# DLT-META JSON Configuration Schemas

*Complete Reference Guide*

Generated: October 23, 2025

# Table of Contents

1. Onboarding JSON Schema (Data Flow Specification)  
 1.1 Source Details Object  
 1.2 CDC Apply Changes Object  
 1.3 Apply Changes From Snapshot Object  
 1.4 Append Flows Array Object  
 1.5 Sink Object (DLT Sink API)  
2. Silver Transformations JSON Schema  
3. Data Quality Expectations (DQE) JSON Schema

# 1. Onboarding JSON Schema (Data Flow Specification)

This is the main configuration file for defining data pipelines in DLT-META.

## Main Configuration Fields

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Type** | **Mandatory** | **Description** |
| data\_flow\_id | String | ✅ Yes | Unique identifier for the pipeline |
| data\_flow\_group | String | ✅ Yes | Group identifier for launching multiple pipelines under single DLT Pipeline |
| source\_system | String | ❌ Optional | Source system identifier (e.g., MYSQL, ORACLE, SAP) |
| source\_format | String | ✅ Yes | Source format: cloudFiles, eventhub, kafka, delta, snapshot |
| source\_details | Object | ✅ Yes | Source-specific configuration (see subsection 1.1) |
| bronze\_catalog\_{env} | String | ⚠️ Recommended | Unity Catalog name for bronze layer (env = dev/prod/staging) |
| bronze\_database\_{env} | String | ✅ Yes | Delta Lake bronze database name |
| bronze\_table | String | ✅ Yes | Delta Lake bronze table name |
| bronze\_table\_comment | String | ❌ Optional | Bronze table comment/description |
| bronze\_reader\_options | Object | ❌ Optional | Spark reader options (e.g., {"header": "true"}) |
| bronze\_partition\_columns | String/Array | ❌ Optional | Bronze table partition columns list |
| bronze\_table\_cluster\_by | Array | ❌ Optional | Bronze table cluster by columns list |
| bronze\_cdc\_apply\_changes | Object | ❌ Optional | Bronze CDC apply changes configuration (see subsection 1.2) |
| bronze\_apply\_changes\_from\_snapshot | Object | ❌ Optional | Bronze apply changes from snapshot config (see subsection 1.3) |
| bronze\_table\_path\_{env} | String | ⚠️ Conditional | Bronze table storage path (mandatory if UC not enabled) |
| bronze\_table\_properties | Object | ❌ Optional | DLT table properties (e.g., {"pipelines.autoOptimize.managed": "false"}) |
| bronze\_sink | Object | ❌ Optional | DLT Sink API properties for bronze layer (see subsection 1.5) |
| bronze\_data\_quality\_expectations\_json\_{env} | String | ❌ Optional | Path to bronze DQE JSON file |
| bronze\_catalog\_quarantine\_{env} | String | ❌ Optional | Unity Catalog name for quarantine data |
| bronze\_database\_quarantine\_{env} | String | ❌ Optional | Database for quarantine data that fails expectations |
| bronze\_quarantine\_table | String | ❌ Optional | Table name for quarantine data |
| bronze\_quarantine\_table\_comment | String | ❌ Optional | Quarantine table comment |
| bronze\_quarantine\_table\_path\_{env} | String | ❌ Optional | Quarantine table storage path |
| bronze\_quarantine\_table\_partitions | Array | ❌ Optional | Quarantine table partition columns |
| bronze\_quarantine\_table\_cluster\_by | Array | ❌ Optional | Quarantine table cluster columns |
| bronze\_quarantine\_table\_properties | Object | ❌ Optional | DLT table properties for quarantine table |
| bronze\_append\_flows | Array | ❌ Optional | Bronze table append flows configuration (see subsection 1.4) |
| silver\_catalog\_{env} | String | ⚠️ Recommended | Unity Catalog name for silver layer |
| silver\_database\_{env} | String | ⚠️ Conditional | Silver database name (mandatory for silver layer) |
| silver\_table | String | ⚠️ Conditional | Silver table name (mandatory for silver layer) |
| silver\_table\_comment | String | ❌ Optional | Silver table comment/description |
| silver\_partition\_columns | Array | ❌ Optional | Silver table partition columns list |
| silver\_table\_cluster\_by | Array | ❌ Optional | Silver table cluster by columns list |
| silver\_cdc\_apply\_changes | Object | ❌ Optional | Silver CDC apply changes configuration (see subsection 1.2) |
| silver\_apply\_changes\_from\_snapshot | Object | ❌ Optional | Silver apply changes from snapshot configuration |
| silver\_table\_path\_{env} | String | ⚠️ Conditional | Silver table storage path (mandatory if UC not enabled for silver layer) |
| silver\_table\_properties | Object | ❌ Optional | DLT table properties for silver table |
| silver\_sink | Object | ❌ Optional | DLT Sink API properties for silver layer |
| silver\_transformation\_json\_{env} | String | ⚠️ Conditional | Path to silver transformation JSON file (mandatory for silver layer) |
| silver\_data\_quality\_expectations\_json\_{env} | String | ❌ Optional | Path to silver DQE JSON file |
| silver\_append\_flows | Array | ❌ Optional | Silver table append flows configuration |

## 1.1 Source Details Object

### For CloudFiles Source Format:

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Type** | **Mandatory** | **Description** |
| source\_database | String | ❌ Optional | Source database name |
| source\_table | String | ❌ Optional | Source table name |
| source\_path\_{env} | String | ✅ Yes | Source file path for CloudFiles |
| source\_schema\_path | String | ✅ Yes | Path to DDL schema file (Spark DDL format) |
| source\_metadata | Object | ❌ Optional | Metadata configuration for \_metadata columns |
| source\_metadata.include\_autoloader\_metadata\_column | String | ❌ Optional | "True"/"False" to add \_metadata column |
| source\_metadata.autoloader\_metadata\_col\_name | String | ❌ Optional | Rename \_metadata column (default: "source\_metadata") |
| source\_metadata.select\_metadata\_cols | Object | ❌ Optional | Extract columns from \_metadata (e.g., {"input\_file\_name": "\_metadata.file\_name"}) |

### For EventHub Source Format:

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Type** | **Mandatory** | **Description** |
| source\_schema\_path | String | ✅ Yes | Path to DDL schema file |
| eventhub.accessKeyName | String | ✅ Yes | EventHub access key name |
| eventhub.accessKeySecretName | String | ✅ Yes | EventHub access key secret name |
| eventhub.name | String | ✅ Yes | EventHub name |
| eventhub.secretsScopeName | String | ✅ Yes | Databricks secrets scope name |
| eventhub.namespace | String | ✅ Yes | EventHub namespace |
| eventhub.port | String | ✅ Yes | EventHub port (typically "9093") |
| kafka.sasl.mechanism | String | ✅ Yes | SASL mechanism (typically "PLAIN") |
| kafka.security.protocol | String | ✅ Yes | Security protocol (typically "SASL\_SSL") |

### For Snapshot Source Format:

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Type** | **Mandatory** | **Description** |
| source\_path\_{env} | String | ✅ Yes | Path to snapshot files |
| snapshot\_format | String | ✅ Yes | Snapshot file format (e.g., "csv", "parquet") |

## 1.2 CDC Apply Changes Object

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Type** | **Mandatory** | **Description** |
| keys | Array | ✅ Yes | Primary key columns for CDC operations |
| sequence\_by | String | ✅ Yes | Column used for ordering changes (e.g., timestamp) |
| scd\_type | String | ✅ Yes | Slowly Changing Dimension type: "1" or "2" |
| where | String | ❌ Optional | Filter condition for applying changes |
| ignore\_null\_updates | Boolean | ❌ Optional | Whether to ignore null updates (default: false) |
| apply\_as\_deletes | String | ❌ Optional | Expression to identify delete operations (e.g., "operation = 'DELETE'") |
| apply\_as\_truncates | String | ❌ Optional | Expression to identify truncate operations |
| column\_list | Array | ❌ Optional | Specific columns to include in CDC |
| except\_column\_list | Array | ❌ Optional | Columns to exclude from CDC (e.g., ["operation", "\_rescued\_data"]) |
| track\_history\_column\_list | Array | ❌ Optional | Columns to track history for (SCD Type 2) |
| track\_history\_except\_column\_list | Array | ❌ Optional | Columns to exclude from history tracking |
| flow\_name | String | ❌ Optional | Custom flow name |
| once | Boolean | ❌ Optional | Process data once (default: false) |
| ignore\_null\_updates\_column\_list | Array | ❌ Optional | Specific columns to ignore null updates |
| ignore\_null\_updates\_except\_column\_list | Array | ❌ Optional | Columns to exclude from ignore null updates |

## 1.3 Apply Changes From Snapshot Object

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Type** | **Mandatory** | **Description** |
| keys | Array | ✅ Yes | Primary key columns for snapshot tracking |
| scd\_type | String | ✅ Yes | Slowly Changing Dimension type: "1" or "2" |
| track\_history\_column\_list | Array | ❌ Optional | Columns to track history for (SCD Type 2) |
| track\_history\_except\_column\_list | Array | ❌ Optional | Columns to exclude from history tracking |

## 1.4 Append Flows Array Object

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Type** | **Mandatory** | **Description** |
| name | String | ✅ Yes | Unique name for the append flow |
| source\_format | String | ✅ Yes | Source format for append flow |
| create\_streaming\_table | Boolean | ✅ Yes | Whether to create streaming table (true/false) |
| source\_details | Object | ✅ Yes | Source details (same structure as main source\_details) |
| comment | String | ❌ Optional | Append flow comment/description |
| reader\_options | Object | ❌ Optional | Spark reader options |
| spark\_conf | Object | ❌ Optional | Spark configuration for this flow |
| once | Boolean | ❌ Optional | Process data once (default: false) |
| target | String | ❌ Optional | Target table name for silver append flows |

## 1.5 Sink Object (DLT Sink API)

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Type** | **Mandatory** | **Description** |
| name | String | ✅ Yes | Sink name identifier |
| format | String | ✅ Yes | Sink format: delta, kafka, eventhub |
| options | Object | ✅ Yes | Format-specific options (e.g., {"tableName": "catalog.schema.table"} for delta) |
| select\_exp | Array | ❌ Optional | SQL expressions to apply before sinking |
| where\_clause | String | ❌ Optional | Filter condition before sinking |

# 2. Silver Transformations JSON Schema

This file defines SQL transformations for silver layer tables.

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Type** | **Mandatory** | **Description** |
| target\_table | String | ✅ Yes | Target silver table name for transformation output |
| select\_exp | Array | ✅ Yes | Array of SQL expressions/column selections (e.g., ["concat(first\_name,' ',last\_name) as full\_name"]) |
| target\_partition\_cols | Array | ❌ Optional | Partition columns for the target table |
| where\_clause | Array | ❌ Optional | Array of filter conditions to apply (e.g., ["country = 'United States'"]) |

### Example:

[  
 {  
 "target\_table": "cars\_usa",  
 "select\_exp": [  
 "concat(first\_name,' ',last\_name) as full\_name",  
 "country",  
 "brand",  
 "model"  
 ],  
 "where\_clause": ["country = 'United States'"]  
 }  
]

# 3. Data Quality Expectations (DQE) JSON Schema

This file defines data quality rules for bronze or silver layers.

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Type** | **Mandatory** | **Description** |
| expect | Object | ❌ Optional | Rules where failing records are INCLUDED in target dataset. Key=rule name, Value=SQL condition |
| expect\_or\_fail | Object | ❌ Optional | Rules where failing records HALT pipeline execution. Key=rule name, Value=SQL condition |
| expect\_or\_drop | Object | ❌ Optional | Rules where failing records are DROPPED from target dataset. Key=rule name, Value=SQL condition |
| expect\_or\_quarantine | Object | ❌ Optional | Rules where failing records go to quarantine table (Bronze layer only). Key=rule name, Value=SQL condition |

### Example:

{  
 "expect\_or\_drop": {  
 "no\_rescued\_data": "\_rescued\_data IS NULL",  
 "valid\_id": "id IS NOT NULL"  
 },  
 "expect\_or\_quarantine": {  
 "quarantine\_rule": "\_rescued\_data IS NOT NULL OR id IS NULL OR amount=0"  
 }  
}

# Notes

* {env} placeholder represents environment suffix: \_dev, \_prod, \_staging, \_it, etc.
* Mandatory Fields (✅): Must be provided in all cases
* Conditional Fields (⚠️ Conditional): Required based on context (e.g., bronze\_table\_path\_{env} is mandatory only when Unity Catalog is not enabled; silver fields are mandatory when defining silver layer)
* Recommended Fields (⚠️ Recommended): Highly recommended for Unity Catalog environments
* Optional Fields (❌): Can be omitted
* All JSON files must be valid JSON format and accessible from the Databricks workspace
* Schema files referenced in source\_schema\_path should be in Spark DDL format
* Source code validation occurs in src/onboard\_dataflowspec.py and src/dataflow\_spec.py