# Qlik Compose for Data Lakes 6.5 Release Notes - November 2019

This version of Compose for Data Lakes introduces new features and enhancements including support for schema evolution in Databricks projects, and silent installation.

In these release notes:



## Attunity Product Compatibility

Attunity Compose for Data Lakes 6.5 is compatible with the following Attunity products only:

- **Attunity Replicate**: Versions 5.5, 6.5, 6.4, and 6.3 during the grace period
- **Attunity Enterprise Manager (AEM)**: Version 6.5

## Upgrading Attunity Compose

In Spark projects, after upgrading from an Attunity Compose for Data Lakes version prior to 6.5, the paths in the **Provisioning Root** of the **Defaults** tab of the project settings window will be reset to their defaults. It is recommended to copy any paths that were changed before upgrading, and paste them back after the upgrade.

## New Databricks Project Type with Schema Evolution

Added support for the Databricks Delta project type with Schema Evolution.

**Note** The Databricks Delta project type is currently in beta and is supported with Databricks 5.3 only.

Schema evolution enables you to ensure that the tables in the Storage Zone are up-to-date with the latest changes to the source schema. Compose for Data Lakes checks for any changes to the source schema whenever the task is run (manually or scheduled). On detecting a change, Compose for Data Lakes will update and validate the project metadata, generate the task instructions, and then run the task.

## Support for Silent Installation

Compose for Data Lakes can now be installed silently (i.e. without requiring user interaction). This is useful if, for example, you need to install Compose for Data Lakes on several machines throughout your organization. Silent upgrade and uninstallation of Compose for Data Lakes are also supported.

## Newly Supported Platform Versions

Support was added for the following:

- Hortonworks 3.1.x as a distribution platform for Hive projects.

  **Note** For Apache Spark project types, the Hortonworks cluster should be configured to set the Spark metadata store to the Hive metadata store.

- ADLS GEN2 HDInsight
- Cloudera 6.1 is now supported as a distribution platform for Spark and Hive projects.

  **Notes**

  - After installing the Cloudera JDBC driver for an Apache Hive project, you need to set the environment variable:

    `ATTUNITY_HIVE_DRIVER_CLASS_NAME`

    to:

    `com.cloudera.hive.jdbc41.HS2Driver`

- When provisioning external tables for Spark projects on Cloudera, the AVRO file format is not supported.

## Newly Supported Replicate Endpoints

Support for the following Replicate target endpoints was added:

Amazon EMR, Microsoft Azure HDInsight, and Google DataProc.

## Enhancements

- Enhanced performance in the discovery and generation of metadata. When working with a large number of tables, only metadata that is relevant for the current project will be provisioned.
- When using an Incrementally Updated ODS task type for provisioning, the Current View will now be created without header columns and the position of the Primary Key columns will correspond to their position in the metadata.
- An option (check box) has been added to remove the **header_modfied_batch** column from newly provisioned Operational Data Store (ODS) Views. This allows you to create provisioned Views identical to the source tables (i.e. without any additional columns).
- An option to change the View prefix was added (to the project settings), thereby enabling the View to be created with the same name as the storage tables.

# Resolved Issues and Customer Requested Enhancements

The following are the resolved issues and customer requested enhancements in this release.

| Component/Process | Type | Description | Ref # |
|---|---|---|---|
| Metadata Expression | Issue | When an expression was used for an attribute in a Primary Key, the alias in the generated code would be incorrect. | 193511 |
| Change Processing | Issue | With some Hive versions, the partition name would be reset if the CDC task completed without any changes. | 195234 |

| Component/Process | Type | Description | Ref # |
|---|---|---|---|
| Change Processing | Issue | **Note** The following is only relevant if you are ***not*** upgrading from a Compose for Data Lakes 6.4 Service Pack. That is to say, it's only relevant if you are upgrading from the GA release.<br><br>When multiple Replicate tasks would write to the same Landing Zone, some of the partition data would not be applied to the Staging tables during Change Processing.<br><br>To implement this fix, you need to run a script in the project database. The script basically replaces the **attcmps_ddl_history** Control Table with a modified version of the table, which is partitioned by the Replicate task name.<br><br>The old Control Table is not dropped immediately; rather, its name is changed to **OBSOLETE_attcmps_ddl_history**. After you have verified that all tasks are running without issue, you can go ahead and drop the table.<br><br>Instructions for performing the aforementioned tasks are provided below. | 193452 |
| Engine | Issue | Errors would sometimes occur when running tasks in a Spark project, as a result of too many files being open. | 193295 |
| Upgrade | Issue | Generating instructions for CDC tasks would sometimes take a very long time, occasionally resulting in timeout errors. | 193294 |

| Component/Process | Type | Description | Ref # |
|---|---|---|---|
| CLI | Issue | Running a Hive project Full Load or Change Processing tasks with the CLI would sometimes fail with the following error:<br>`Task 'task_name' either does not exist or is not a STORAGE task. ComposeForDataLakes Control Program completed with error.` | 193538 |
| Schema Evolution | Issue | In some scenarios, when using schema evolution, the Compose task would fail when writing rows.<br>The following error would be displayed:<br>`org.apache.spark.SparkException: Task failed while writing rows.` | 190160 |
| Discovery | Issue | When the columns with the same name appeared in different tables, the attribute names were not preserved during discovery. | 187832<br>161075<br>188383 |
| Schema Evolution | Issue | When a "truncate DDL" operation was recorded in the ddl_history Control Table, schema evolution would fail with the following error :<br>`[Metadata ] [ERROR] Read replicate ddl changes error: SYS-E-JSONEMPTY, JSON is null or empty` | 186463<br>186878 |
| Generate | Issue | When two tables used the same source or when an entity was duplicated, an error would occur when generating instructions for a CDC task. | 185270 |

| Component/Process | Type | Description | Ref # |
|---|---|---|---|
| View Generation | Issue | Parts of the SQL syntax used by Compose were not compatible with Apache Impala (which is not officially certified for use with Compose), resulting in failure to generate Views.<br><br>The issue was resolved by modifying the Compose SQL syntax to be more compatible with Apache Impala. | 184753 |
| Workflow | Issue | The error port was missing from the workflow task element. | 186106 |
| CDC-Only Tasks | Issue | When running CDC-only tasks, Compose would sometimes fail to discard timed out connections from the Connections Cache.<br><br>Compose would then reuse the timed out connection repeatedly, resulting in continuous timeout errors. | 183306 |
| Provisioning - Spark project | Enhancements | See Enhancements above. | 193922 |
| Performance | Issue | Due to a caching issue:<br>» CDC tasks would take a long time to complete.<br>» The Schema Evolution window would remain open for too long. | 195595 |
| Discovery | Issue | Metadata discovery would take a long time as Compose would extract columns from all the tables in the source schema instead of just the selected tables. | 193939 |

| Component/Process | Type | Description | Ref # |
|---|---|---|---|
| Hadoop platform | Issue | When multiple deletions were performed on the same ID in a Control Table, the Compose task would fail with the following error:<br>`Cardinality Violation in Merge statement` | 195286 |

# Known Issues

The following are the known issues in this release.

| Component/Process | Description | Ref # |
|---|---|---|
| Provisioning Tasks | If you are using Amazon EMR Hive distribution version 5.20.0 or higher, the value for the `spark.sql.parquet.fs.optimized.committer.optimization-enabled` parameter is set by default to be **True**. Before running a Spark provisioning task, you must configure this parameter's value to be **False** - i.e. `spark.sql.parquet.fs.optimized.committer.optimization-enabled=false` | 191640 |
| Schema Evolution | When a new column is added to the model based on the source data type instead of the target data type, an error is displayed on validation. | CMPS-8295 |
| Metadata | In an Apache Spark project, when deleting an attribute from the Metadata and then adding back the same attribute to the Metadata, the affected tables need to be dropped and recreated. | CMPS-7862 |
| Provisioning Tasks | Mapping from multiple sources is not supported, in both Spark and Hive projects. | CMPS-7770 |
| Provisioning Tasks | In Spark projects, Full Load overwrites previous data, so when running two tasks on the same target, the second task overwrites the first task results. Splitting the Full Load into several tasks also doesn't work for the same reason. | CMPS-7457 |

| Component/Process | Description | Ref # |
|---|---|---|
| Schema Evolution | When a table is added to the source, the new table is added to the task (the mapping is created correctly and associated with the task), and the table is populated with data.<br><br>However, when opening the mapping, there is an error that the source table does not exist in the database.<br><br>**Workaround:**<br><br>Click **Clear Cache** in the **Manage Data Storage Tasks** window (and then open the mapping again). | CMPS-7210 |
| Knox | In the **Knox gateway path** field, Compose for Data Lakes automatically appends "/hive" to the specified path. As users are unaware of this, they will also specify the path with "/hive" (resulting in "/hive/hive") causing the connection to fail. | CMPS-6626 |
| Provisioning Tasks - Mapping | Two Data Storage Change Processing tasks with different sources that have mapping to the same entity, will result in incorrect data in incremental provisioning to ODS or HDS. | CMPS-6545 |
| Identical task names across Compose projects | Multiple Compose projects with tasks that have the same name are currently not supported in AEM Metadata. | CMPS-6524 |
| Generate | After performing a non-supported change in the metadata, regenerating the task instructions will appear to succeed without errors or warnings, but the task will fail if run later. | CMPS-6437 |
| Hard delete | Hard delete performed on records with expressions, lookup, or derived attributes on Primary Keys does not work. | CMPS-5480 |
| UI | When displaying an entity in the **Physical Metadata** tab and going back to **Logical Metadata** tab, the entity initially appears without any attributes. | CMPS-5421 |

| Component/Process | Description | Ref # |
|---|---|---|
| Column Renaming | Renaming a column in Parquet or Avro format will cause loss of all data in that column. | CMPS-5416 |
| Schema Change | After running a Compose for Data Lakes Change Processing task and a schema change occurs, if you run the Compose Full Load task again, the task will fail.<br>**Workaround:**<br>Run the Replicate Full Load again. | CMPS-5394 |
| Replicate Control Tables | If Replicate's **attrep_ddl_history** and **attrep_history** Control Tables are not in the same schema, Compose fails at runtime. | CMPS-5346 |
| Derived Attribute | A statement error occurs when changing the name of an attribute that is included in a derived attribute expression. | CMPS-5178 |
| Multiple Landing Zones | Tasks that ingest data from several Landing Zones fail during generation of task instructions.<br>Workaround:<br>Create several tasks - one for each Landing Zone. | CMPS-5159 |
| Validation | Validation of the Data Lake does not detect Compose columns (e.g. FROM_DATE) that have been renamed. | CMPS-4963 |
| Spark Provisioning | When a Spark project is defined with a Microsoft Azure Data Lake Storage Gen1 data store, defining HDFS as a provisioning target is not currently supported. | CMPS-7859 |

# Generating and Running the Upgrade Script

**Note** The following is only relevant if you are ***not*** upgrading from a Compose for Data Lakes 6.4 Service Pack. That is to say, it's only relevant if you are upgrading from the GA release.

1. Install Compose for Data Lakes 6.5.
2. From the Windows **Start** menu, open **Attunity Compose for Data Lakes** >

**Compose for Data Lakes Command Line** and run the following command:

```
composecli generate_upgrade_scripts
```

A script is created for each Hive project under `data\projects\<project name>\ddl-scripts`.

The script you need to run for each of your Hive projects is called:

```
ComposeUpgradeFrom6.4To6.4SP1_<hive_project_name>__<timestamp>.sql
```

3. Leave the Compose for Data Lakes Command Line open as you will need it to generate the task ETLs (described below)
4. In the project database, run the script(s) for each of your Hive projects.

Once you are sure that the new Control Tables are working without issue, you can drop the old Control Table (**OBSOLETE_attcmps_ddl_history**).

## Generating Task ETLs

**To generate the task ETLs:**

1. Run the following command:

   ```
   ComposeCli.exe connect
   ```

2. Generate all ETLs on Compose for Data Lakes 6.4 by running the following command:

   ```
   ComposeCli.exe generate_etls
   ```

   Any invalid tasks will be skipped and an appropriate error will be printed to the output.

   **Notes**

   - The ETL generation process may take a while (depending on the number of tasks and projects) as Compose for Data Lakes needs to connect to each of the relevant databases.
   - If you prefer, you can regenerate the ETL instructions manually for each task. Note however that a task will not be able to run until its ETL instructions are regenerated.