Attunity Compose for Data Lakes 3.3 Release Notes - October 2017

Attunity Compose is a family of products that accelerate and simplify data warehouse design, development, testing, deployment and updates. Attunity Compose for Data Lakes, and specifically Attunity Compose for Hive, automate the data pipeline to create analytics-ready data on Hive.

Attunity Compose for Data Lakes is a separately licensed product. Customers who are already licensed for Attunity Replicate or Attunity Compose and are interested in evaluating Attunity Compose for Data Lakes, should contact Attunity Support or their Attunity Account Manager for more details.

Attunity Compose for Data Lakes 3.3 introduces several new features and enhancements including schema evolution, Hive partitioning, and Sqoop Incremental Import (Load).

In these release notes:

Automatic Schema Evolution

With Automatic Schema Evolution, Compose for Data Lakes automatically updates the target with changes to the source schema. This ensures that the Delivery Zone tables always reflects the latest supported changes to the source schema.

When this option is enabled, Compose for Data Lakes will check 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.

The following DDL changes are supported:

- Add column
- Create table
- Drop column

Note Automatic Schema Evolution is currently in Beta status.

Support for Hive Partitions and Buckets

Compose for Data Lakes 3.3 introduces support for Hive partitioning, which is a way of dividing a table into related parts based on the values of partitioned columns such as date, city, and department. Using Hive partitioning in the right context and on appropriate columns makes it easier to query a portion of the data.

Hive partitions are dynamic created, eliminating the need to manually create the actual partition directories ahead of time.

Note that partition keys are not physically stored as columns in Hive, but rather, as directory names.

Tables or partitions are sub-divided into buckets, to provide extra structure to the data that may be used for more efficient querying. Bucketing works based on the value of hash function of some column of a table.

This new functionality is available in the **Partition Key** and **Bucket Key** tabs in the bottom right of the **Physical Metadata** tab. Note that, with the exception of primary keys that have been allocated as partition keys, all of a table's primary key columns are automatically added to the **Bucket Key** tab (which must contain at least one primary key column).

The following limitations apply:

- Updates on partition keys are not supported. If a record that includes a partition key is updated in the source, the updated partition key will be ignored, but the rest of the fields will be updated.
- Updates on bucket keys are not supported. If a record that includes a bucket key is updated in the source, a new record will be inserted into the target.

Soft and Hard Delete

Customers can now choose what action Compose for Data Lakes should perform in the Delivery Zone when DELETE operations are performed on the source tables.

To this end, the following options have been added to the **New Project** window:

When a record is deleted from the source database:

- Mark the matching Delivery Zone record as deleted
- Mark the matching Delivery Zone record as deleted in historical tables, but delete the record from other tables
- Do nothing

Source Variables and Data Consolidation from Identical Sources

Compose for Data Lakes 3.3 introduces support for source variables, which facilitates data consolidation from multiple identical sources.

Any variables that you wish to use in your project should be defined in the new **Variables** tab in the **Project Settings** window. The defined variables will be displayed in the **Source Landing Zone** definitions (under the **Variables** heading), allowing you to provide values (i.e. data) for each of the variables.

You can then create a new attribute in the Delivery Zone table and define an expression that replaces the variables with the specified data during data ingestion. Although variables can be used for a variety of purposes, they are especially useful if you need to ingest data from several identical sources (in terms of table metadata) into a single, uniform table. For instance, if an organization has several factories and wishes to consolidate their data into a single Delivery Zone table, you could setup a project that replaces the variables with the location of each of the factories.

Support for Multiple Replicate Servers

This version adds support for multiple Replicate Servers. Now, when configuring a Source Landing Zone, customers with multiple Replicate Servers can choose from which Replicate Server to associate a Replicate task.

Support for Amazon EMR

In addition to Hortonworks and Cloudera that are already supported by Compose for Data Lakes, this version introduces support for Hive on Amazon EMR including the ability to store the Delivery Zone files on Amazon S3.

Sqoop Incremental Import (Load)

A **Sqoop Incremental Import (Load)** option has been added to the Task Settings **General** tab. Select this option if you need to apply source changes to Hive, but the source - for example, Apache Sqoop - does not support automated processes such as CDC or reading the database transaction logs. In this case, during the task, Compose for Data Lakes will query user-designated Landing Zone columns (of type DATETIME) for changes and, on detecting a change, add a new version to the target record.

When this option is selected, a Landing Zone column of type DATETIME - for example, "Last Modified" must be mapped to the FROM__DATE column in the corresponding Delivery Zone table(s).

Note Currently, to perform "Sqoop Incremental Import (Load)" on several Landing Zones, a separate task needs to be run for each of the Landing Zones.

UI Improvements

- Added the Export to TSV option to several lists
- Added the ability to customize more column names to the project settings
- Added an option to hide Non-SQL commands to the **Task Commands** window.
- Added a Description column to the **Attributes Domain** window
- When configuring a **New Attribute**, users now have the option to:
  - Make the attribute a partition key (see Support for Hive Partitions and Buckets)
  - Create an expression that can be used, among others, to make the attribute function as a derived attribute
  - Automatically add the attribute to all tables
  - Select where the attribute should be located in the table(s)

Resolved Issues and Customer Requested Enhancements

The tables below lists the resolved issues and customer-requested enhancements for this release.

Component/Process	Type	Description	Ref #
Data Consolidation	Enhancement	See Source Variables and Data Consolidation from Identical	CMPS-4906

Component/Process	Type	Description	Ref #
		Sources.	
Drop and Recreate	Issue	When dropping and creating tables, all views would be dropped instead of only the views of the related tables.	CMPS-4869
Scheduler	Issue	A task run conflict would occur if the next scheduled task instance was due to run, but the previous instance was still running.	CMPS-5211

Known Issues

The following are the known issues in this release.

Component/Process	Description	Ref #
Automatic Schema Evolution	If "Automatic Schema Evolution" is enabled and changes to the schema occurs *after* Replicate Full Load completes but *before* Compose Full Load is run, then the Compose Full Load and CDC tasks will not take these changes into account. Workaround: Run Replicate Full Load again or manually apply the schema changes that occurred after the Replicate Full Load completed.	CMPS-5377
	After running a Compose CDC task and a schema change occurs, if you run the Compose Full Load task again, the task will fail. Workaround: Run Replicate Full Load again.	CMPS-5394
	Renaming a column in Parquet or Avro format will cause loss of all data in that column.	CMPS-5416
	Archive tables are currently not affected by automatic schema evolution. For example, a column added to the source will be added to the Change Table but not to the archived Change Tables. Workaround: Update their structure manually.	CMPS-5439
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
Scheduler	When recreating the target tables and generating new task instructions, Compose does not stop the scheduler. This causes the "Run" operation to fail. Workaround: Disable the scheduler for the specified task before generating the task instructions (as described in the "Scheduling Tasks" section in Help), generate the task instructions, and finally, enable the scheduler.	CMPS-5335
Monitor	Task run details of the currently completed task are shown for *all* tasks in the monitor's **History** tab.	CMPS-5334

Component/Process	Description	Ref #
Scheduling	CDC can be scheduled to run (and will run) while Full Load is still running.	CMPS-5333
Expression for Attribute	When creating an expression for an attribute, clicking **OK** does not parse the expression, which may result in an invalid expression. The workaround is to click the **Parse Expression** button before clicking **OK**.	CMPS-5298
Validation	Validation of the Delivery Zone does not detect Compose columns (e.g. FROM_DATE) that have been renamed.	CMPS-4963
Mappings	Multiple mappings associated with the same table causes an error when generating instructions for CDC tasks.	CMPS-5098
Multiple Landing Zones	Tasks that ingest data from several Landing Zones fail during generation of task instructions. Workaround: Create several tasks - one for each Landing Zone.	CMPS-5159
Derived Attribute	A statement error occurs when changing the name of an attribute that is included in a derived attribute expression.	CMPS-5178
Hard Delete	Tasks fails if the table is configured to use hard deletes and one of the primary keys consists of an attribute with an expression or a derived attribute.	CMPS-5480
Drop and Create	When large tables are being dropped, Drop and Create sometimes fails the first time. Workaround: Retry the operation until it succeeds.	CMPS-5427
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
Discover and Generate	Discover and Generate read all the tables in the database, regardless of the selection. This may take a while if the database contains numerous tables.	CMPS-5332
UI - Mappings	Multiple mappings for same table is not supported for CDC tasks.	CMPS-5098
Drop and Create	Drop and recreate fails after changing the Views prefix or suffix in the project settings. Workaround: Drop the tables first, change the project settings and then recreate.	CMPS-4649

Component/Process	Description	Ref #
Project Settings	Drop and recreate tables is required after modifying the **Exclude the "To Date" column from tables with history** setting.	CMPS-4492
Project Reset	Archived tables are sometimes not dropped when performing a project reset. This usually occurs if two project resets were performed in succession with only the second reset configured to drop archived tables.	CMPS-4198
Discovery	Discovering a table with a space in one of its column names does not create the mapping for that column. Workaround: Create the table manually	CMPS-3294
Amazon EMR Landing Zone - Change Tables Settings	When the **Delete the Change Tables** option is selected in the Source Landing Zone settings, Compose fails to run tasks.	CMPS-5487