viewing Ops Details

Click the task name to go to the Ops details page:

- **Task Name Navigation Area:** Return to the list, copy the task name, and add to favorites.
- **Task Actions Area:** Start, stop, edit, delete, and permissions (Delete and Permissions are collapsed into "More").
- **Task Basic Information Area:** Task name, ID, type, data source, compute resource, run identity, start time, creator, and creation time. Data source and compute resource support click-through navigation.
- **Run Results:** See [Run Results](#).
- **Run Logs:** See [Run Logs](#).
- **Data Reconciliation:** See [Real-Time Synchronization Task Data Reconciliation](#).
- **Configuration Details:** Displays the task configuration of the latest published version.

Step 5: Common Ops Operations

Operation	Entries	Action
Starting	Start in the list / operation area on the details page	See Step 3
Stop.	Stop in the list / operation area on the details page	Graceful stop
Force stop	Force stop in the list / operation area on the details page	Skip graceful process and terminate immediately.
Restart	Restart in the list / operation area on the details page	Equivalent to stop + start

Editing an alarm policy	Edit in the list / operation area on the details page	Go to the task editing page.
Permission	Permissions in the list / More on the details page	See Permission Management
Modify owner.	More > Modify owner in the list	See Modify Owner
Alarm	More > Alarm in the list	Directly modify alarm channels / alarm types / thresholds without republishing.
Deleting an alarm policy	More > Delete in the list	Delete after secondary confirmation; running tasks must be stopped first.

Permission Management

Task permissions are controlled based on ACL entities. You can assign permissions such as **Manage** / **Run** / **Edit** / **View** to members. For details on permission semantics, see [Members and Permission Management](#).

Changing the Owner

The owner is the actual account that runs the task. For a running task, after the owner is changed, **the task must be restarted for the change to take effect**. Before the restart, an info icon appears after the Owner column in the list, indicating that the owner has been changed and the change will take effect only after a restart. When you change the owner from the Edit Task page, you must publish the task before starting it.

Execution result

Task Execution Status and Progress

Real-time tasks display progress in three stages: **Environment Preparation** → **Full Sync** → **Incremental Sync**.

Phase	Description	Progress display
Preparing the Environment	Task initialization (resource readiness, connection establishment, and more)	One-time phase
Full synchronization	Sync all existing data from the source table to the target table.	Progress percentage + completed / total tables
Incremental synchronization	Continuously sync real-time changes from the source.	Current latency (such as "2-second latency"), and "Stopped" when it is stopped

Trace Details

Display the sync status of each table in a list: source database, source Schema (if any) / source table, target Catalog / target Schema / target table, target connection method, table creation DDL / rows read, rows written, rows updated, rows deleted, and status.

Action	Description
Filter search	Source/target database, Schema, table, target connection method, and status can all be filtered.
Sort	Read quantity/write quantity support sorting.
Status	Normal / Abnormal (The specific reason is displayed when an exception occurs during running).
Jump	Click the target table to open the Catalog table details page in a new window.
Table creation DDL	When the target connection method is "Auto table creation", the complete table creation DDL is displayed; when it is "Use existing table", – is displayed.

DDL Change Records

This feature is supported only when the source type is **MySQL / Tencent Cloud MySQL / TDSQL-C MySQL**. For other source types, "DDL change records are not supported for the current link" is displayed.

Record fields: source database, source Schema (if any) / source table, target Catalog / target Schema / target table, change time, change policy, source table DDL statement, target table DDL statement, and change status.

Change Status	Conditions for Occurrence
Successful	The change policy is "automatic change" and the execution is successful.
Failed	The change policy is "automatic change" but the execution fails. You can click the info icon at the end of the row to view the failure reason, and the task ignores this change by default.
Ignored	The change policy is "ignore".

Run Logs

Capability	Description
Historical logs	Historical logs can be viewed in any state, including after the system is stopped.
Quick diagnosis	Text matching + keyword highlighting allows you to quickly locate matching content.
Automatic refresh	Disabled by default. To view the latest logs, click Refresh manually.
Allows you to query logs.	Supports keyword search and filtering by log level and time.

Data Reconciliation

Data reconciliation results are displayed in **Data Reconciliation**. For detailed configuration rules, alarms, and failure handling, see [Real-Time Synchronization Task Data Reconciliation](#).

Verification Result

After completing the Ops operations, you can confirm in the following ways:

1. The **task list status** has been changed to the target status, for example, to **Running** after the task is started, or to **Stopped** after the task is stopped.
2. The progress updates for the environment preparation, full sync, and incremental sync stages on the **Task Details** page are as expected.
3. The number of rows read and written for each table in the **Link Details** list continues to grow during the incremental sync stage.
4. There are no abnormal events in the **Run Logs**.

FAQs

Q: Why Does a Task Remain in the "In Progress" State?

A: "Operating" is an intermediate state of start, stop, and restart. It usually transitions to the next state automatically within minutes. If it remains in this state for a long time:

1. Check whether the run logs contain any error information.
2. Check whether the compute resource is in an abnormal state. For details, see [Compute Resource Usage Monitoring](#).
3. If you confirm that the system is stuck, contact technical support for assistance.

Q: How Do I Troubleshoot the "Running (Abnormal)" State?

A: Running (Abnormal) is usually triggered when the number of restarts within 10 minutes is > 1. Troubleshoot in the following order:

1. The abnormal reason prompt after the status.
2. The most recent error stack in the run logs.
3. Filter abnormal tables in Link Details and view the cause.
4. Check whether there are any failed changes in the DDL change records. When the change policy is "Automatic Change" and the execution fails, the task ignores the failure by default, but this may cause abnormal data writes later.
5. Check the alarm notifications to confirm whether the Failover restart quantity threshold has been exceeded.

Q: Why Does a Task Still Run with the Old Account After the Owner Is Changed?

A: After the owner is changed, **the task must be restarted for the change to take effect**. Click **Restart** at an appropriate time. Otherwise, the task continues to run as the original owner. If the change was made from the Edit Task page, you must publish the task before starting it.

Q: Can a Task Be Recovered After Deletion?

A: Deletion is a terminal operation. Perform the steps in the following order with caution: Stop → Confirm no dependencies → Delete. A running task cannot be deleted directly. You must stop it before deleting it.

Q: Can I Start Multiple Tasks at Once?

A: The current task list does not provide a batch start entry. Each task must be started individually.

References

- [Real-Time Whole-Database Multi-Table Task Configuration](#)
- [Real-Time Database/Table Sharding Task Configuration](#)
- [Real-Time Synchronization Task Data Reconciliation](#)
- [Offline Synchronization Task Ops Monitoring](#)
- [Configure Notification Channels](#)
- [Member and Permission Management](#)
- [Compute Resource Usage Monitoring](#)

Real-Time Sync Task Data Reconciliation

Last updated: 2026-10-09 16:10:49

Overview

Data reconciliation periodically compares data between the source and target to promptly detect data inconsistencies. DataBuddy real-time synchronization tasks support two reconciliation rules based on **the number of data records** and **data content (primary key)**, covering both full and incremental scopes, and provide alarm notifications and discrepancy detail viewing. This document describes the supported scope, configuration methods, and result viewing for data reconciliation.

Prerequisites

- A real-time whole-database or database/table sharding synchronization task has been created.
- The source and target must be within the [supported link](#) range.
- If the "Verify data content" rule is used, the source table must contain a primary key or a unique index.

Use Limits

Limit	Description
Task type	Real-time whole-database migration tasks / real-time sharded database and table migration tasks are supported
Primary key	Data content validation supports only primary key-based reconciliation. Tables without a primary key or unique index are not supported.
MongoDB	MongoDB sources do not support incremental validation (only full validation is supported).
Number of tables in a single task	When the number of databases and tables involved in a task exceeds 5000 , data validation cannot be configured per table (this does not affect the running of the overall synchronization task).
Reconciliation period	The minimum reconciliation interval is 60 minutes .
Difference details preview	A maximum of 10000 difference records can be previewed at a time, and only primary key fields are displayed.
Resource consumption	Validation tasks and synchronization tasks share the same compute resources , consuming resource group resources.

Target-side filter condition	The filter SQL for incremental validation is executed in Flink SQL and supports only Flink SQL syntax (for example, write the current date as <code>\${YYYY-MM-dd}</code> instead of <code>CURRENT_DATE()</code> , and use <code><></code> instead of <code>!=</code> for inequality).
------------------------------	--

Supported Links

Task type	Source End
Whole-database migration task	MySQL,TDSQL-C MySQL,Oracle,MongoDB,Kafka
Sharded database and table task	MySQL,TDSQL-C MySQL,Oracle

Operation Steps

Step 1: Enabling Data Reconciliation

Data reconciliation is configured in the **Data Reconciliation** step of the task configuration process:

1. Go to the **Data Reconciliation** step in the [Real-time Whole-Database Multi-Table Task Configuration](#) or [Real-time Database/Table Sharding Task Configuration](#) process.
2. Turn on the **Data Reconciliation** switch. When it is off, no tables are reconciled. When it is on, all tables are reconciled uniformly according to the global reconciliation policy.

Step 2: Configuring a Global Reconciliation Policy

Parameter	Description	Required
Reconciliation rule	Verify data record count / Verify data content. See the description below for details.	Yes
Reconciliation field	Displayed only when Data Content Validation is selected. Currently, only primary key-based reconciliation is supported.	Depends on the rule
Reconciliation scope	Full reconciliation / Incremental reconciliation. See Reconciliation scope for details.	Yes
Reconciliation time	Reconciliation interval, minimum 60 minutes	Yes

Reconciliation Rules

Rule	Action	Scenario
Validate the number	Directly compare the row counts of the source and target tables through the data	Small data volume, focusing on total count consistency.

of data records	source connection by using <code>count (*)</code> .	
Validate data content	Compare the primary key sets of the source and target ends. If the primary keys are inconsistent, the data is considered inconsistent.	Stable primary key and focus on whether each record is written to the database; source/target must have a primary key or unique index.

Reconciliation Scope

Scope	Action	User input
Full reconciliation	Reconcile full data between source / target tables	–
Incremental reconciliation	Reconcile data filtered by specified conditions, suitable for time-based partitioning.	Source-side filter condition + target-side filter condition

Incremental Reconciliation Note:

- Time field filtering supports [system built-in parameters](#).
- Fine-grained reconciliation is supported (for example, by hourly partitions), and grouping syntax can be used.
- The source and target filter conditions must each comply with the SQL syntax of the corresponding data source.
- If a field in the filter condition does not exist in the source or target, you need to modify it by table in "Single-Table Reconciliation". Otherwise, reconciliation for that table will fail.

Example (reconciliation by hourly partitions):

```
select count(1), partition from table where partition >
${yyyyMMddHHmmss-8H} group by partition
```

Note:

The filter SQL on the target side is executed in Flink SQL, and only Flink SQL syntax is supported:

- Use `${YYYY-MM-dd}` to express the current date. Do not write `CURRENT_DATE()` .
- Use `<>` for "not equal to". Do not write `!=` .

Reconciliation fails if the SQL does not comply with Flink SQL syntax.

Step 3: (Optional) Customizing Reconciliation by Table

A global policy applies to all tables in the task. If differentiated processing is required:

1. In the reconciliation step of the configuration process, click the single table that requires differentiated processing.
2. You can enable or disable data reconciliation for a single table, or modify the incremental reconciliation conditions for a single table.

Note:

Changes to single-table reconciliation take effect after the task is published and running. If a table requires different filter conditions or does not participate in reconciliation, adjust it completely before saving.

Step 4: (Optional) Configuring Reconciliation Alarms

The alarm rule page for real-time synchronization tasks supports configuring data reconciliation alarms. **This takes effect only for tasks with data reconciliation enabled.**

Alarm Metric	Description	Threshold
Data reconciliation / Task failed.	The data reconciliation resident task failed to run.	Send N alarms every X minutes/hours/days upon failure.
Data reconciliation / Reconciliation failed	Some tables in the reconciliation details are in the reconciliation failed state.	Send N alarms every X minutes/hours/days upon failure.
Data reconciliation / Reconciliation time	The duration of a single data reconciliation exceeds the threshold.	Reconciliation time > X minutes/hours/days
Data reconciliation / Number of inconsistent data records	The number of data discrepancy records in some tables in the reconciliation details is not 0.	Number of inconsistent data records > N

- Alarms are triggered for **the number of inconsistent data records at table granularity**. An alarm is triggered when the number of inconsistent records in any table exceeds the threshold.
- For alarm channels and basic rules, see [Set Notification Channel](#).

Viewing Reconciliation Results

Real-time synchronization tasks perform continuous and periodic reconciliation based on the configured first reconciliation time and reconciliation interval.

Entries

Go to the task Ops details page (see [Real-Time Synchronization Task Ops Monitoring](#)), and switch to the **Data Reconciliation** page.

Reconciliation Overview Metrics

Metric	Description
Total Number of Tables	Number of tables involved in the synchronization task; in sharded database and table scenarios, counted by the number of tables on the destination side
Number of Consistent Tables	Number of tables with identical row counts or primary keys between source and destination
Number of Inconsistent Tables	Number of tables with differences between source and destination (number of differences $\neq$ 0)
Number of Failed Reconciliations	Number of Tables That Failed During Reconciliation
Number of Unreconciled Tables	Number of tables for which data reconciliation is not enabled

Reconciliation Details

The reconciliation results are displayed row by row for each table in the details. For records that are reconciled based on primary keys and have the number of differences not equal to 0, **view difference details** is supported:

Item	Description
Difference details preview limit	10000 entries
Display Fields	Display primary key fields only

Reconciliation Task Statuses and Supported Operations

The reconciliation task is a resident task that runs automatically at the configured time interval:

Status	Description	Available Operations
Reconciliation task not started	Not started	Running
Reconciliation task running	Running	Stop.

Reconciliation task stopped	Manually stopped	Running
Reconciliation task failed	Startup failed or running failed	Running
Reconciliation task in progress	Intermediate state	–

Verification Result

After the operation is completed, you can confirm in the following ways:

1. In the **Data Reconciliation** overview, verify that the **total number of tables** and the counts of each category meet expectations, and that the **number of unreconciled tables** equals the number of tables without reconciliation configured.
2. New reconciliation records are generated periodically in Reconciliation Details.
3. When reconciliation alarms are configured, notifications can be received through the alarm channel when trigger conditions are met.
4. When a table-specific custom reconciliation configuration takes effect, the reconciliation rules for that table differ from the global rules when details are viewed.

FAQs

Q: Does Enabling Data Reconciliation Slow Down Task Performance?

A: There will be some impact. Data reconciliation needs to pull data from the source and target, which **places a certain load on the data source**. In addition, the reconciliation task and the synchronization task **share the same computing resources**, consuming resource group resources. Recommendations:

- Do not set the reconciliation interval too short (minimum 60 minutes).
- For large tables, use **Verify Data Row Count**. For small and medium tables, use **Verify Data Content**.
- Resource group monitoring metrics include the resource usage of data reconciliation instances. Evaluate the resource group specification based on the actual load.

Q: How Do I Troubleshoot a Single-Table Reconciliation Failure?

A: Common causes:

1. If a field in the **filter condition does not exist in the source or target**, adjust the incremental reconciliation condition for that table in the single-table reconciliation configuration to make the field name match.
2. **The filter SQL on the target side uses non-Flink SQL syntax**, such as `CURRENT_DATE()` and `!=`. Replace them with `${YYYY-MM-dd}` and `<>`.
3. **Incremental reconciliation is configured for MongoDB**. MongoDB does not support incremental reconciliation. Change it to full reconciliation or disable reconciliation for this table.

4. The task involves more than 5000 tables and cannot be configured on a per-table basis. The current scale cannot support single-table reconciliation. Split the task into multiple tasks.

Q: Why Is the Number of Differences 0 but the Content Still Inconsistent for Primary Key-Based Reconciliation?

A: Yes. Currently, **Verify Data Content** only compares the primary key set, not other field values. If your business requires full-field consistency comparison, consider the following recommendations:

- Add field value verification through a data quality task. For details, see [Create a Quality Task](#).
- Or customize a comparison script through an SQL task. For details, see [SQL Task](#).

Q: What Should I Do If Only 10,000 Difference Details Are Displayed?

A: Difference details are only used to quickly locate difference samples. When the number exceeds 10000, consider the following recommendations:

1. Determine the cause of the inconsistency by using difference samples, such as missed DDL synchronization, omitted filter conditions, or write failures.
2. After fixing the synchronization task, restart the reconciliation once.
3. If full comparison is required, we recommend that you create a separate SQL task to write the source and target primary key sets to intermediate tables and then compute the difference.

Q: Can a Reconciliation Task Be Stopped Separately Without Affecting Data Synchronization?

A: Yes. The reconciliation task is a resident task. You can click **Stop** individually in **Data Reconciliation**, and this **does not affect the main data synchronization task**. After stopping, click **Run** to resume.

References

- [Real-Time Whole-Database Multi-Table Task Configuration](#)
- [Real-Time Database/Table Sharding Task Configuration](#)
- [Real-Time Synchronization Task Ops Monitoring](#)
- [Create Compute Resource](#)
- [Configure Notification Channels](#)
- [System Built-in Parameters](#)

File Upload

Uploading Local Data to a Table

Uploading Local Files to a Table

Last updated: 2026-10-09 16:10:57

Overview

Local files are uploaded to tables to parse local structured and semi-structured files into structured data tables, which are then stored in a specified Schema under the Catalog (data catalog). After the upload is complete, the table can be searched in the [data catalog](#) and can be directly used for SQL queries, dashboards, and downstream task processing.

The entry card in the console is named **Upload File to Table**. It opens as a drawer on the right, and the overall workflow is "Select File → Preview Data and Confirm Schema → Storage Settings → Create Table".

Applicable Scenarios

- **One-time data analysis:** Directly create tables from manually organized sales tables, inventory tables, reconciliation files, and other data for temporary analysis.
- **Quickly load sample data into the lake:** Load small-scale sample data into tables to verify SQL logic or build dashboard prototypes.
- **Scattered files without modeling:** When source data consists only of files and no connectable database is available, tables are generated directly from the files.

To **periodically** and continuously pull new files into tables, use [File to Table Sync](#) instead. To retain the original file format without parsing, use [Local File Upload to Volume](#).

Prerequisites

Condition	Description
Data Integration Permission	The current account has the operation permission for the Data Integration module in the associated Workspace. For details, see Members and Permission Management .
Computing Resource	Available interactive cluster computing resources exist in the current Workspace for parsing files and creating tables. For details, see Create Computing Resources .
Target Catalog and Schema	Have write permission on the target Catalog and Schema. If no available Schema exists, create one immediately in the storage settings step.

File to Upload	File types and sizes comply with supported file types and constraints .
----------------	---

Use Limits

Limit	Description
File Type	CSV, JSON, Avro, Parquet, Text, and XML, with corresponding extensions <code>.csv</code> , <code>.json</code> , <code>.avro</code> , <code>.parquet</code> , <code>.txt</code> , and <code>.xml</code>
Number of files	At least 1 and up to 10 files per operation, with mixed types allowed.
File Size	Total size of all files does not exceed 2 GB.
Preview Rows	Data preview displays up to 50 rows.

For details about the parsing behavior, field naming rules, and complete error messages of each type, see [Supported File Types and Constraints](#).

Operation Steps

Step 1: Entering File Upload to Table

1. Log in to the DataBuddy console and switch to the target Workspace.
2. Select **Data Integration** in the left sidebar and confirm that you are on the Access Method Tab.
3. In the File area, click the **Upload File to Table** card. The system opens the Upload File to Table drawer on the right.

Step 2: Selecting a Local File

Two methods are supported for adding files:

- **Drag and drop:** Drag files directly to the file upload area in the drawer.
- **Click to select:** Click **Select File** to select one or more files in the system file browser.

Note:

When the number of files or the total size exceeds the limit, the page displays a non-blocking message that disappears automatically after a few seconds. Files exceeding the limit are not added to the list. Split or reduce the files and try again.

Step 3: Selecting a Service Resource

The service resource is used to parse uploaded files and create tables, and it is a required option.

Parameter	Description	Required	Default Value
Service resources	Single choice. The drop-down list displays only interactive cluster compute resources that the current account has permission to use, and shows the current status of the resources.	Yes	–

- After you select a resource, the page displays the creator and resource type. Click **Details** to open the compute resource details page in a new tab.
- If the selected resource is not started, the system automatically starts it before parsing the file.

Tip

Compute resources are billed based on actual usage. For resource specifications and the automatic start/stop policy, see [Compute Resource Overview](#).

Step 4: Previewing Data and Confirming the Schema

After the file is uploaded, you automatically go to the Data Preview step. The system first writes the file to the lakehouse and then parses it. After parsing is complete, the preview table displays data samples, and all operations become available again.

In this step, you need to confirm the following:

1. Confirm the table name: The table name cannot be the same as an existing table.
2. Confirm fields: Select the fields to be stored, and modify the field names, field types, and DECIMAL precision as needed. Keep at least one field.
3. To correct parsing rules: Click **Advanced Settings** and adjust the delimiter and row tag parsing parameters based on the file type.

For complete instructions on field editing, see [Preview and Schema Editing](#). For parsing parameters that can be adjusted for each file type, see [File Parsing and Inference](#).

Note:

Modifying Advanced Settings triggers reparsing, and manually adjusted field names and field types may be reverted to the system-inferred results. It is recommended to stabilize the parsing rules before editing fields.

Step 5: Storage Settings

After confirming the preview results, go to the Storage Settings step to specify the storage location for the table and the handling method for tables with the same name.

Parameter	Description	Required	Default Value
Target Location	Select Catalog and Schema in the directory tree. The table will be created under the selected Schema. Only directories that you have permission to access are displayed.	Yes	–
Solution	Overwrite with same name: overwrite the existing table with the same name using the current data; Create a copy: create a new table named "TableName_1".	Yes	–

- The directory tree supports searching for catalogs and schemas by keyword.
- When you hover over a catalog, **+ Schema** appears on the right. Click it to create a Schema directly without exiting the current process.
- Click **Data Catalog** to browse the complete data catalog in a new tab.

Warning

Selecting "Overwrite with the same name" replaces the original data in the target table with the data uploaded this time, and the overwritten data cannot be recovered. To retain the original table data, select "Create a new copy".

Step 6: Creating a Table and Verifying the Result

1. Click **Create Table**. During table creation, all options in the current step are unavailable.
2. After the table is created successfully, the page automatically redirects to the location of the table in [Data Catalog](#).
3. In the data catalog, confirm that the table has been generated and that the fields are as expected. You can then query the table using SQL. For metadata management of the table, see [Data Catalog Overview](#).

Reusing the Most Recent Upload

When you open the "Upload File to Table" drawer, if you have a successfully submitted upload record within the last 24 hours, the page prompts you with the most recently uploaded file. Click **Preview File** to reuse that batch of files and continue with the subsequent steps without reselecting local files.

Feature	Description
Validity period	24 hours; records are no longer displayed after expiration.
Visibility scope	Only the current account's own upload records are displayed; upload records of other members are not displayed.

FAQs

Q: What should I do if a single upload exceeds 10 files or the total size exceeds 2 GB?

A: Upload data in multiple batches. If the data volume is large or periodic ingestion is required, use [File to Table Sync](#) to process data through offline sync tasks.

Q: Can multiple files with inconsistent fields be merged into a single table?

A: Yes. In table merge mode, the system takes the union of fields from all files, and data rows missing a field are filled with null. If the file structures differ significantly, it is recommended to use separate table upload to avoid generating a large number of empty fields.

Q: Will nested structures in JSON and XML be expanded into multiple columns?

A: No. Nested content is parsed as a single text field. To break it down, you can process it with SQL functions after loading it into the table. For details, see [File Parsing and Inference](#).

Q: Will field names change after table creation if the source file field names are in Chinese?

A: Yes. The system automatically normalizes field names by converting Chinese characters to pinyin and replacing illegal characters with underscores. You can manually change them to the desired names in the Data Preview step. For details about the rules, see [Supported File Types and Constraints](#).

Q: Can the table structure be adjusted after upload?

A: Field adjustments after table creation must be completed in the data catalog or through SQL tasks. If only the parsing of this upload is incorrect, it is recommended to correct it in the Data Preview step and then recreate the table.

Q: What should I do if "No permission to perform the upload service" is displayed?

A: This usually occurs when permissions for the Data Integration module or compute resource usage have changed. Contact your administrator to verify the permission configuration. For details, see [Member and Permission Management](#).

References

- [Supported File Types and Constraints](#)
- [File Parsing and Inference](#)
- [Preview and Schema Editing](#)
- [Upload COS Files to Table](#)

- [Upload Local Files to a Volume](#)
- [File-to-table synchronization](#)
- [Compute Resource Overview](#)
- [Data Catalog Overview](#)

Supported File Types and Constraints

Last updated: 2026-10-09 16:11:00

Overview

This document lists the supported file types, single upload constraints, and common error messages for [local upload to table](#), helping you perform necessary preprocessing and splitting of files before uploading.

Supported File Types

File Type	Extension name.	Description
CSV	<code>.csv</code>	Comma-separated structured text, a common export format
JSON	<code>.json</code>	JSON document; supports three layout modes: a single object, an array, and multiple objects (one per line).
Avro	<code>.avro</code>	Self-describing binary row-based format that requires no additional field specification
Parquet	<code>.parquet</code>	Self-describing binary columnar format that requires no additional field specification.
Text	<code>.txt</code>	Plain text file with custom delimiters
XML	<code>.xml</code>	XML document. A row tag is required (the system will infer it automatically).

For details about the parsing behavior and adjustable parameters of each type, see [File Parsing and Inference](#).

Note:

A single upload supports **mixing multiple types**. For example, you can upload 1 CSV + 2 JSON files at a time. However, it is recommended that you upload files with **similar structures** in the same batch. Otherwise, a large number of empty fields will be generated in table merge mode.

Constraints for a Single Upload

Constraint item	Limit	Validation message
Number of files	At least 1, up to 10	You can upload up to 10 files.

Total file size	Total size of all files ≤ 2 GB.	You can upload files up to 2 GB in total.
-----------------	---------------------------------	---

Naming Rules of Fields

Source file fields may contain Chinese characters, illegal characters, or other content that is not suitable for table creation. Therefore, DataBuddy **automatically normalizes field names** during the file upload to table process. You can also manually adjust them in the Data Preview stage:

Original field	Processed Field	Rule
Space	Clear	Clear a field name if it contains only spaces; if all field names are empty, assign <code>c_1</code> and <code>c_2</code> in sequence.
Chinese	Pinyin	For example, "Name" → <code>mingcheng</code>
Invalid characters	<code>_</code>	For example, <code>a&b</code> → <code>a_b</code>

Definition of valid characters: letters, digits, underscores, `/`, `=`, and `-`.

Validation rules for the field name input box:

Rule	Description
First character	Must be a letter, digit, or underscore.
Character range	Only letters, digits, underscores, <code>/</code> , <code>=</code> , and <code>-</code> are allowed.
Length	No more than 64 characters.

Table names are processed according to the same rules: the table name is automatically populated after the original file name is parsed.

Common Error Messages

Scenario	Note.	Handling Recommendation
File quantity exceeded	You can upload up to 10 files.	Delete some files and retry.
Total capacity exceeded	You can upload files up to 2 GB in total.	Split files or keep only the necessary subset.

The file type is not supported.	Unable to select files.	Check whether the file type is in the supported list or convert the file type first.
Insufficient permissions	No permission to perform the upload service.	Check whether module permissions or service resource permissions have changed. For details, see Member and Permission Management .
Fallback error	Upload failed. {Error details}.	Troubleshoot based on the specific error description.

References

- [File Parsing and Inference](#)
- [Preview and Schema Editing](#)
- [Upload COS Files to Table](#)
- [File-to-table synchronization](#)

File Parsing and Inference

Last updated: 2026-10-09 16:11:03

Overview

During the data preview phase, DataBuddy **automatically infers** the file structure and field types. You can correct the inference results through the **Conversion Settings** entry. This document describes the adjustable parameters by file type.

The entry to conversion settings is located above the field table on the Data Preview page. Click **Advanced Settings** to open the parameter panel.

CSV

Configuration	Default Value	Description
Column delimiter	Automatically inferred; falls back to <code>,</code> when inference fails.	Options: custom, <code>,</code> , <code>;</code> , Tab
Escape character	<code>\</code>	Options: <code>"</code> or <code>\</code> . Used to escape and store illegal characters that appear in values.
Use first row as header	Selected	When this option is cleared, field names are automatically generated as <code>t__c0</code> / <code>t__c1</code> , and the original first row is written as a data row.
Span rows	Unselected	Whether a single record is allowed to span multiple lines (for cases where values contain line breaks).

Note:

Example: The source file department field value is `<script://alert('abc')>`. When the escape character is `"`, it is stored as `<script:":"/alert("'abc'")>`. When the escape character is `\`, it is stored as `<script:\\/alert(\'abc\')>`.

Text

Configuration	Default Value	Description
Column delimiter	Automatically inferred; falls back to <code> </code> when inference	Options: custom, <code>;</code> , <code> </code> , Tab

	fails.	
Row delimiter	Automatically inferred; falls back to line break <code>\n</code> when inference fails.	Options: custom, carriage return <code>\r</code> , line break <code>\n</code>
Use first row as header	Selected	When this option is cleared, field names are automatically generated as <code>t__c0</code> / <code>t__c1</code> , and the original first row is written as a data row.
Escape character	<code>\</code>	Used to escape illegal characters

JSON

Configuration	Default Value	Description
Convert time text to time type.	Selected	When a field value matches time characteristics, it is automatically recognized as a time type; otherwise, it is parsed as text.
Allow single quotes	Selected	When this option is selected, single quotes are parsed as double quotes; when this option is deselected and the text contains single quotes, a message is displayed indicating that the json format cannot be inferred.
Filter comments	Selected	Only single-line <code>//</code> and multi-line <code>/* */</code> standard comments are supported; non-standard comments such as <code>?</code> comments and agreed keys are not supported.
Span rows	Selected	When this option is selected, only complete JSON bodies are processed and incomplete rows are ignored; when this option is deselected, incomplete rows are parsed as empty rows (all fields are null).

Recommended JSON format:

- Multiple JSON body format: Each line is an independent JSON object.
- Array format: The root element is a JSON array, and each array element is a record.

Nested processing: When JSON nesting exists, the nested field is parsed as a single text object and **is not automatically expanded into multiple columns**. To break it down, post-process it after loading into the table by using functions such as `get_json_object` / `from_json` in an SQL task.

XML

Configuration	Default Value	Description
Row Tag	Automatically inferred (using the outermost first-level tag)	Determines which tag serves as the delimiter for a row of data. If automatic inference is incorrect, you can specify it manually.
Use Attributes as Fields	Selected	When this option is selected, XML element attributes are treated as separate fields; when this option is deselected, attributes are ignored.
Attribute Prefix	-	When "Use Attributes as Fields" is selected, the field name format is <code>attribute prefix + attribute name</code> .

Note:

Row tag nesting behavior: If a row tag appears in a nested structure in the XML, only the level defined by the row tag is treated as a data row. Data at the parent level of that row tag is considered meaningless and is not written to any field.

Attribute prefix conflict handling: If the attribute prefix or attribute name conflicts with an existing field name, `_sequence number` is automatically appended to the end (for example, `a_name_1`) to avoid overwriting.

Avro / Parquet

Avro and Parquet are **self-describing formats**, and the files already contain Schema information. The system directly reads and displays the inference results without requiring additional conversion settings.

To adjust field types or field names, you can modify them directly in the field table on the Data Preview page. For details, see [Preview and Schema Editing](#).

Automatic Inference of Field Types

During the preview phase, the system automatically infers the type for each field based on the first few rows of data. The currently supported field types include (available in the field type drop-down menu):

Type family	Description
String	STRING / VARCHAR
Integer	INT / BIGINT
Floating point	FLOAT / DOUBLE

High-precision number	DECIMAL (precision and scale must be specified)
Boolean	BOOLEAN
Time	DATE / TIMESTAMP

DECIMAL precision settings: When the field type is set to DECIMAL, hovering over the type displays an edit icon. Click it to set the precision and scale.

Parameter	Value Range	Description
Precision (precision)	1 ~ 38	Maximum number of digits to the left of the decimal point
Scale (scale)	0 to Precision	Maximum number of digits to the right of the decimal point. A red error message is displayed when the precision limit is exceeded.

Relationship Between Advanced Settings and Field Editing

- Advanced Settings modifies the **parsing rules** (such as delimiters and row tags). After modification, the system reparses the file, and the preview data and field list are refreshed.
- The field name, field type, and table creation field modify the **Schema of the target table**. These modifications do not trigger parsing again and only affect the field definitions used for the final table creation.

If you have manually modified field names or types before adjusting Advanced Settings, the field list may revert to the system's default inference results after a reparse is performed. It is recommended to **stabilize the conversion rules before editing fields**.

References

- [Supported File Types and Constraints](#)
- [Preview and Schema Editing](#)
- [File-to-Table Synchronization – Parsing Configuration](#)

Preview and Schema Editing

Last updated: 2026-10-09 16:11:06

Overview

Data Preview is a key step in the [local upload to table](#) process, between Upload File and Storage Settings.

On this page, you can:

- Adjust the **field name**, **field type**, and **sharding field** to finalize the Schema for data storage.
- View the first 50 rows of data samples to verify the parsing results.

Table Creation Methods

After parsing the file, you can choose how to create the table:

Method	Action	Scenario
Create a new table.	Create a new table in the target catalog.	Initial Upload of Scattered Files
Overwrite an existing table.	Select an existing table and overwrite its data.	Update data in an existing table.

Storage Location

The target location settings for table creation include two configuration items:

- For catalog and schema, you can select a table-type catalog. If no suitable catalog or schema is available, click **Create Data Catalog** or the plus icon (+) to create a schema.
- Table name: If you choose to overwrite an existing table, select an existing table. If you choose to create a new table, enter a table name.

Field Operations

The field headers in the table preview support the following operations:

Operation	Entries	Description
Alter Field Name.	Click the field name in the table header.	Field names cannot be empty or duplicate; the default value is the name inferred from parsing.
Modify the field type.	Click the type icon next to the field name.	A type drop-down menu appears, from which you can select STRING / INT / BIGINT / FLOAT / DOUBLE / DECIMAL / BOOLEAN / DATE / TIMESTAMP.

Set DECIMAL precision.	Select DECIMAL as the field type, hover over it to display the edit icon, and then click it.	Set precision (1—38) and scale (0—precision). For details, see File Parsing and Inference – DECIMAL Precision Settings .
Select table creation fields.	Select field checkboxes.	Only selected fields are written to the target table, and other fields are not stored. Select at least one field (the last remaining field cannot be cleared).

Data Preview Table

Feature	Description
Preview Rows	Display up to 50 rows.
Number of Data Rows/Columns	Number of rows and columns actually read from the file
Data Preview Table	Display preview data based on the selected fields.

Advanced Settings Entry

An **Advanced Settings** entry is provided above the preview table, allowing you to adjust parsing rules for the current file type. For details, see [File Parsing and Inference](#).

Note:

After you modify Advanced Settings, the system reparses the file and refreshes the preview data. If you have previously changed field names or field types manually, edit the fields only after the conversion rules become stable to avoid repeated modifications.

Go to the Next Step

After confirming that the preview data, field names, field types, and archived fields are correct, click **Create Table** to go to Storage Settings, where you can select the target Catalog/Schema and processing method. For details, see [Local Upload to Table – Step 5: Storage Settings](#).

References

- [Upload Local Data to Table](#)
- [Supported File Types and Constraints](#)
- [File Parsing and Inference](#)

Uploading COS Files to a Table

Last updated: 2026-10-09 16:11:10

Overview

The Tencent Cloud COS file upload to table feature is used to parse structured or semi-structured files already stored in COS buckets and **directly** load them into target tables under Catalog, eliminating the need to download files locally before they are uploaded.

The overall interaction is consistent with [local upload to table](#). The main difference lies in the Source step: browse and select files from a COS bucket instead of local browser files.

Applicable Scenarios

- **Data has been accumulated in COS:** For example, tracking logs, reconciliation files, and third-party SaaS export files are regularly delivered to COS, and you may want to quickly create tables for one-time analysis.
- **Cross-account and cross-region data ingestion:** Ingest files from other COS buckets (with access permissions) into the lakehouse of the current Workspace.
- **Avoid local transfer of large files:** Avoid downloading multi-GB files before uploading them. Instead, complete parsing and table creation directly in the cloud.

To **periodically** and continuously pull new date-partitioned files from COS into tables, use [File to Table Sync](#) instead.

Prerequisites

Condition	Description
Data Integration Permission	The current account has the operation permission for the Data Integration module in the associated Workspace.
COS Access Authorization	The current Tencent Cloud account has been granted the COS access permission required for this module (CAM authorization).
Computing Resource	Created data interactive cluster computing resources for parsing and table creation.

Operation Steps

Step 1: Entering COS File Upload to Table

In the console, go to the **Data Integration** module. On the Access Method Tab, click the **Tencent Cloud COS File Upload to Table** card. The system opens the "Tencent Cloud COS File Upload to Table" drawer on the

right.

Step 2: Selecting a COS Bucket

Parameter	Description
Select COS Bucket.	The drop-down list displays the COS buckets that the current account has permission to access.

The buckets in the dropdown list come from the COS resources of the current Tencent Cloud account. If the list is empty, see [COS Authorization Process](#) to complete authorization or create a bucket.

Step 3: Selecting a File

After a bucket is selected, the list displays the files and folders in the bucket root directory.

Ac tio n	Description
Br o w se	Click a folder name to go to the next level. Empty folders do not display subitems.
R eq uir ed or N ot	Select the checkboxes before file names. Multiple selection is supported. Select at least one file.
Li mi t	Only files can be selected, and folders cannot be selected . File type and quantity constraints are the same as those for uploading local files to a table (up to 10 files, with a total size no greater than 2 GB).

If the list displays "No bucket address retrieved yet" or a bucket displays "No data available", go to the COS console first to confirm that the bucket and files actually exist.

Step 4: Selecting Compute Resources

Compute resources are required and support single selection only. Only interactive cluster compute resources that the current account has permission to use are displayed. If the selected resource is not started, the system starts it automatically before parsing. For details, see [Local File Upload to Table – Step 3: Select Service Resources](#).

Step 5: Data Preview

Click **Preview Table** to enter data preview. The subsequent behaviors (parsing inference, table creation method, field editing, and advanced settings) are exactly the same as those for local upload to table:

- For details about the parsing and inference logic, see [File Parsing and Inference](#).
- For details about field editing and table name modification, see [Preview and Schema Editing](#).

Step 6: Storage Settings and Table Creation

Go to the Storage Settings step, select the target Catalog/Schema and processing method (overwrite with the same name or create a new table), and click **Create Table Now** to complete table creation. For details, see [Local File Upload to Table – Step 5: Storage Settings](#).

COS Authorization Process

DataBuddy accesses COS through the Tencent Cloud CAM system. If the current account has not granted DataBuddy access to COS, an authorization guide is displayed when you go to "Select COS Bucket":

1. The page displays the current Tencent Cloud account information:
 - Account name: the username of the current Tencent Cloud account.
 - Account ID: the UIN of the current account.
 - Required permissions: the COS permission items required by this module.
1. Click the **Copy** icon to copy the above information.
2. Click **Tencent Cloud CAM** to go to the [Tencent Cloud CAM console](#) in a new tab.
3. After adding COS-related permission policies to accounts or roles in CAM as needed, return to the DataBuddy page and click **Refresh**:
 - After authorization is successful, the "Select COS Bucket" section displays a list of accessible buckets.
 - If you still have no permissions and the guide page is still displayed, check whether the CAM policy has taken effect.

FAQs

Q: Why does the file list show "No Data" after a COS bucket is selected?

A: The root directory of the bucket may indeed be empty, or the selected bucket may be located in a different region or under a different sub-account. Go to the COS console to check the actual status of the bucket.

Q: Can I select a COS folder to import data into a table in batches?

A: Currently, selecting folders is not supported. Only file selection is supported. To batch import files from an entire directory into a table, use [File-to-Table Sync](#).

Q: What should I do if the total size of COS files exceeds 2 GB?

A: As with local uploads, a single upload is still subject to the 2 GB limit. We recommend splitting the upload into multiple parts or using [File-to-Table Sync](#) to ingest data through offline sync tasks.

Q: Is cross-region COS supported?

A: It depends on the network reachability between the current Workspace and the COS bucket. For specific reachable regions, see the list of supported regions displayed in the DataBuddy console.

References

- [Upload Local Files to Table](#)
- [Upload Local Files to a Volume](#)
- [File-to-table synchronization](#)
- [COS](#)
- [CAM](#)

Uploading Local Files to a Volume

Last updated: 2026-10-09 16:11:15

Overview

Uploading local files to a volume allows you to upload local files of **any format** (images, audio/video, PDFs, model weights, raw logs, and so on) to a Volume under Catalog as **raw bytes**, preserving the original file structure. After the upload is complete, the files are visible in the [Data Catalog](#) as Volume resources and can be consumed by downstream consumers such as Notebooks, Python jobs, and Agents.

Unlike [local upload to a table](#), this process **does not parse files or generate structured tables**:

Dimension	Upload local files to a table	Upload local files to a volume
File Processing Method	Parse files into structured fields by file type	Store as raw bytes
File Type	Only certain file types	Any format
Whether Computing Resources Are Required	Yes (for parsing)	No
Maximum Number of Files per Upload	10	100
Maximum Size per File	(Total size limit: 2 GB)	5 GB (per file)
Target Location	Catalog → Schema → Table	Catalog → Schema → Volume

Applicable Scenarios

- **Machine learning sample ingestion:** Upload labeled images, audio, and text corpora as original files to a Volume for direct reading and training by Notebooks.
- **Models and configuration files:** Upload model artifacts such as model weights (`.pt` / `.pkl`) and configuration files (`.yaml`).
- **Stage raw logs:** Place the raw logs to be parsed into a Volume first, and then parse them into a table through a Python job in a Workflow.
- **Image and document archiving:** Upload unstructured assets such as PDFs, contracts, and scanned documents in batches.

Operation Steps

Step 1: Entering File Upload to Volume

In the console, go to the **Data Integration** module. On the Access Method Tab, click the **File Upload to Volume** card. The system displays the File Upload to Volume dialog box.

Step 2: Selecting a File

Two methods are supported:

- **Drag and drop:** Drag files directly to the upload area.
- **Click to select:** Click the **Select File** button to select one or more files in the system file browser.

Constraint item	Limit	Validation message
Number of files	At least 1, up to 100	You can upload up to 100 files.
Single File Capacity	≤ 5 GB	A single file cannot exceed 5 GB.
File Type	Any format	Extension name validation is not performed.

Complete error list:

Scenario	Note.
File quantity exceeded	Upload failed. You can upload up to 100 files.
Single file exceeds size limit.	Upload failed. A single file cannot exceed 5 GB.
Network disconnected.	Upload failed. The network connection is disconnected.
Permission changed.	Upload failed. No permission to perform the upload service.
Fallback	Upload failed. {Error details}.

After files are added to the list, you can:

- Click **X** to remove a single file.
- Click **Select File** again to add more files.

Step 3: Selecting a Destination Path

Parameter	Description	Required
Target Path	Select the target Volume. Only Volume type directories can be selected. If no Volume exists in the current Catalog / Schema, create one immediately in the tree directory.	Yes

To create a Volume:

- Hover over the Schema. A + icon appears on the right. Click it to open the creation panel.
- For details about Volume creation, see [Unstructured Data Catalog](#).

Tip

The Volume creation action is **independent of the current upload process**. Even if you cancel this upload after creating a Volume, the created Volume remains in the Catalog.

Parameter	Description	Required
Same-Name Handling	Auto rename (default): generate a new file name <code>file_name_1</code> ; replace existing file : overwrite the existing file with the same name with the latest data.	Yes

Only Catalogs / Schemas / Volumes on which the current account has write permission are displayed. You can use the search box to fuzzy match directory names.

Step 4: Uploading

Click **Upload**:

- The system returns to the Data Integration homepage and displays the **Upload Progress List** in the lower-right corner.
- The progress percentage of each file is displayed as a background bar in the list.

Operations supported by the progress list:

Operation	Description
Go to resource.	After the upload is complete, click the link icon next to the file to go to the Volume details page where the file resides. You cannot go to the page before the upload is complete.
Hide / expand list.	Click the close icon in the upper-right corner to hide the list as a collapsed progress indicator at the top of the page. Click the indicator to expand the list again.

Step 5: Verifying the Result

After the upload is complete, you can:

1. Navigate to the target Volume in [Data Catalog](#) and confirm that the file exists.
2. Read the Volume file by its file path in [Notebook](#).
3. Use a Python File task in [Workflow](#) to read and process files in the Volume.

FAQs

Q: What should I do if a single upload exceeds 100 files or 5 GB?

A: Upload data in multiple batches. If a large amount of raw data is stored in COS or a file system, use [File to Volume Sync](#) to ingest data through periodic offline sync tasks.

Q: Can files uploaded to a volume be parsed back into structured tables?

A: Yes. After uploading, you can create a SQL task or Python task to read Volume files and write them to structured tables. You can also use [File to Table Sync](#) to load files from the Volume into tables in batches.

Q: What happens if I close the browser during upload?

A: Incomplete uploads will be interrupted. When you enter the page again, files that have been successfully uploaded to the Volume are not affected, while interrupted files need to be uploaded again.

Q: Can an upload be resumed after a network interruption?

A: Resumable upload is not supported. If a "Network connection disconnected" message appears, wait for the network to recover and then upload the file again.

Q: Why can't I select the destination path?

A: The target Volume path must at least reach the volume node (the full path is catalog > schema > volume > file directory). You can click the small triangle on the left side of the file tree to expand it until the volume node or file directory node is reached.

References

- [File Upload](#)
- [Upload Local Data to Table](#)
- [Upload COS Files to Table](#)
- [File to Volume Sync](#) (for periodic data ingestion scenarios)
- [Unstructured Data Catalog](#)
- [Data Catalog Overview](#)