

Qlik Compose August 2021 Release Notes (including SR01 and SR02)



Skipping versions: Customers who are not upgrading directly from the previous version are strongly encouraged to review the release notes for all versions higher than their currently installed version.

In these release notes:

- *Migration and upgrade (page 2)*
- *What's new? (page 13)*
- *End of Life/Support and Deprecated Features (page 20)*
- *Resolved Issues (page 21)*
- *Known Issues (page 37)*

For more information about a particular feature, please refer to the Compose Help.

1 Migration and upgrade

This section describes various upgrade scenarios and considerations.

1.1 Compose for Data Warehouses upgrade path

Direct upgrade is supported from Compose February 2021 or Compose May 2021 only.

When upgrading from Compose for Data Warehouses, the upgrade path is as follows:

Upgrading from supported versions

- Customers upgrading from 6.5 or 6.6 should first upgrade to 6.6.1, then to Compose February 2021, and finally to Compose August 2021.
- Customers upgrading from 6.6.1 should first upgrade to Compose February 2021, and then to Compose August 2021.

Upgrading from unsupported versions

- Customers upgrading from 6.3 or 6.4 should first upgrade to 6.5, then to 6.6.1, then to Compose February 2021, and finally to Compose August 2021.
- Customers upgrading from 3.1 should contact Qlik Support.

See also:

For information on the procedure for upgrading from Compose for Data Warehouses to Compose February 2021, see the [February 2021 release notes](#).

For information on upgrading from Compose for Data Lakes, see *Migration from Compose for Data Lakes (page 5)*.

1.2 Required post-upgrade actions when upgrading from Qlik Compose

ETL script enhancements

After upgrading, in order to benefit from the latest enhancements to the task ETL scripts:

- Customers with Data Warehouse projects should regenerate all task ETLs either by selecting the task and clicking the **Generate** button in the **Manage Tasks** and **Manage Data Marts** windows, or by running the `generate_project` CLI as described in the [Compose online help](#).
- Customers with Data Lake projects should regenerate all task ETLs by selecting the task and clicking the **Generate** button in the **Manage Storage Tasks** window, or by running the `generate_project` CLI as described in the [Compose online help](#).

Upgrade scripts

After upgrading, you need to run two upgrade scripts.

Upgrade script 1



Should be run when upgrading to Compose August 2021 or later.

Various performance enhancements in this release require modifications to the internal Compose tables in the following data warehouses:

- Microsoft SQL Server
- Oracle
- Microsoft Azure Synapse Analytics
- Google Cloud BigQuery
- Amazon Redshift

If you have Data Warehouse projects configured to use any of the above databases, you need to generate an upgrade script and then run it in each of the relevant databases.



Running the script in Google Cloud BigQuery and Amazon Redshift databases will delete historical monitoring metadata.

Upgrade script 2



Should be run when upgrading to Compose August 2021 Service Release 02 or later.

This upgrade script must be run after upgrading as the database structure has been slightly modified to correctly report the error mart for each source (as part of the *Uniform source consolidation (page 13)* feature).

Generating and running the upgrade scripts

1. From the **Start** menu, open the **Compose Command Line** console and run the following command:

```
ComposeCli.exe connect
```

2. Run the following command:

```
ComposeCli.exe generate_upgrade_scripts
```

For each of your projects, the CLI output will tell you the name of the script and its location. Each script has a different name, consisting of the script identifier (the bold part), the project name, and a timestamp.

Example of Upgrade script 1:

```
C:\Program Files\Qlik\Compose\data\projects\Project_1\ddl-scripts\ComposeUpgradeFrom2021_5To2021_8Project_1__210714142110.sql
```

Example of Upgrade script 2:

```
C:\Program Files\Qlik\Compose\data\projects\Project_2\ddl-scripts\ComposeUpgradeFrom2021_8SP4To2021_8SP9Project_2__220114142110.sql
```

3. Access each of your databases using SQL Workbench or a similar tool and run the script(s).
4. When the script(s) completes successfully, generate and run your tasks in Compose.

Updating the Views in Data Lake projects

If you are upgrading from a version prior to Qlik Compose February 2021 Patch Release 04, you need to recreate the Compose Views for each of your Data Lake projects. The Views can be recreated using the Compose web console or using the Compose CLI.

If Compose detects a mismatch between the Logical Metadata (defined via the **Metadata** panel) and the Storage Zone metadata, the view recreation operation will fail and you will need to validate and adjust the storage before retrying the operation.

Recreating the Views with the web console

To recreate the Views using the web console, simply select **Recreate Views** from the menu in the top-right of the **STORAGE ZONE** panel. You will be prompted to confirm the operation as it might take some time, during which the Views data might not be accessible.

Recreating the Views with the CLI

To recreate the Views using the CLI:

1. From the **Start** menu, open the **Compose Command Line** console.
2. Run the following command to connect to the server:

```
ComposeCli.exe connect
```
3. When the command completes successfully, run the following command:

```
ComposeCli.exe recreate_views --project project_name
```

Where:

- `--project` is the name of the project for which you need the views to be recreated.

Example:

```
ComposeCli.exe recreate_views --project myproject
```

1.3 Licensing

Existing Compose for Data Warehouses customers who want to create and manage Data Warehouse projects only in Qlik Compose can use their existing license. Similarly, existing Compose for Data Lakes customers who want to create and manage Data Lake projects only in Qlik Compose can use their existing license.

Customers migrating from Qlik Compose for Data Warehouses or Qlik Compose for Data Lakes, and who want to create and manage *both* Data Warehouse projects and Data Lakes projects in Qlik Compose, will need to obtain a new license. Customers upgrading from Compose February 2021 can continue using their existing license.

It should be noted that the license is enforced only when trying to generate, run, or schedule a task (via the UI or API). Other operations such as Test Connection may also fail if you do not have an appropriate license.

1.4 Migration from Compose for Data Lakes

This section explains how to migrate your projects from Qlik Compose for Data Lakes to Qlik Compose. You can migrate both your project definitions and your data although the latter is only required if you need to migrate production data. Before starting, make sure you have installed Qlik Compose on a dedicated machine.



Migration can be performed from Compose for Data Lakes 6.6 only.

The migration process consisting of the following steps:

- *Step 1: Prepare project deployment and data migration in Compose for Data Lakes (page 7)*
- *Step 2: Upgrade your databases to the versions supported by Qlik Compose (page 9)*
- *Step 3: Deploy the projects on Qlik Compose and migrate the data (page 9)*



Before migrating to a production environment, It is strongly recommended to:

- *Back up your storage data*
- *Back up the Compose "data" folder (default location: C:\Program Files\Qlik\Compose\data)*
- *Try out the migration in a test environment first*

Migrating the objects and settings

The following objects and project settings can be migrated:

Supported Objects

Supported Project Settings

- Data sources
 - Attribute domains
 - Attributes Entities
 - Mappings
 - Storage tasks
 - Command tasks
 - Workflows
 - Reusable transformations
 - Lowest Date and Highest Date
 - Generate DDL scripts but do not run them
 - Ignore Mapping Data Type validation
 - Prefix for storage tables
 - Suffix for Replicate Change Tables
-

Limitations and considerations

- Notifications and task schedules will not be migrated.
- The following objects/settings will not be migrated as the corresponding functionality is either different in Qlik Compose or not available:
 - The **Variables** tab in the project settings
 - The **Use buckets** option in the Storage Zone connection settings. This was only relevant for Spark projects and Hive versions preceding 3.0, which are not currently supported in Qlik Compose.
 - Partitions defined in the **Physical Metadata** tab's **Partition Key** sub-tab.
 - The **Mark the matching Storage record as deleted in history tables, but delete the record from other tables** option. Qlik Compose performs "soft" deletes only.
 - The **Archive the Change Tables** option.
 - The Provisioning Zone in Spark projects
 - Schema evolution. Schema evolution has been completely redesigned in Qlik Compose.
 - In the Storage task settings:
 - The **Sqoop incremental import (load)** setting
 - The **If Header__FROM_DATE is not connected** setting
 - The **Default History Resolution** setting will always be according to the input time (and not Daily)
 - Multiple mappings for a single logical entity. The mappings will be imported, but as only one mapping is supported you will need to delete all of the mappings (in Qlik Compose) except the one that you want to keep.
- In Spark projects, a single task performs Full Load and Change Processing. In Qlik Compose, this task will be split into two separate tasks: Full Load and Change Processing. Any workflows with such tasks will need to be modified to reflect the new behavior.
- The default "From Date" and "To Date" column names in Qlik Compose are **header__fd** and **header__td** respectively (where "header__" is a hardcoded prefix used to differentiate Compose columns from standard columns). During migration, the column names defined for your Qlik Compose for Data Lakes projects will be retained (i.e. **header__<existing-name>**). However, any new Data Lake projects you create will use the defaults, unless you override them in the project settings **Naming** tab.

Migrating the data



You should only migrate the data if it is used in a production environment and you have historical records that you need to preserve. Otherwise, it is simpler (and preferable) to simply [reload the Replicate task](#), which will populate the landing database with the most recent data.

The data migration procedure consists of several steps (which differ slightly between Hive projects and Spark projects), including generating a script and then running that script on your storage cluster (designated by the **Host** parameter in the storage settings). The script copies the data from the old storage tables to the newly created storage tables and modifies the data in the internal Compose columns (in the storage tables) to conform to Qlik Compose. The script also copies the change processing context so tasks will be able to resume from the point they were stopped in Compose for Data Lakes.

Limitations and considerations

- The storage contains INSERTED, UPDATED and DELETED records:
 - For HDS entities: INSERTED, UPDATED and DELETED records will be indicated by the letters "I", "U" and "D" respectively.
 - For ODS entities: INSERTED and UPDATED records will be indicated by the letter "I", and deleted records will be indicated by the letter "D".

Step 1: Prepare project deployment and data migration in Compose for Data Lakes

In Compose for Data Lakes, for each of the projects that you want to migrate, do the following:

a). Create the deployment package

1. In the main Compose for Data Lakes window, right-click the desired project and select **Create Deployment Package** from the context menu.
The **Create Deployment Package - <Project_Name>** window opens.
2. Provide a **Version** number and a **Description** in the designated fields and then click **OK**.
Depending on your browser settings, either a ZIP file containing a JSON file (with the project settings) and a **readme.txt** file will be saved to your browser's default download location or you will be prompted to download the ZIP file.
The ZIP file name is in the following format:
<Project_Name>_deployment_<Date>__<Time>.zip
3. Copy the deployment ZIP file to the Qlik Compose machine or make it accessible to the Qlik Compose machine (by placing it in a shared folder, for example).

b). Prepare the Hive project data for migration




Only applicable if you have Hive projects and you need to migrate their data.

Switch to **Monitor** view and run your **Change Processing** (CDC) tasks (by selecting each task and clicking the **Run** button). Wait for the tasks to complete. This will ensure that the storage tables contain the latest updates.

c). Prepare the Spark project data for migration



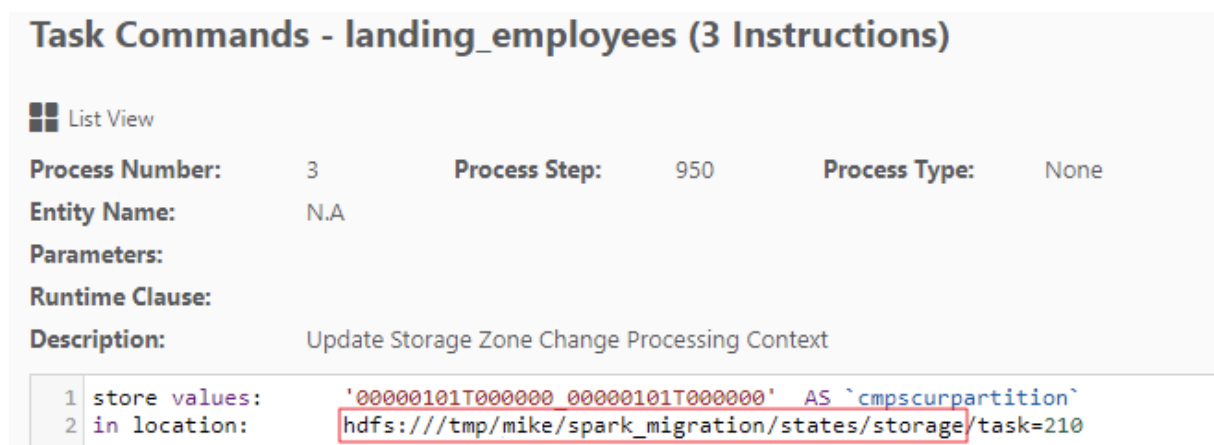
Only applicable if you have Spark projects and you need to migrate their data.

1. Select **File and Table Action > Create External Tables** from the **Storage Zone**  menu. This is required as the option to work with files alone is not available in Hive. After the external storage tables are created, click **Validate External Tables** to validate them.
2. Switch to **Monitor** view and:
 - a. Run your **Change Processing** (CDC) and/or **Storage** tasks. Wait for the tasks to complete. This will ensure that the storage tables contain the latest updates.
 - b. Run the **Compactor** task and wait for it to complete. This merges the changes (i.e. history) from the **delta_hds** folder with the **hds folder** and then deletes the **delta_hds** folder.
3. Make a note of the Compose for Data Lakes states storage folder path in the file system as you will need to specify it later.

To do this:

- a. Open the **Manage Data Storage Tasks** window by clicking **Data Storage Tasks** in the bottom right of the **Storage Zone** panel.
- b. In the left pane, select a task and then click the **Task Commands** toolbar button. The **Task Commands - landing <name> (<n> instructions)** dialog opens.
- c. Double-click the last command in the list. The dialog switches to item view and displays the last command.

The storage path is shown below the **Description** field as shown in the following example:



Task Commands - landing_employees (3 Instructions)

List View

Process Number: 3 **Process Step:** 950 **Process Type:** None

Entity Name: N.A

Parameters:

Runtime Clause:

Description: Update Storage Zone Change Processing Context

```
1 store values: '00000101T000000 00000101T000000' AS `cmpscurpartition`
2 in location: hdfs:///tmp/mike/spark_migration/states/storage/task=210
```

- d. Copy the path up to and including the word "storage".

Step 2: Upgrade your databases to the versions supported by Qlik Compose

For more information on supported platforms, see the [Compose Help](#).



If you are migrating your data:

- *To prevent data conflicts, it is strongly recommended to take the Compose for Data Lakes machine offline at this stage.*
- *You will need to create new databases for each of your projects (as new tables with the same names as the existing tables will be created during the data migration).*

Step 3: Deploy the projects on Qlik Compose and migrate the data

On the Qlik Compose machine, do the following:

a). Migrate the project definitions

1. Create a **Data Lake** project as described in the [Compose online help](#).



When migrating Hive projects, best practice is to create a project with the same (or at least very similar) options. For example, if the project you are migrating was configured to use an Operation Data Store, then you should also select this option when creating the new project.

2. Define a Storage Zone. For instructions, see the [Compose online help](#).



Makes sure to specify:

- *The same **Host** that was defined for the storage in the Compose for Data Lakes project.*
- *A different **Database** (as new tables with the same name will be created during the data migration).*

3. From the **Start** menu, open the **Compose Command Line** console and run the following command:.

```
ComposeCli.exe connect
```

4. Merge the Compose for Data Lakes project with the Qlik Compose project by running the following command:

```
ComposeCli.exe adjust_cfdl_project --project project_for_migration --infile deployment_zip_file_path
```

Where:

- `--project` is the name of the project you created in Qlik Compose.
- `--infile` is the path to the deployment ZIP file that you copied to the Qlik Compose machine.

Example:

```
ComposeCli.exe adjust_cfdl_project --project sales --infile sales_deployment_2021_06_02__19_24_21.zip
```

The adjusted deployment package will be created in the following location:

```
<COMPOSE_INSTALL_DIR>\data\projects\<project_name>\deployment_package
```

The file name will be in following format:

```
<project_name>__<timestamp>__QlikComposeDLMigration.zip
```

5. Deploy the package as follows:
 - a. In the main Compose window, select the project you created earlier. Then, click the **Deployment** drop-down menu in the toolbar and select **Deploy**. The **Deploy** window opens.
 - b. Click **Select** and browse to the location of the deployment package created in Step 4. In the **Open** window, either double-click the deployment package ZIP file or select the file and click **OK**. The package details will be displayed.
 - c. Click **Deploy** to deploy the package. When prompted to replace the existing project, confirm the operation. The project will be deployed.



Review all settings to make sure they are correct.

6. For each of your sources, open the **Edit Data Source** dialog, and make sure the **Associate with Replicate task** check box is selected. Then choose a Replicate task. Note that although this was optional in Compose for Data Lakes, it is required in the new Compose product. For an explanation of how to do this, see the [Compose online help](#).
7. Create the Storage Tables as described in the [Compose online help](#).

b). Migrate the data from a Hive project



Only applicable if you performed the data migration preparation procedure outlined in Step 1: Prepare project deployment and data migration in Compose for Data Lakes (page 7).

1. From the **Start** menu, open the **Compose Command Line** console and run the following command:

```
ComposeCli.exe connect
```

2. Then generate the migration script, as shown in the following example command:

```
ComposeCli.exe create_cfdl_data_migration_script --project hive --infile "C:\Migration\MyProject_deployment_2021_06_15__13_28_20.zip "
```

Where:

- `--project` is the name of the project you created in Qlik Compose.
- `--infile` is the path to your Compose for Data Lakes project deployment file before it was adjusted. See steps 1-3 of *Migration and upgrade (page 2)* above.

The script will be generated in the following location:

```
<COMPOSE_INSTALL_DIR>\data\projects\<<project-name>\ddl-scripts\C4DL_DATA_MIGRATION_SCRIPT_<timestamp>.sql
```

3. Run the **Schema Evolution** process and choose **Ignore Changes** as described in the [Compose online help](#). This basically "resets" Schema Evolution, thereby ensuring that old changes will not be applied to the storage should you want to run Schema Evolution at some point in the future.
4. Access the storage cluster using SQL Workbench or a similar tool and run the script.



If an error occurs when running the script, run the script again from the point of failure.

5. When the script completes successfully, generate and run your tasks in Compose.

c). Migrate the data from a Spark project



Only applicable if you performed the data migration preparation procedure outlined in Step 1: Prepare project deployment and data migration in Compose for Data Lakes (page 7).

1. From the **Start** menu, open the **Compose Command Line** console and run the following command:

```
ComposeCli.exe connect
```

2. Then generate the migration script, as shown in the following example command:

```
ComposeCli.exe create_cfdl_data_migration_script --project hive --states_path hdfs://cluster.qliktech.com:8020/compose/states/storage --infile "C:\Migration\MyProject_deployment_2021_06_15__13_28_20.zip"
```

Where:

- `--project` is the name of the project you created in Qlik Compose.
- `--infile` is the path to your Compose for Data Lakes project deployment file before it was adjusted. See steps 1-3 of *Migration and upgrade (page 2)* above.
- `--state_path` is the path to the Compose for Data Lakes states storage folder in the HDFS file system that you made a note of earlier.

The script will be generated in the following location:

```
<COMPOSE_INSTALL_DIR>\data\projects\<<project-name>\ddl-scripts\C4DL_DATA_MIGRATION_SCRIPT_<timestamp>.sql
```

3. Run the **Schema Evolution** process and choose **Ignore Changes** as described in the [Compose online help](#). This basically "resets" Schema Evolution, thereby ensuring that old changes will not be applied to the storage should you want to run Schema Evolution at some point in the future.
4. Access the storage cluster using SQL Workbench or a similar tool and run the migration script.



If an error occurs when running the script, run the script again from the point of failure.

5. When the script completes successfully, generate and run your tasks in Compose.

1.5 Compatibility with related Qlik products

Qlik Replicate is required for landing data into the data warehouse or storage while Qlik Enterprise Manager offers a convenient way of monitoring and controlling Compose tasks running on different servers. This section lists the supported versions for each of these products.

Compose August 2021 Initial Release

Compose August 2021 Initial Release is compatible with the following Replicate and Enterprise Manager product versions:

- **Qlik Replicate** - Qlik Compose is compatible with Replicate November 2020 latest service release, and Replicate May 2021 latest service release.
- **Enterprise Manager** - Qlik Compose is compatible with Enterprise Manager May 2021 latest service release.

Compose August 2021 Service Packs 01 and 02

Compose August 2021 Service Packs 01 and 02 are compatible with the following Replicate and Enterprise Manager product versions:

- **Qlik Replicate** - Qlik Compose is compatible with Replicate May 2021 latest service release and Replicate November 2021 (and its subsequent service releases).
- **Enterprise Manager** - Qlik Compose is compatible with Enterprise Manager November 2021 (and its subsequent service releases) only.

2 What's new?

The following section describes the enhancements and new features introduced in Qlik Compose August 2021.

2.1 What's new in Data Warehouse projects?

The following section describes the enhancements and new features introduced in Qlik Compose Data Warehouse projects.

Uniform source consolidation



Requires Compose August 2021 Service Release 03 or later.

To enable uniform source consolidation configuration, a new **Consolidation** tab has been added to the data warehouse task settings.

When the **Consolidate uniform sources** option is enabled, Compose will read from the selected data sources and write the data to one consolidated entity. This is especially useful if your source data is managed across several databases with the same structure, as instead of having to define multiple data warehouse tasks (one for each source), you only need to define a single task that consolidates the data from the selected data sources.

Consolidation tab showing selected data sources

Settings - ETL_SR_STAG_SAP_IT_DW_MASTER		
General	Advanced	Consolidation
<input checked="" type="checkbox"/> Consolidate uniform sources		
<input checked="" type="checkbox"/>	SR_STAG_SAP_IT	
<input checked="" type="checkbox"/>	SR_STAG_SAP_FR	
<input checked="" type="checkbox"/>	SR_STAG_SAP_DE	
<input type="checkbox"/>	SR_STAG_SAP_RU	

Environment variables



Requires Compose August 2021 Service Release 03 or later.

Environment variables allow developers to build more portable expressions, custom ETLs, and Compose configurations, which is especially useful when working with several environments, as is the case with DTAP (Development, Testing, Acceptance and Production). Different environments (for example, development and production) often have environment-specific settings such as database names, schema names, and Replicate task names. Variables allow you to easily move projects between different environments without needing to manually configure the settings for each environment. This is especially useful if many settings are different between environments. For each project, you can use the predefined environment variables or create your own environment variables.

Snowflake VARIANT data type support

When creating a Landing Zone, the new **Discover the VARIANT data type as** option lets you choose whether columns of the VARIANT data type will be created as JSON or XML in the Snowflake database.



VARIANT data type support in Replicate is planned for the November 2021 release. Until then, this feature can be used to discover tables with this data type that were created manually in the Landing Zone, or to manually set certain columns in the model to the JSON/XML data type.

Support for data profiling and data quality rules when using Google Cloud BigQuery



Requires Compose August 2021 Service Release 01 or later.

Added support for data profiling and data quality rules when using Google Cloud BigQuery as a data warehouse.

Attributes case sensitivity support



Requires Compose August 2021 Service Release 01 or later.

In previous versions, attempting to create several Attributes with the same name but a different case would result in a duplication error. Now, such attributes will be created with an integer suffix that increases incrementally for each attribute added with the same name. For example: Sales, SALES_01, and Sales_02.

Associating a Replicate task that writes to a Hadoop target



Requires Compose August 2021 Service Release 01 or later.

This version introduces support for associating a Replicate task that writes to a Hadoop target.

Performance improvements

- The "Discover" queries have been optimized to return results much faster than in previous versions
- When loading the data mart tables in Microsoft Azure Synapse Analytics, the 'FORCE ORDER' hint has been added to the statements that populate the temporary **DIM_MUT** tables and to the statements that populate the fact tables. For each dimension, the statistics are updated the moment before they are used to load the fact tables.
- The **New Star Schema** wizard has been scaled to allow a significantly larger number of tables to be selected.

Performance improvements that require Compose August 2021 Service Release 01 or later

- Validating a model with self-referencing entities is now significantly faster than in previous versions. For instance, it now takes less than a minute (instead of up to two hours) to validate a model with 5500 entities.
- The time it takes to "Adjust" the data warehouse has been significantly reduced. For instance, it now takes less than three minutes (instead of up to two hours) to adjust a data warehouse with 5500 entities.
- Significantly improved the loading speed of data mart Type 2 dimensions with more than two entities. In order to benefit from this improvement, customers upgrading with existing data marts needs to regenerate their data mart ETLs.

Data mart UX improvement



Requires Compose August 2021 Service Release 01 or later.

The Data Mart Dimensions tree and the Star Schema Fact tab were redesigned to provide a better user experience.

2.2 What's new in Data Lake projects?

The following section describes the enhancements and new features introduced in Qlik Compose Data Lake projects.

Support for excluding deleted records from ODS views



Requires Compose August 2021 Service Release 03 or later.

A **Deleted records in ODS views** section has been added to the **General** tab of the project settings, with the following options:

- **Exclude the corresponding record from the ODS views** - This is the default option as records marked as deleted should not usually be included in ODS views.

- **Include the corresponding record in the ODS views** - Although not common, in some cases, you might want include records marked as deleted in the ODS views in order to analyze the number of deleted records and investigate the reason for their deletion. Also, regulatory compliance might require you to be able to retrieve the past record status (which requires change history as well).



As this was the default behavior in previous versions, you might need to select this option to maintain backward compatibility.

Databricks on Google Cloud



Requires Replicate May 2021 Patch Release 02 or later.

Databricks on Google Cloud 8.x is now supported.

Generating projects using the CLI

Previously available only for Data Warehouse projects, this version introduces support for generating Data Lake projects using the Compose CLI. This can be especially useful when deploying projects between different environments. When the `generate_project` command is run, Compose:

- Validates the metadata.
- Creates any storage tables that do not exist.
- Validates the storage.
- Adjusts the storage if needed.
- Generates all storage tasks.

The "Associate with Replicate task option" is now mandatory

When defining a Landing Zone for Data Lake projects, associating a Replicate task (by selecting the **Associate with Replicate task** check box) is now mandatory. Therefore, after upgrading to this version, make sure to set this option if it is not already set.

2.3 All Projects

The following section describes the enhancements and new features that are common to both Data Lake projects and Data warehouse projects.

Task Recovery

The ability to set SQL state classes and error codes for triggering task recovery has been added to the project settings' **Environment** tab.

General	Naming	Environment
Environment Type:		Other <input type="text"/>
Environment Title:		<input type="text"/>
Creating or Dropping Storage Tables		
Limit the number of database connections to:	<input type="text" value="20"/>	<input type="text"/>
Task Recovery		
Maximum retry count:	<input type="text" value="5"/>	<input type="text"/>
Interval between retry attempts (sec):	<input type="text" value="2"/>	<input type="text"/>
Retry on these SQL state classes:	<input type="text" value="08"/>	
Retry also on these error codes:	<input type="text"/>	



Limitations and considerations:

- *Schema evolution retries are not supported.*
- *ODBC statements comprise a small part of the task execution sequence. However, as the task retry mechanism is JDBC-based, ODBC statements will not be retried even if the specified SQL state/error code is encountered.*

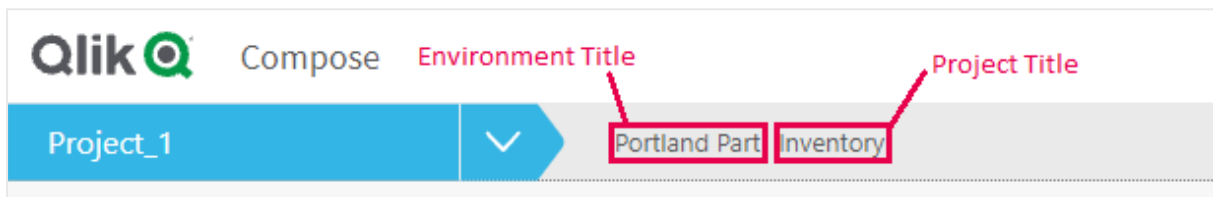
New Project title setting



Requires Compose August 2021 Patch Release 12 or later.

A new **Project title** setting had been added to the Environment tab of the project settings. The project title will be shown in the console banner. If both an **Environment Title** and a **Project Title** are defined, the project title will be displayed to the right of the environment title. Unlike the **Environment title** and **Environment type**, which are unique for each environment, the project title is environment independent. This means that the project title will always be retained, even when deploying to a different environment.

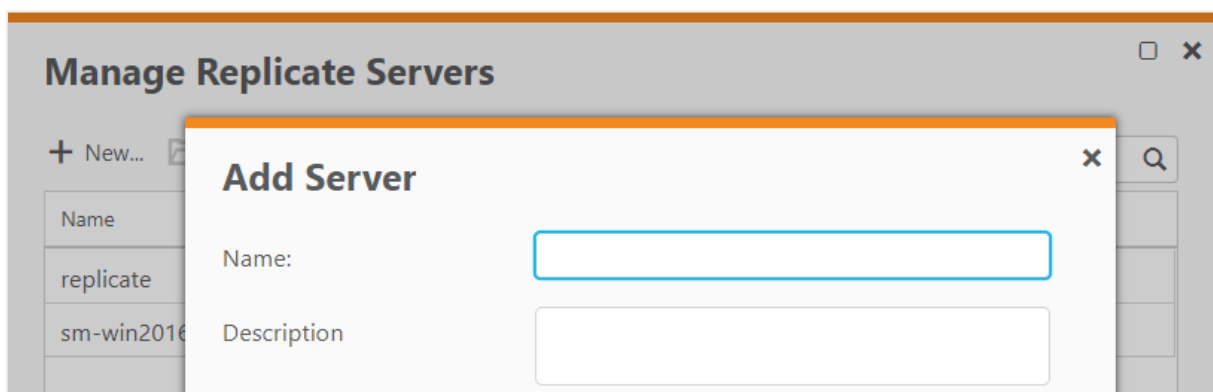
The following image shows the banner with both an **Environment title** and a **Project title**:



The banner text is shown without the **Environment title** and **Project title** console labels. This provides greater flexibility as it allows you to add any banner text you like, regardless of the actual label name. For example, specifying `Project owner: Mike Smith` in the **Project title** field, will display that text in the banner.

Support for multiple Replicate Servers

This version introduces support for configuring connectivity to Multiple Replicate Servers. This can be especially useful if you need to set up several projects in Compose, each with a different Replicate Server and task.



License

The parameter for registering a license has been changed from `--req` to `--infile`, thereby aligning it with other Qlik data integration products. In addition, the need to append the path with an "@" symbol is no longer applicable.

Example of previous command:

```
ComposeCLI.exe register_license --req @c:\Admin\Temp\lic.txt
```

Example of new command:

```
ComposeCLI.exe register_license --infile c:\Admin\Temp\lic.txt
```

Performance

Creating and dropping tables via Compose is now significantly faster when a large number of tables is involved.

2.4 Support for Microsoft Edge Browser



Requires Compose August 2021 Service Release 01 or later.

This version introduces support for accessing the Compose console using Microsoft Edge.

2.5 Compose Support in Enterprise Manager

From Enterprise Manager Service Release SR2, the following functionality will now be supported when working with Qlik Compose August 2021 (or above):

- Monitoring of Data Lake projects
- Monitoring of workflows - Note that monitoring information is provided at the workflow level only. For information on individual tasks in the workflow, you need to drill-down to the Compose project from Enterprise Manager.
- Monitoring tasks that have the same name, but are defined in different Compose projects. To facilitate such support:
 - In the API, Compose task names are now appended with a double underscore and the project name. For example, **MyTask__MyProject**.



Task names without an appended project name are no longer supported. Therefore, any scripts or applications using the old name format (i.e. without the `__projectName` suffix) will need to be updated to use the new format.

- In the console user interface, a new **Project** column has been added, allowing you to easily differentiate between two tasks with the same name.

The newly supported functionality has also been incorporated into the following API methods (REST, .NET, and Python): GetTaskList, GetTaskDetails, RunTask, and StopTask.

3 End of Life/Support and Deprecated Features

This section provides information about End of Life versions, End of Support features, and deprecated features.

- Internet Explorer will no longer be supported from Compose May 2022.

4 Resolved Issues


The following section lists the resolved issues in Compose August 2021 Initial Release, Compose August 2021 Service Release 01, and Compose August 2021 Service Release 02.

4.1 Resolved issues in Compose August 2021 Initial Release

The following section lists the resolved issues in Compose August 2021 Initial Release.

Component/Process	Description	Customer Case #
UI - Table Creation Modifiers	The Table Creation Modifiers feature was mistakenly available in Data Lake projects. It has now been removed.	N/A
UI - Lookup	A "Project Settings is null" error would be encountered when clicking the Show Lookup Data button in the Select Lookup Table window.	N/A
Amazon EMR	When using Amazon EMR with Glue, the following error would sometimes be encountered when performing Full Load of many tables (excerpt): ERROR processing query/statement. Error Code: 10006, SQL state: TStatus(statusCode:ERROR_STATUS,	2193872
Project deployment - Snowflake	After deploying a project with a Snowflake data warehouse, Compose would fail to run queries on Snowflake.	2201916
Generate project command	When the database_already_adjusted parameter was included in the generate_project command, Compose would drop and recreate the data mart tables.	2191511
Microsoft SQL Server	When using Windows authentication, the JDBC connection would fail with a "This driver is not configured for integrated authentication" error while running a Compose task.	2201049
Generate Project Documentation	When trying to generate project documentation, an "Object reference not set to an instance of an object." error would sometimes be encountered.	2189238

4 Resolved Issues

Component/Process	Description	Customer Case #
Microsoft SQL Server	<p>Using VARCHAR to calculate the checksum would sometimes result in an incorrect value. This was fixed by using VARCHAR(MAX) instead.</p> <p>After upgrading, you need to regenerate all the task ETLs (which can also be done using the generate_project CLI command). For details, see the Compose Help. It is also recommended to reload any data that contains strings. For an explanation of how to reload data, see Reloading data from the source tables to the Landing Zone in the Compose Help.</p>	2192529
ETL Statements - Satellite Tables	In rare scenarios related to backdating, the value in the "From Date" column would be different from the value in the "To Date" column of the preceding record.	2165287
ETL Statements - Error marts	In rare scenarios, when the mapping contained filters and more than one data validation rule was defined, the filters would not be applied when moving the data to the error marts, resulting in incorrect data.	N/A
Snowflake	<p>When the Optimize for initial load option was enabled and the data being loaded referenced missing records, the data warehouse task would sometimes fail with the following error: Cannot return unknown process parameter: Parameter StgRowsCnt_3327 not found.</p> <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"> <p> <i>If the data being loaded does contains references to missing records, to prevent task errors resulting from missing data, make sure to add the missing records and load them in Compose as soon as they become available.</i></p> </div>	2226814
Microsoft Azure Synapse Analytics - Data Quality	Filters used in Data Quality Rules would also be applied to the error mart (before the error mart rules). This would result in the error mart showing errors only for records that followed the filter <i>and</i> the Data Quality Rules.	2202255
Mail Server Settings	When the mail server settings' Use SSL option was enabled, Compose would fail to send an email if TLS protocol was required.	2229617

4.2 Resolved issues in Compose August 2021 SR1

The following section lists the resolved issues in Compose August 2021 Service Release 01.

Jira issue: RECOB-4045

Salesforce case: 10967

Type: Issue

Component/Process: Generate project CLI

Description: Running the generate_project CLI command with the --database_already_adjusted parameter would drop the Qlik table "TPIL_DMA_RUNNO".

Jira issue: RECOB-3999

Salesforce case: 9804

Type: Issue

Component/Process: Generate project CLI

Description: The generate_project CLI command would fail with the following error:

```
--database_already_adjusted
```

Jira issue: RECOB-4057

Salesforce case: N/A

Type: Issue

Component/Process: Data Mart

Description: Creating a denormalized new dimension would create the root dimension only.

Jira issue: RECOB-3990

Salesforce case: 2264064

Type: Issue

Component/Process: Workflows

Description: In rare cases, it would not be possible to create, edit, or duplicate workflows.

Jira issue: RECOB-3937, RECOB-3859

Salesforce case: 2236402, 5136

Type: Issue

Component/Process: Upgrade

Description: After migrating to 2021.5, projects containing two domain attributes with the same name but a different case (e.g. abc and Abc) would fail to load with the following error:

`SYS,GENERAL_EXCEPTION, An item with the same key has already been added.`

Jira issue: RECOB-3987

Salesforce case: N/A

Type: Issue

Component/Process: Project Deployment

Description: It would not be possible to open a project after deployment if one schema was missing.

Jira issue: RECOB-4043

Salesforce case: 9043

Type: Issue

Component/Process: Data Mart

Description: Fact tables would contain obsolete VIDs from dimensions, resulting in orphaned records.

Jira issue: RECOB-4033

Salesforce case: 9805

Type: Issue

Component/Process: Data Mart

Description: Data mart loading tasks would sometimes fail with the following error:

`cannot write value for process parameter twice: 1265: Duplicate write to param DimCnt_Tot`

Jira issue: RECOB-3204

Salesforce case: 2214622

Type: Issue

Component/Process: Loading data mart dimensions into Snowflake and Microsoft Azure Synapse Analytics

Description: When a data mart ETL task failed, the next task would sometimes load duplicate rows into dimensions.

Jira issue: RECOB-3957

Salesforce case: 2231873

Type: Issue

Component/Process: Data marts

Description: Adding data mart dimensions would sometimes fail without a clear error.

Jira issue: RECOB-3954

Salesforce case: 8634

Type: Issue

Component/Process: Data warehouse validation

Description: The following error would occur when validating the data warehouse:

Index was out of range. Must be non-negative and less than the size of the collection

Jira issue: RECOB-3902

Salesforce case: 7392

Type: Issue

Component/Process: Snowflake

Description: The data warehouse ETL would fail to create a transient table with a "already exists" error.

Jira issue: RECOB-3934

Salesforce case: 8399

Type: Issue

Component/Process: CLI

Description: Importing a project repository to a new project that does not exist it would fail with the following error:

Project: 'Project_name' does not exist.

Jira issue: RECOB-3636

Salesforce case: 2248515

Type: Issue

Component/Process: Backdating

Description: Backdated data in the Data Warehouse would not get updated in the Data Mart.

Jira issue: RECOB-3703

Salesforce case: 2240557

Type: Issue

Component/Process: Backdating

Description: Migrating a project from an older version would disable the backdating options. The issue was resolved by adding a new CLI command line that sets the "Add actual data row and a precursor row" option for all entities as well as in the project settings.

```
composecli set_backdating_options --project project_name
```

After running the command, refresh the browser to see the changes.

Jira issue: RECOB-3719

Salesforce case: 2260256

Type: Issue

Component/Process: Discovery from Snowflake

Description: When a landing table had a foreign key, discovering the table would result in the following error (excerpt):

Specified argument was out of the range of valid values.

Jira issue: RECOB-3799

Salesforce case: 2264057

Type: Issue

Component/Process: Validation and Schema Evolution

Description: Validation of Databricks storage and Snowflake data warehouse would be excessively long. The slow Databricks validation would also impact schema evolution.

4.3 Resolved issues in Compose August 2021 SR2

The following section lists the resolved issues in Compose August 2021 Service Release 02.

Jira issue: RECOB-4528

Salesforce case: 17678

Type: Issue

Component/Process: Pivot table - Google BigQuery

Description: In Google BigQuery projects, the data mart pivot table displays a "no data error" when there is data in tables.

Jira issue: RECOB-4529

Salesforce case: 17465

Type: Issue

Component/Process: Data profiler - Google BigQuery

Description: In Google BigQuery projects, the following error would be encountered when using the data profiler: "SYS,GENERAL_EXCEPTION,Sequence contains no elements"

Jira issue: RECOB-4535

Salesforce case: 16513

Type: Issue

Component/Process: OID and VID Columns

Description: The OID and VID column names would include the entire path from the fact source to the dimension instead of just the dimension name.

Jira issue: RECOB-4555

Salesforce case: 2260638

Type: Issue

Component/Process: MySQL source

Description: When setting up a MySQL source connection, testing the connection would return the following error: "Object reference not set to an instance of an object".

Jira issue: RECOB-4557

Salesforce case: 19777

Type: Issue

Component/Process: Export CLI

Description: After deleting an entity, export of projects using the CLI would sometimes fail.

Jira issue: RECOB-4584

Salesforce case: 19673

Type: Issue

Component/Process: Data mart loading

Description: When a dimension contained more than 10 entities, loading of the data mart would fail with the following error: "Case expressions may only be nested to level 10.Operation cancelled by user"

Jira issue: RECOB-4595

Salesforce case: 20256

Type: Issue

Component/Process: Data mart task generation

Description: Data mart task generation would fail when attributes of the same entity were assigned to different satellite tables.

Jira issue: RECOB-4633

Salesforce case: 20347

Type: Issue

Component/Process: Bulk Operations

Description: Generating Bulk Operations would not include the last data mart in the list.

Jira issue: RECOB-4636

Salesforce case: 20746

Type: Issue

Component/Process: Data mart loading

Description: Some projects could not be opened after upgrading.

Jira issue: RECOB-4464

Salesforce case: 14522

Type: Issue

Component/Process: CLI

Description: Running the "generate_project" command with the "database_already_adjusted" parameter would reset the data mart to the "Create Tables" state.

Jira issue: RECOB-3917

Salesforce case: 2256585

Type: Issue

Component/Process: Data mart dimensions

Description: Sometimes, rows in dimensions would incorrectly be marked as obsolete.

Jira issue: RECOB-4459

Salesforce case: 17328

Type: Issue

Component/Process: CLI - Export CSV

Description: Running the export_csv command would cause ETL Set generation to fail for lookups with the following error:

SYS,GENERAL_EXCEPTION,startIndex cannot be larger than length of string.<p>Parameter name: startIndex

Jira issue: RECOB-4481

Salesforce case: 17567

Type: Issue

Component/Process: Data marts

Description: Data Mart creation would sometimes fail with the following error "Sequence contains no matching element".

Jira issue: RECOB-4482

Salesforce case: 17567

Type: Issue

Component/Process: Data marts

Description: An error would sometimes be encountered when trying to delete a star schema.

Jira issue: RECOB-4390

Salesforce case: 12810

Type: Issue

Component/Process: ETLs

Description: The ETL for handling data mart dimensions would use the non-optimized approach for one of the statements.

Jira issue: RECOB-4386

Salesforce case: 14640

Type: Issue

Component/Process: Snowflake

Description: After four hours of inactivity, a "Snowflake Authentication token has expired" error would be shown.

Jira issue: RECOB-4500

Salesforce case: 5008

Type: Issue

Component/Process: ETLs

Description: Verification of unused and/or outdated column mapping expressions would lead to redundant errors.

Jira issue: RECOB-4501

Salesforce case: 17659

Type: Issue

Component/Process: Data Marts

Description: Validation of Type 2 dimensions would sometimes fail with an error that no Type 2 columns were detected (and that the dimension should be created as Type 1), even though Type 2 relationships existed in the dimension.

Jira issue: RECOB-4370

Salesforce case: N/A

Type: Issue

Component/Process: Security

Description: Fixes critical vulnerabilities (CVE-2021-45105, CVE-2021-45046, CVE-2021-44228) that may allow an attacker to perform remote code execution by exploiting the insecure JNDI lookups feature exposed by the logging library log4j. The fix replaces the vulnerable log4j library with version 2.16.

Jira issue: RECOB-4293

Salesforce case: 15341

Type: Issue

Component/Process: UI

Description: Editing a data mart entity after creating the data mart would result in all of the fields being reordered alphabetically.

Jira issue: RECOB-4199

Salesforce case: 12178

Type: Issue

Component/Process: Project settings - Snowflake only

Description: Enabling the **Write metadata to the TDWM tables in the data warehouse** option in the project settings would have no effect.

Jira issue: RECOB--4320

Salesforce case: 2160919

Type: Issue

Component/Process: Deployment packages

Description: The source schema connection would not be updated after deploying a deployment package.

Jira issue: RECOB-4258

Salesforce case: 13575

Type: Issue

Component/Process: Data mart

Description: Data mart creation would fail when there were more than 500 relationships.

Jira issue: RECOB-4330

Salesforce case: 13852

Type: Issue

Component/Process: Amazon Redshift

Description: An error would occur when trying to connect to Amazon Redshift using SSL.

Jira issue: RECOB-4351

Salesforce case: 16688

Type: Issue

Component/Process: Data Marts

Description: When there was a 3-tier relationship - for example, Entity_A→Entity_B→Entity_C - and the Fact table contained columns from Entity_A and Entity_C, changes in the relationship values in Entity_B (which should have updated columns from Entity_C in the Fact) would not be updated in the Fact table.

Jira issue: RECOB-4071

Salesforce case: 5258

Type: Issue

Component/Process: Live Views

Description: Reading from live views would take an excessively long time.

Jira issue: RECOB-4387

Salesforce case: 16511

Type: Issue

Component/Process: Microsoft Azure Synapse Analytics

Description: Columns with numeric(n,n) data types would not be retrieved from the Landing Zone.

Jira issue: RECOB-4339

Salesforce case: 5276

Type: Issue

Component/Process: Import

Description: The following error would sometimes be encountered when importing a data mart:

```
SYS,GENERAL_EXCEPTION,Sequence contains no matching element
```

Jira issue: RECOB-4388

Salesforce case: 14522

Type: Issue

Component/Process: Compose CLI Project Generation

Description: Generating the project would truncate the data mart tables when running the following command:

```
ComposeCli.exe generate_project --project <project name> --database_already_adjusted
```

After generating the project, you need to clear the cache by running the following command:

```
ComposeCli.exe clear_cache --project <project_name> --type storage
```

Jira issue: RECOB-4316

Salesforce case: N/A

Type: Issue

Component/Process: Data Mart Tasks

Description: When loading dimensions, a column would sometimes be used twice, causing the data mart task to fail.

Jira issue: RECOB-4235

Salesforce case: 13170

Type: Issue

Component/Process: Data Mart Tasks

Description: A runtime parameter ("MutCnt_8323" or similar) was incorrectly initialized, causing the data mart task to fail.

Jira issue: RECOB-4104

Salesforce case: 2160919

Type: Enhancement

Component/Process: Microsoft Azure Synapse Analytics Performance

Description: Performance was improved by adding indexes to Transactional and State Oriented fact tables.

Jira issue: RECOB-4105

Salesforce case: 2160919

Type: Enhancement

Component/Process: Microsoft Azure Synapse Analytics Performance

Description: Performance was improved by creating the TEMP table as a HEAP table instead of a HASH table.

Jira issue: RECOB-4106

Salesforce case: 2160919

Type: Enhancement

Component/Process: Microsoft Azure Synapse Analytics Performance

Description: Performance was improved by updating the statistics after each incremental load of the dimensions.

Jira issue: RECOB-4126

Salesforce case: 10967

Type: Enhancement

Component/Process: Microsoft Azure Synapse Analytics Performance

Description: Performance was improved for data mart ETL tasks by adding indexes (over columns used for join clauses) to intermediate tables.

Jira issue: RECOB-4109

Salesforce case: 10247

Type: Issue

Component/Process: Diagnostics

Description: Diagnostic packages would contain the server name of the customer environment, which would sometimes result in users being locked out when the package was deployed in our internal testing environment. Now, the diagnostic packages will be generated without the server name.

Jira issue: RECOB-4113

Salesforce case: 2222648

Type: Issue

Component/Process: Project Documentation

Description: The project documentation for Multi-Table ETLs and Post-Loading ETLs was generated without contents.

Jira issue: RECOB-4142

Salesforce case: 10996

Type: Enhancement

Component/Process: Compose CLI Timeouts

Description: A session expired error would sometimes occur during the CLI commands that took a long time to complete (e.g. `import_csv`). To resolve such timeouts, users can now add the "`-timeout seconds`" parameter to the command. Setting "`--timeout -1`" will run the command without it timing out.

Jira issue: RECOB-3928

Salesforce case: 7892

Type: Issue

Component/Process: Post-ETL Error Reporting

Description: Errors in Post-ETL stored procedures run on Microsoft Azure Synapse Analytics would not be reported.

Jira issue: RECOB-4149

Salesforce case: 2218407

Type: Issue

Component/Process: ETLs on Snowflake

Description: While working with Snowflake via the private link configuration, the engine task would sometimes stop unexpectedly.

5 Known Issues

The table below lists the known issues for this release.

Component/Process	Description	Ref #
Schema Evolution - New Columns	<p>When using Replicate to move source data to Compose, both the Full Load and Store Changes replication options must be enabled. This means that when Replicate captures a new column, it is added to the Replicate Change Table only. In other words, the column is stored without being added to the actual target table (which in terms of Compose is the table containing the Full Load data only i.e. the landing table).</p> <p>For example, let's assume the Employees source table contains the columns First Name and Last Name. Later, the column Middle Name is added to the source table as well. The Change Table will contain the new column while the Replicate Full Load target table (the Compose Landing table) will not.</p> <p>In older versions of Compose for Data Warehouses, mappings relied on the Full Load tables (the Compose Landing tables), meaning that users were not able to see any new columns (i.e. Middle Name in the above example) until they were created in the Full Load tables via a reload.</p> <p>From Compose May 2021, the Compose Discover and Mappings windows show changes to new columns that exist in both the Change Tables and the Replicate Full Load target tables. This allows Schema Evolution to suggest adding columns that exist in either of them.</p> <p>Although this is a much better implementation, it may create another issue. If a Full Load or Reload occurs in Compose before the Replicate reload, Compose will try to read from columns that have not yet been propagated to the Landing tables (assuming they exist in the Change Tables only). In this case, the Compose task will fail with an error indicating that the columns are missing.</p> <p>Should you encounter such a scenario, either execute a reload in Replicate or create an additional mapping without the new columns to allow Compose to perform a Full Load from the Landing tables.</p>	N/A

Component/Process	Description	Ref #
Data Mart UI	The Expand All and Collapse All buttons do not function properly in the Transaction Date tab of the Star Schema Wizard .	RECOB-3251
Data Mart UI	When searching for an entity in the New/Edit Star Schema window or in the New/Edit Dimension window, the list is not expanded automatically to show matching results. As a workaround, expand the list to manually see the search results.	RECOB-3364