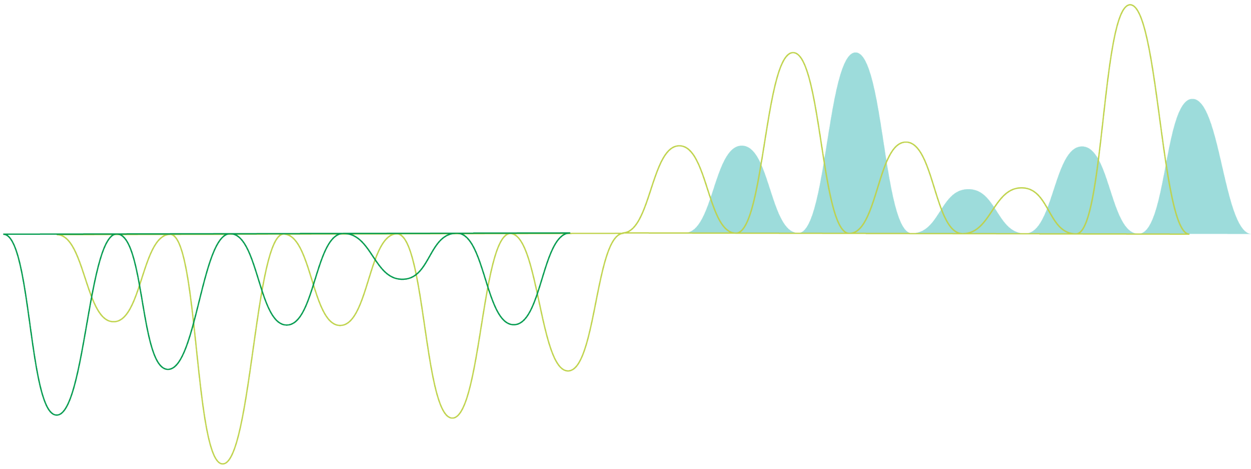
Qlik Compose for Data Lakes Setup and User Guide

Qlik Compose™ for Data Lakes

April 2020 (Version 6.6)

Last updated: December 07, 2020



HELP.QLIK.COM

1 Introduction

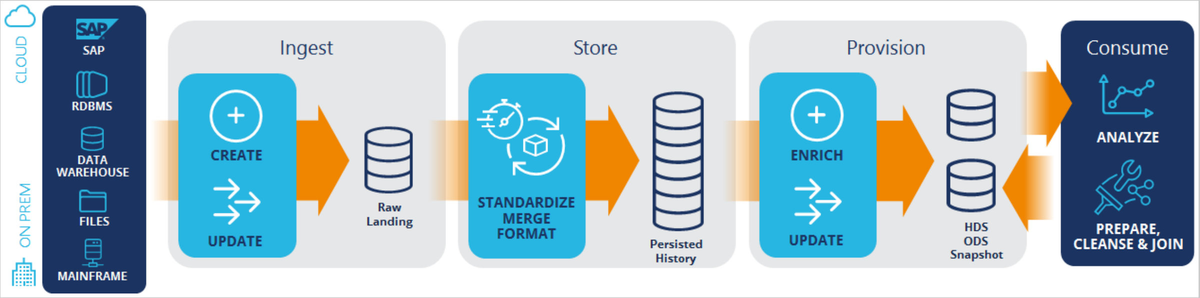
The need for better decision-making is driving businesses to incorporate Business Intelligence (or BI) into day-to-day practices. Business Intelligence is all about providing *relevant* information. For instance, if you know what your consumers are buying, you can develop products that match the current consumption trends and consequently boost profits. Likewise, BI enables you to identify key trends and patterns in your organization’s data and make the connections between important areas of your business that may otherwise seem unrelated.

However, setting up and maintaining a system that is capable of moving large volumes of data from a variety of sources further downstream for immediate and ongoing analysis is both complex and costly. Compose for Data Lakes overcomes the complexity with automation, using significantly fewer resources at lower cost.

Working in tandem with Qlik Replicate to facilitate analytics-driven business intelligence, Qlik Compose for Data Lakes's unique technology leverages Apache Spark to provide fast, flexible delivery of information from a wide variety of heterogeneous sources to Apache Hive, Amazon S3, or HDFS, residing on either ephemeral storage such as Amazon EMR or perpetual storage.

Qlik Compose for Data Lakes Architecture

The Qlik Compose for Data Lakes data flow is illustrated in the following diagram and described below:



1. **Ingest:** The source tables are loaded into the Landing Zone using Qlik Replicate, Apache Sqoop or other third-party replication tools.

   When using Qlik Replicate to move the source table to the Landing Zone, you can define either a **Full Load** replication task or a **Full Load and Store Changes** task to constantly propagate the source table changes to the Landing Zone.

2. **Store:** After the source tables are present in the Landing Zone, Compose for Data Lakes auto-generates metadata based on the data source(s). Once the metadata

and the mappings between the tables in the Landing Zone and the Storage Zone have been finalized, Compose for Data Lakes creates and populates the Storage Zone tables.

3. **Provision:** Subsets of stored data can be provisioned (as a snapshot or using incremental updates) to downstream Operational/Historical Data Stores located in Amazon S3, Apache Hive, Google Cloud, or HDFS.

It should be noted that even though setting up the initial project involves both manual and automatic operations, once the project is set up, you can automate the tasks by designing a Workflow in Compose for Data Lakes and/or utilizing the Compose for Data Lakes scheduler.

Limitations and Considerations

- Changes to record keys are not supported. If a source table record changes in a column that is mapped to a key in Qlik Compose for Data Lakes, and a value in that column changes in a specific record, Compose for Data Lakes will treat it as a new record.

2 Qlik Compose for Data Lakes Installation and Setup

This chapter describes how to install and set up Qlik Compose for Data Lakes.

Note that as Replicate serves as a data (and metadata) provider for Qlik Compose for Data Lakes, you also need to install *Qlik Replicate* in your organization. For a description of the Replicate installation procedure, refer to the *Replicate Setup and User Guide*.

In this chapter:

Preparing your System for Compose for Data Lakes

Qlik Compose for Data Lakes should be installed on a Windows Server machine that is able to access the Storage Zone and Landing Zone defined in your Compose for Data Lakes project. Note that, although not required, Qlik Compose for Data Lakes and Qlik Replicate can be installed on the same machine.

For information on the supported databases and versions, see Supported Platforms, Databases and Replicate Versions .

Before installing Compose for Data Lakes, make sure that the following prerequisites have been met:

- **Hardware configuration for the Compose for Data Lakes machine:**

Component	Basic System	Large System	Extra-Large System
Processor **Note** Additional cores may improve performance when several tasks are running concurrently.	Quad core	Quad core base	8-core base
Memory **Note** Additional memory may improve performance when several tasks are running concurrently.	8 GB	16 GB	32 GB
Disk requirements **Note** For all configurations, RAID is recommended for higher system availability in case of disk failure.	100 GB SSD	500 GB 10,000 RPM RAID	500 GB 15,000 RPM RAID
Network	1 Gb	10 Gb	Two 10 Gb

- **Hive Ports:** The following firewall ports should be open for inbound connections on the Hive machine:
  - HortonWorks: 10500
  - Amazon EMR: 10000
  - Cloudera: 10000

  These are the default ports; the actual port depends on the Hive Cluster configuration.
- **Compose Ports:** The following firewall ports should be open on the Compose for Data Lakes machine: 80/443

  - If the Qlik Compose for Data Lakes Agent is installed locally on the Compose for Data Lakes server machine, a random port is used each time.
    - If the Qlik Compose for Data Lakes Agent is installed remotely, port 3102 is used on the Compose Agent side.
    - If AEM is installed, port 443 on the Compose for Data Lakes server will be used for communication.
- Microsoft Visual Studio C++ 2015 X64 Redistributable installed on the machine. .NET Framework 4.5.2 or above installed on the machine.
- TLS v1.2 needs to be fully installed and configured prior to installing on a Windows 2016 Server.
- **Supported Browsers:** The following browsers can be used to access the Console (located on the machine):
  - Internet Explorer: 11 and above
  - Mozilla Firefox: Latest version
  - Google Chrome: Latest version

Firefox and Chrome automatically update themselves to the latest version.

Installing Compose for Data Lakes

The following section describes how to install Qlik Compose for Data Lakes.

To install Compose for Data Lakes:

1. Run the Compose for Data Lakes setup file (Compose_for_Data_Lakes_<version.build>.exe).

   The **Qlik Compose for Data Lakes** setup wizard opens.
2. Click **Next**. Select **I accept the terms of the license agreement** and then click **Next** again.
3. Optionally, change the installation directory and then click **Next**.
4. Click **Next** and then click **Next** again to start the installation.
5. When the installation completes, click **Finish** to exit the Wizard.

   Note As part of the installation, a new Windows Service called Attunity Compose for Data Lakes is created.

Post Installation

1. Perform the steps described in Installing the Hortonworks or Cloudera JDBC Driver for Apache Hive.
2. Open the Qlik Compose for Data Lakes console as described in Accessing Qlik Compose for Data Lakes.

> **Note** When you first open the Qlik Compose for Data Lakes Console, you will be prompted to register an appropriate license. Register the license that you received from Qlik.

Installing Compose for Data Lakes Silently

Compose for Data Lakes can be installed silently (i.e. without requiring user interaction). This option is useful, for example, if you need to install Compose for Data Lakes on several machines throughout your organization.

Before commencing the installation, make sure that the prerequisites have been met. See Preparing your System for Compose for Data Lakes.

The following topics describe:

- Silently Installing Compose for Data Lakes
- Silently Uninstalling Compose for Data Lakes
- Silently Upgrading Compose for Data Lakes

Silently Installing Compose for Data Lakes

The installation process consists of two stages:

1. Creating a Response File
2. Running the Silent Install

Creating a Response File

Before starting the installation, you need to create a response file.

To create the response file

1. From the directory containing the Compose for Data Lakes setup file, run the following command (note that this will also install Compose for Data Lakes):

   ```
   Compose_for_Data_Lakes_<version_number>.exe /r /f1<my_response_file>
   ```

   where:

`<my_response_file>` is the full path to the generated response file.

Example:

```
Compose_for_Data_Lakes_<version_number>.exe /r /f1C:\Compose_
install.iss
```

2. To change the default installation directory, open the response file in a text editor and edit the *first* **szDir** value as necessary.
3. To change the default data directory, edit the *third* **szDir** value as necessary.
4. Save the file as **<name>.iss**, e.g. **Compose_install_64.iss**.

Running the Silent Install

To silently install Compose for Data Lakes, open a command prompt and change the working directory to the directory containing the Compose for Data Lakes setup file. Then issue the following command (where <response file> is the path to the response file you created earlier):

Syntax:

```
<Compose_setup_file> /s /f1<my_response_file> [/f2<LOG_FILE>]
```

Example:

```
C:\>Compose_for_Data_Lakes_<version_number>.exe /s /f1C:\temp\1\Compose_
install.iss /f2C:\temp\1\silent_x64_install.log
```

If the installation was successful, the log file should contain the following rows:

```
[ResponseResult]

ResultCode=0
```

Silently Upgrading Compose for Data Lakes

Note Before starting the upgrade:

1. Create a response file. See Step 1 of "Creating a Response File" in Silently Installing Compose for Data Lakes
2. It is strongly recommended to back up the Compose for Data Lakes "Data" folder.

To silently upgrade Compose for Data Lakes:

1. Open a command prompt and change the working directory to the directory containing the Compose for Data Lakes setup file.

2. Issue the following command (where `<my_response_file>` is the path to the response file you created earlier):

Syntax:

```
<COMPOSE_KIT> /s /f1<my_response_file> [/f2<LOG_FILE>]
```

Example:

```
C:\>Compose_for_Data_Lakes_<version_number>.exe /s /f1C:\temp\1\Compose_
upgrade.iss /f2C:\temp\1\silent_x64_up.log
```

If the upgrade was successful, the log file should contain the following rows:

```
[ResponseResult]

ResultCode=0
```

Silently Uninstalling Compose for Data Lakes

Silently uninstalling Compose for Data Lakes also comprises:

1. Creating a Response File
2. Running the Silent Uninstall

The process is the same as for silently installing Compose for Data Lakes. For instructions, see Silently Installing Compose for Data Lakes

Installing the Compose Agent

When defining a Qlik Compose for Data Lakes for Spark project, the Qlik Compose for Data Lakes Agent must be installed on the remote Spark machine, which may either be ephemeral (i.e. part of an Amazon EMR, Microsoft Azure HDInsight, or Google Dataproc cluster) or non-ephemeral.

The installation procedure differs according to whether your Hadoop cluster is ephemeral or non-ephemeral.

Hadoop Cluster Type	Topic
Non-Ephemeral Cluster	See Installing Compose Agent in a Non-Ephemeral Environment

Ephemeral Cluster	See one of the following topics: » Launching an Amazon EMR Cluster with Compose Agent » Setting up a Microsoft Azure HDInsight Cluster with Qlik Compose Agent » Launching a Google Dataproc Cluster with Compose Agent

Which Installation Package Do I Need?

The Compose Agent package you need to install depends on the Hadoop target platform.

The available platforms are as follows:

Platform	Required Package
Hortonworks	compose-agent-<version>-<build>.x86_64.rpm
Amazon EMR	compose-agent-<version>-<build>.x86_64.rpm
Cloudera	compose-agent-<version>-<build>.x86_64.rpm
Microsoft Azure HDInsight	compose-agent-<version>-<build>.amd64.deb
Google Cloud Storage (Dataproc)	compose-agent-<version>-<build>.amd64.deb

To obtain the package for your target platform, download the following file from the Qlik Customer Zone:

AttunityComposeForDataLakes_Agent_<Version>_Linux_X64.zip

After installing the Qlik Compose for Data Lakes Agent, you need to provide the connection settings to the Spark machine or to your ephemeral cluster.

For more information on providing the connection settings, see Compose Agent Settings.

Installing Compose Agent in a Non-Ephemeral Environment

This topic explains how to install the Compose Agent in a non-ephemeral Hadoop cluster environment, which may exist either on-premises or in the cloud. The package you need to install depends on your environment. For more information, see Which Installation Package Do I Need?

Requires Java runtime 1.8 and above.

Installing or Upgrading the RPM Package

To install the Compose Agent:

Run the following command:

```
[user=username] [group=groupname] [verbose=true] [debug=true]
password=your-compose-agent-password platform=my-platform -ivh compose-
agent-<version>-<build>.x86_64.rpm
```

The `my-platform` parameter can have one of the following values: `hortonworks`, `emr`, `cloudera`, `dataproc`, `hdinsight`.

To upgrade the Compose Agent:

Run the following command:

```
rpm -Uvh compose-agent-<version>-<build>.x86_64.rpm
```

- Before upgrading the Compose Agent, you should stop all Compose Agent tasks and services, and start them again only after the Compose Agent upgrade has completed successfully.
- If Compose for Data Lakes is installed on the same machine as Compose for Data Warehouses, before upgrading Compose for Data Lakes, you must stop the Qlik Compose for Data Warehouses service. After upgrading Compose for Data Lakes, you can restart the Compose for Data Warehouses service.
  Alternatively, you can uninstall the current version of Compose for Data Lakes and then install the new version using the same folder.

To start the Compose Agent:

After installation or upgrade, start the Compose Agent by running the command:

```
./compose-agent.sh start
```

After a few seconds, verify that the Agent was installed or upgraded successfully by running the command:

```
./compose-agent.sh status
```

The following message should be displayed:

```
Qlik Compose Engine is running
```

To uninstall the Qlik Compose for Data Lakes Agent:

Run the following command:

```
rpm -e compose-agent
```

Installing or Upgrading the Debian Package

To install the Compose Agent:

Run the following command:

```
[user=usename] [group=groupname] [verbose=true] [debug=true]
password=your-compose-agent-password dpkg -i compose-agent-<version>-
<build>.amd64.deb
```

To upgrade the Compose Agent:

```
dpkg -i compose-agent-<version>-<build>.amd64.deb
```

To uninstall the Compose Agent:

Run the following command:

```
dpkg -r compose-agent
```

Optional Parameters

Parameter	Description
[user=*usename*]	Overrides the default user under which the Java service runs. The default user name is "Compose".
[group=*groupname*]	Overrides the default group under which the Java service runs. The default group name is "Compose". Only the root user and the specified user can run the service. Other users in the group cannot run the service.
[verbose=true]	Sets the logging mode to verbose.
[debug=true]	Sets the logging mode to debug.
--prefix=/installation_dir Not supported with Debian.	Prefixes the installation directory with the specified path. For example, if you specified: `--prefix=/mydir1/mydir2` The Compose Agent would be installed here: **/mydir1/mydir2/attunity/acompose**

Installing the Hortonworks JDBC Driver for Apache Hive

Perform the steps described in Installing the Hortonworks or Cloudera JDBC Driver for Apache Hive.

Configuration Options

- The **site_compose-agent_login.sh** file under the **bin** directory is a site specific process environment configuration file that you can modify as required. This may be useful, for example, if you want Compose for Data Lakes to run with a specific Java version (for instance, when several Java versions are installed).
- **Spark home:** When using an on-premises Hadoop cluster, you need to specify the location of the Spark_Home variable (or $SPARK_HOME on Linux). This is not required when using Amazon EMR.

Changing the Compose Agent Password

If your cluster is active for an extended period, best practice is to periodically change the Compose Agent password.

To do this:

Run the following command from <INSTALL_DIR>\bin:

```
acjs.sh server setadminpassword new_password old_password
```

Example:

```
acjs.sh server setadminpassword 745hghTUYIIOJNOGO34 RE9R0EJVJFMA0GIW068
```

Launching an Amazon EMR Cluster with Compose Agent

The procedure below explains how to launch an Amazon EMR cluster with Compose Agent.

1. Create an Amazon S3 bucket that your Amazon EMR cluster has read access to.
2. Edit the **compose-agent-<version>-<build>-emr-installer.sh** file and replace the default password (`emr`) with your own password. This is the password that you need to specify in the Compose Agent settings.
3. Upload the following files to this bucket:
   - compose-agent-<version>-<build>.x86_64.rpm (Provided by Qlik)
   - compose-agent-<version>-<build>-emr-installer.sh (Provided by Qlik)
   - HiveJDBC41.jar

     To obtain this file, download the Amazon Hive JDBC Driver from the Amazon website.
4. Launch your EMR cluster with the following minimum requirements:
   - **EMR version:**
     - emr-5.15.0

- **The following services:**
  - Hadoop
  - Spark
  - Hive
  - Tez

5. Add a step of type “Custom JAR” to your EMR definition.
   1. In the **JAR location** field, specify the Amazon **script-runner.jar** for your region (located in s3://region.elasticmapreduce/libs/script-runner/**script-runner.jar**).

      For more information, see: https://docs.aws.amazon.com/emr/latest/ReleaseGuide/emr-hadoop-script.html
   2. In the **Arguments** field:
      1. Pass the bucket location (i.e. including the file name) of the **compose-agent-<version>-<build>-emr-installer.sh** script as an argument to the script-runner JAR.
      2. Pass the bucket folder in which the **compose-agent-<version>-<build>.x86_64.rpm** file resides as an argument to the **compose-agent-<version>-<build>-emr-installer.sh** script. If there are multiple **compose-agent-<version>-<build>.x86_64.rpm** files in the specified location, the script will always take the latest file.
      3. Pass the bucket location (i.e. including the file name) of the **HiveJDBC41.jar** file as an argument to the **compose-agent-<version>-<build>-emr-installer.sh** script.

      Make sure to separate the arguments with a space.

      Example:

      s3://mybucket/compose compose-agent-<version>-<build>-emr-installer.sh s3://mybucket/compose s3://mybucket/compose/HiveJDBC41.jar

      When you terminate a cluster the associated HDFS storage will also be terminated. Therefore, if you want stored and provisioned data to be retained when a cluster is terminated, set the data store type (i.e storage) and the provisioning target in Compose for Data Lakes to Amazon S3.

Configuring Compose for Data Lakes to work with the Compose Agent on Amazon EMR

Server name:

When you configure Compose for Data Lakes to work with the remote Compose Agent, you need to select **Remote server** and enter the remote **Server name**. This can be done using any of the following methods:

- Map the cluster IP in the Windows **hosts** file and update the IP address each time a new cluster is launched:

  Example: 12.3.45.678 amazon.emr.cluster

  With this method, although you need to update the cluster IP address in the **hosts** file whenever a new cluster is launched, the host name (**amazon.emr.cluster** in the above example) specified in the **Compose Agent Settings** window never needs to be changed.
- In the **Compose Agent Settings** window, update the **Server name** field with the new IP address whenever a new cluster is launched.

Password:

The password is the password that you replaced in the **compose-agent-<version>-<build>-emr-installer.sh** script as described above.

Setting up a Microsoft Azure HDInsight Cluster with Qlik Compose Agent

You can either launch a Microsoft Azure HDInsight cluster with the Compose Agent or install the Compose Agent on an active Microsoft Azure HDInsight cluster. This section explains how to do both as well as how to configure Qlik Compose for Data Lakes to work with the Compose Agent on a Microsoft Azure HDInsight cluster.

Launching a Microsoft Azure HDInsight Cluster with Qlik Compose Agent

The procedure below explains how to launch a Microsoft Azure HDInsight cluster with Qlik Compose Agent.

1. Create an Microsoft Azure Blob Storage container to which your HDInsight cluster will have read access.
2. Edit the **compose-agent-<version>-<build>-hdi-installer.sh** file and replace the default password (`azure`) with your own password. This is the password that you need to specify in the Compose Agent settings.

3. Upload the following files to the container:
   - compose-agent-<version>-<build>.amd64.deb (provided by Qlik)
   - compose-agent-<version>-<build>-hdi-installer.sh (provided by Qlik)
   - HiveJDBC41.jar (Simba Hive JDBC Driver)

     To obtain this file, download the Hortonworks JDBC Driver for Apache Hive (v2.6.2.1) from the Hortonworks website.
4. Launch your Microsoft Azure HDInsight cluster with the following minimum requirements:
   - **Cluster type:** Spark
   - **Version:** Spark 2.1.0 and above
5. In Step 5 of the cluster launch - **Advanced Settings** - click **Script Actions** and then click **Submit New**.
6. In the **Submit script action** window, enter the following values:
   a. **Script type:** - Custom
   b. **Name:** Any
   c. **Bash script UI:** Select the **compose-agent-<version>-<build>-hdi-installer.sh** script in the container you created earlier and copy its URL to this field.
   d. **Node type(s):** Select **Head**.
   e. **Parameters:** Select the **HiveJDBC41.jar** and **compose-agent-<version>-<build>.amd64.deb** files in the container you created earlier and copy their URLs to this field.
   f. Leave the **Persist this script action rerun when new nodes are added to the cluster** check box selected.
   g. Click **Create**.

Installing Qlik Compose Agent on an Active Microsoft Azure HDInsight Cluster

1. Edit the **compose-agent-<version>-<build>-hdi-installer.sh** file and replace the default password (azure) with your own password. This is the password that you need to specify in the Compose Agent settings in the Compose for Data Lakes console.
2. Copy the following files to your cluster head node:
   - compose-agent-<version>-<build>.amd64.deb (provided by Qlik)
   - compose-agent-<version>-<build>-hdi-installer.sh (provided by Qlik)

- HiveJDBC41.jar (Simba Hive JDBC Driver)

  To obtain this file, download the Hortonworks JDBC Driver for Apache Hive (v2.6.2.1) from the Hortonworks website.

3. Open a shell on the cluster head node and run the script using the root user (either through a root shell or using sudo):

   ```
   ./compose-agent-<version>-<build>-hdi-installer.sh compose-agent-
   <version>-<build>.amd64.deb HiveJDBC41.jar local
   ```

Configuring Compose for Data Lakes to work with the Compose Agent on a Microsoft Azure HDInsight Cluster

Server name:

When you configure Compose for Data Lakes to work with the remote Compose Agent, you need to select **Remote server** and enter the remote **Server name**. This can be done using any of the following methods:

- Map the cluster IP in the Windows **hosts** file and update the IP address each time a new cluster is launched:

  Example: 12.3.45.678 microsoft.hdinsight.cluster

  With this method, although you need to update the cluster IP address in the **hosts** file whenever a new cluster is launched, the host name (**microsoft.hdinsight.cluster** in the above example) specified in the **Compose Agent Settings** window never needs to be changed.

- In the **Compose Agent Settings** window, update the **Server name** field with the new IP address whenever a new cluster is launched.

Password:

The password is the password that you replaced in the **compose-agent-<version>-<build>-hdi-installer.sh** script as described above.

Launching a Google Dataproc Cluster with Compose Agent

For supported Google Dataproc versions, see Supported Hive Distributions.

The procedure below explains how to launch a Google Dataproc Cluster with Qlik Compose Agent.

1. Create a Google Cloud Storage bucket that your Google Dataproc cluster has read access to.
2. Edit the **compose-agent-<version>-<build>-dataproc-installer.sh** file and replace the default password (`google`) with your own password. This is the password that you need to specify in the Compose Agent settings.
3. Upload the following files to the bucket you created earlier:
   - compose-agent-<version>-<build>.amd64.deb (Provided by Qlik)
   - compose-agent-<version>-<build>-dataproc-installer.sh (Provided by Qlik)
   - HiveJDBC41.jar (Simba Hive JDBC Driver)

     To obtain this file, download the Hortonworks JDBC Driver for Apache Hive (v2.6.2.1) from the Hortonworks website.
4. From the **Navigation menu** in the Google Cloud Platform console, select **Compute Engine** > **Metadata**.
5. In the **Metadata** window:
   a. Add the following metadata items:
      - att-cmps-package-folder-url
      - att-cmps-hive-jdbc-jar-url

   b. Click **Save**.
6. Return to the **Navigation menu** and select **Dataproc** > **Clusters**.
7. Configure your cluster settings as desired and then configure the following settings which are required for Compose Agent:
   a. Expand the **Advanced options** and click the **Add initialization action** button.

   a. In the **bucket/folder/file** field, browse to the **compose-agent-<version>-<build>-dataproc-installer.sh** file in the bucket you created earlier.
8. Create your Google Dataproc cluster.

Configuring Compose for Data Lakes to work with the Compose Agent on a Google Dataproc Cluster

Server name:

When you configure Compose for Data Lakes to work with the remote Compose Agent, you need to select **Remote server** and enter the remote **Server name**. This can be done using any of the following methods:

- Map the cluster IP in the Windows **hosts** file and update the IP address each time a new cluster is launched:

  Example: 12.3.45.678 google.dataproc.cluster

  With this method, although you need to update the cluster IP address in the **hosts** file whenever a new cluster is launched, the host name (**google.dataproc.cluster** in the above example) specified in the **Compose Agent Settings** window never needs to be changed.
- In the **Compose Agent Settings** window, update the **Server name** field with the new IP address whenever a new cluster is launched.

Password:

The password is the password that you replaced in the **compose-agent-<version>-<build>-dataproc-installer.sh** script as described above.

Verifying that Compose Agent is Correctly Installed

There are several ways of verifying that Compose Agent is installed correctly.

These are as follows:

- Check the color of the connectivity icon in the upper right corner of the Compose for Data Lakes Console:
  - Green indicates that a successful connection to Compose Agent has been established.
  - Red indicates that there was a problem connecting to Compose Agent.
- Select **Compose Agent Settings** from the **Management** menu in the main window and click **Test Connection**.
- Check for [Error] messages in the **compose_agent.log** which can be accessed as described in Viewing and Downloading Compose for Data Lakes Log FilesViewing and Downloading Compose for Data Lakes Log Files

Installing the Hortonworks or Cloudera JDBC Driver for Apache Hive

1. Download the latest Hortonworks JDBC Driver for Apache Hive from the Simba website:

   `https://www.simba.com/product/apache-hive-driver-with-sql-connector/`

   Then, extract the **HiveJDBC41.jar** file from the **Simba_HiveJDBC41_<version>.zip** file.

   -Or-

   Download the Hive JDBC Driver from the Cloudera website:

   `https://www.cloudera.com/downloads/`

   Then, extract the **HiveJDBC41.jar** file from the zip file that contains the **Hive JDBC Connector.**

   > **Note** You need to register on the Simba and Cloudera websites before you can download the Hortonworks or Hive JDBC Driver.

2. Copy the **HiveJDBC41.jar** file to the following location(s), depending on where the Compose Agent is installed:

   - If the Compose for Data Lakes Agent is installed locally (i.e. as part of the Compose for Data Lakes installation), copy the HiveJDBC41.jar file to the following location on the Windows Compose for Data Lakes machine:

     <Compose_Installation_Dir>\java\jdbc

   - If the Compose Agent is installed remotely (required for Apache Spark projects), copy the HiveJDBC41.jar file to the following location on the Linux Compose Agent machine:

     <Compose_Installation_Dir>/jdbc

3. If the Compose Agent is installed locally (i.e. on Windows), restart the Attunity Compose for Data Lakes service.

4. If the Compose Agent is installed on Linux, restart the Compose Agent Server by running the following command:

   ```
   ./compose-agent.sh restart
   ```

Setting the Hostname and Changing the HTTPS Port

After installing Qlik Compose for Data Lakes, you can use the Compose for Data Lakes CLI to set the hostname and HTTPS port for accessing the Qlik Compose for Data Lakes server machine.

Under normal circumstances, you should not need to set the hostname. However, on some systems, connecting using HTTPS redirects to localhost. If this occurs, set the hostname of the Compose machine by running the command shown below.

To set the hostname

Run the following command:

```
<product_dir>\bin\ComposeCtl.exe configuration set --address address
```

where *address* is the hostname of the Compose for Data Lakes server machine.

To change the HTTPS port

Run the following command:

```
<product_dir>\bin\ComposeCtl.exe configuration set --https_port port_number
```

where *port_number* is the HTTPS port number of the Compose for Data Lakes server machine. The default HTTPS port is 443.

Setting Up HTTPS for the Compose for Data Lakes Console

Industry-standard security practices dictate that web user interface for enterprise products must use secure HTTP (HTTPS). Compose for Data Lakes enforces the use of HTTPS and will not work if HTTPS is configured incorrectly.

As Compose for Data Lakes uses the built-in HTTPS support in Windows, it relies on the proper setup of the Windows machine it runs on to offer HTTPS access. In most organizations, the IT security group is responsible for generating and installing the SSL server certificates required to offer HTTPS. It is strongly recommended that the machine on which Compose for Data Lakes is installed already has a valid SSL server certificate installed and bound to the default HTTPS port (443).

Checking if an SSL Certificate is Installed

To check whether an SSL certificate is installed, you can use the following command:

```
netsh http show sslcert | findstr /c:":443 "
```

If an SSL certificate is installed, the output should look like this:

```
netsh http show sslcert | findstr /c:":443 "

   IP:port : 192.168.1.13:443

   IP:port : 192.168.1.11:443

   IP:port : [fe80::285d:599c:4a55:1092%11]:443

   IP:port : [fe80::3d0e:fb1c:f6c3:bc52%23]:443
```

With a valid SSL certificate installed, the Qlik Compose for Data Lakes web user interface will automatically be available for secure access from a web browser using the following URL:

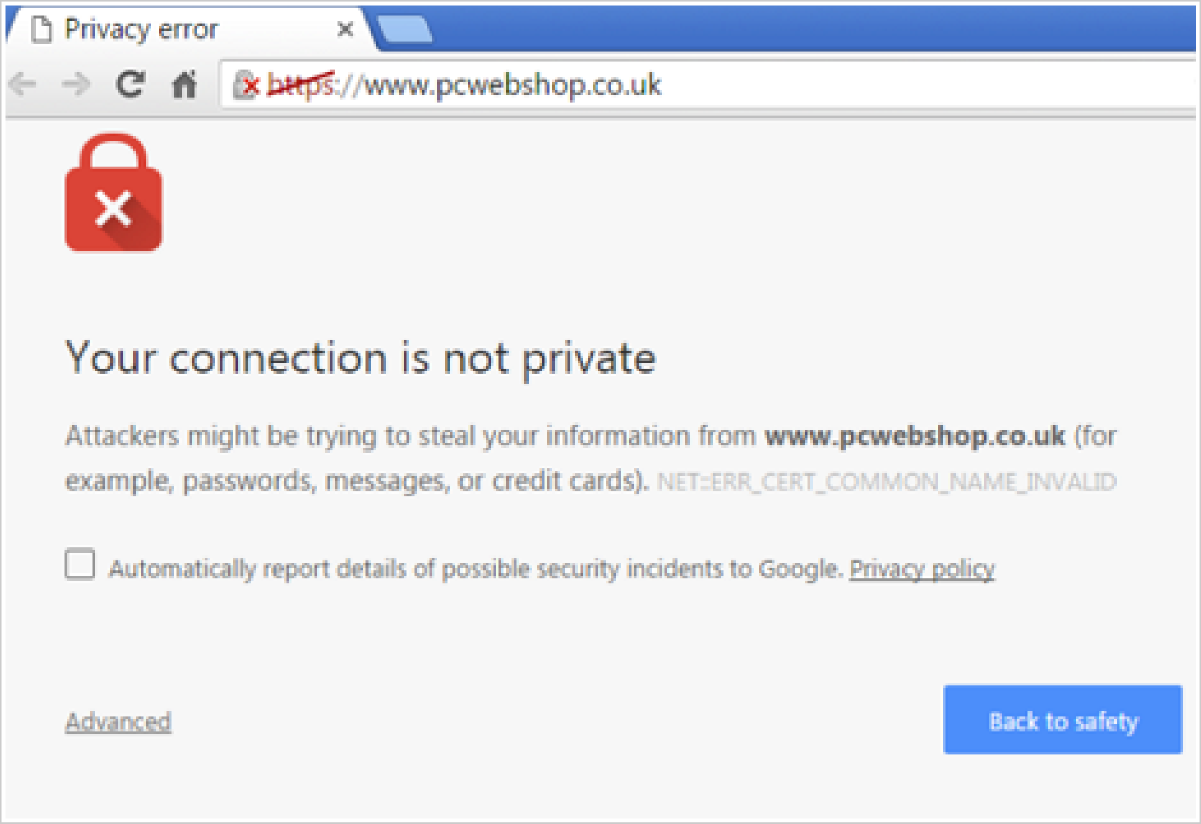
```
https://<machine-name>/attunitycompose_datalakes
```

Using the Self-Signed Certificate

Due to the way the HTTPS protocol works, there is no way for Compose for Data Lakes to automatically provide and install a valid SSL server certificate. Still, in the event that no SSL server certificate is installed, Compose for Data Lakes automatically generates and installs a self-signed SSL server certificate (as a temporary measure). This certificate is generated on the Compose for Data Lakes machine and cannot be exported or used elsewhere.

It should be noted that browsers do not consider the certificate to be valid because it was not signed by a trusted certificate authority (CA).

When connecting with a browser to a server that uses a self-signed certificate, a warning page is shown such as this one in Chrome:

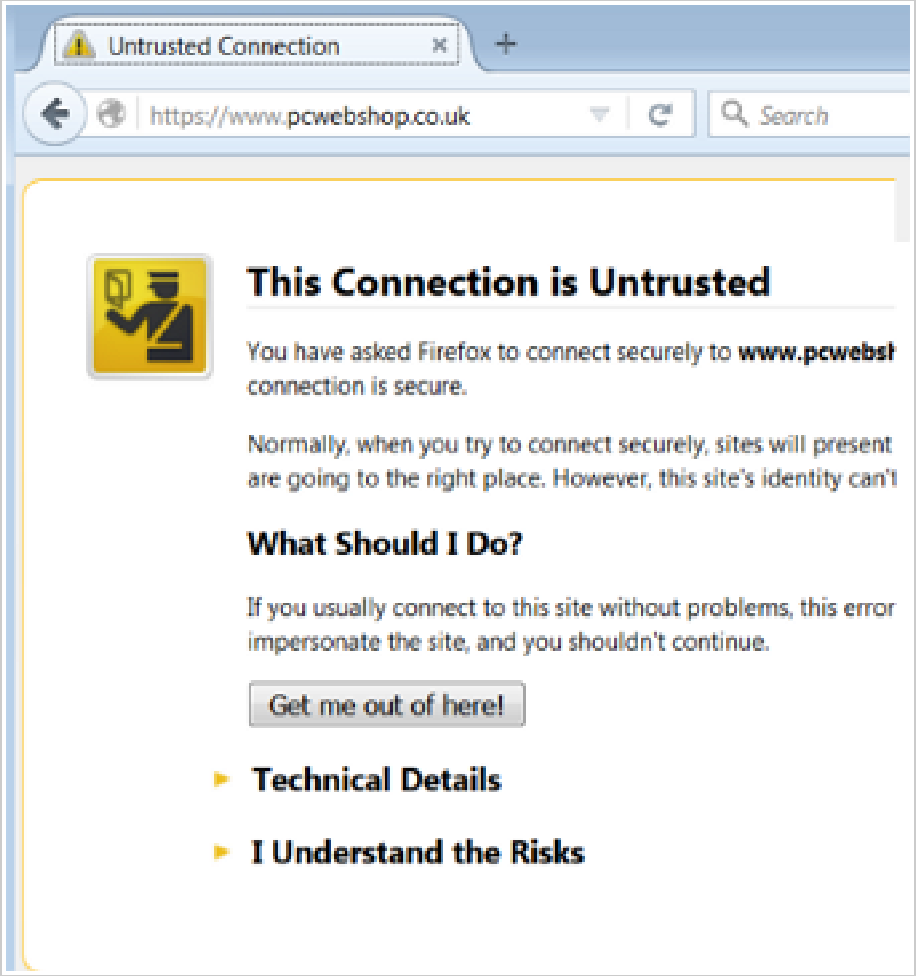


Your connection is not private

Attackers might be trying to steal your information from **www.pcwebshop.co.uk** (for example, passwords, messages, or credit cards). NET::ERR_CERT_COMMON_NAME_INVALID

□ Automatically report details of possible security incidents to Google. Privacy policy

Or this one in Firefox:



This Connection is Untrusted

You have asked Firefox to connect securely to **www.pcwebsl** connection is secure.

Normally, when you try to connect securely, sites will present are going to the right place. However, this site's identity can't

What Should I Do?

If you usually connect to this site without problems, this error impersonate the site, and you shouldn't continue.

The warning page informs you that the certificate was signed by an unknown certificate authority. All browsers display a similar page when presented with a self-signed certificate. If you know that the self-signed certificate is from a trusted organization, then you can instruct the browser to trust the certificate and allow the connection. Instructions on how to trust the certificate vary between browsers and even between different versions of the same browser. If necessary, refer to the help for your specific browser.

Some corporate security policies prohibit the use of self-signed certificates. In such cases, it is incumbent upon the IT Security department to provide and install the appropriate SSL server certificate (as is the practice with other Windows products such as IIS and SharePoint). If a self-signed certificate was installed and needs to be removed, then the following command can be used:

```
composeCtl.exe certificate clean
```

Note that after the self-signed certificate is deleted, connections to the Qlik Compose for Data Lakes machine will not be possible until a valid server certificate is installed.

Should you want to generate a new self-signed certificate (to replace the deleted certificate), simply restart the Attunity Compose for Data Lakes service.

Replacing the Self-Signed Certificate on Windows

The instructions below are intended for organizations who wish to replace the self-signed certificate generated by the Compose for Data Lakes Server on Windows with their own certificate. The process, which is described below, involves removing the self-signed certificate and then importing the new certificate.

See also Setting Up HTTPS for the Compose for Data Lakes Console.

Before starting, make sure that the following prerequisites have been met:

- The replacement certificate must be a correctly configured SSL PFX file containing both the private key and the certificate.
- The common name field in the certificate must match the name browsers will use to access the machine.

To remove the self-signed certificate created by Qlik Compose for Data Lakes:

1. Stop the Attunity Compose for Data Lakes service.
2. Open a command prompt (using the "Run as administrator" option) and change the path to the Compose for Data Lakes **bin** directory. The default path is C:\Program Files\Attunity\Compose for Data Lakes\bin.
3. Run the following command:

   ```
   composeCtl.exe certificate clean
   ```

To import your own certificate:

1. Run mmc.exe to open the Microsoft Management Console.
2. From the **File** menu, select **Add/Remove Snap-in**.

   The **Add or Remove Snap-ins** window opens.
3. In the left pane, double-click **Certificates**.

   The **Certificates snap-in** wizard opens.
4. Select **Computer account** and then click **Next**.
5. In the **Select Computer** screen, make sure that **Local computer** is selected and then click **Finish**.
6. Click **OK** to close the **Add or Remove Snap-ins** window.

7. In the left pane, expand the **Certificates** folder. Then, right-click the **Personal** folder and select **All Tasks > Import**.
8. In the **File to Import** screen, select your PFX certificate file. Note that by default the **Open** window displays CER files. In order to see your PFX files, you need to select **Personal Information Exchange** from the drop-down list in the bottom right of the window.
9. Click **Next** and enter the private key password.
10. Continue clicking **Next** until you reach the **Completing the Certificate Import Wizard** screen. Then click **Finish** to exit the wizard.
11. In the **Personal > Certificates** folder, double-click the newly imported certificate.

    The **Certificate** window opens.
12. Scroll down the **Details** tab until you see the **Thumbprint** details and copy them to the clipboard.
13. Open a command prompt and run the following commands:

    Syntax:

    ```
    ¢ netsh http add sslcert ipport=0.0.0.0:443 certhash=[YOUR_
    CERTIFICATE_THUMBPRINT_WITHOUT_SPACES] appid={4dc3e181-e14b-4a21-
    b022-59fc669b0914}
    ```

    Example:

    ```
    netsh http add sslcert ipport=0.0.0.0:443
    certhash=5f6eccba751a75120cd0117389248ef3ca716e61 appid={4dc3e181-
    e14b-4a21-b022-59fc669b0914}
    ```

    Syntax:

    ```
    ¢ netsh http add sslcert ipport=[::]:443 certhash=[YOUR_CERTIFICATE_
    THUMBPRINT_WITHOUT_SPACES] appid={4dc3e181-e14b-4a21-b022-
    59fc669b0914}
    ```

    Example:

    ```
    netsh http add sslcert ipport=[::]:443
    certhash=5f6eccba751a75120cd0117389248ef3ca716e61 appid={4dc3e181-
    e14b-4a21-b022-59fc669b0914}
    ```
14. Close the command prompt and Microsoft Management Console.
15. Start the Attunity Compose for Data Lakes service.

Setting up HSTS

Note Supported from Compose for Data Lakes 6.6 SP11 only.

HSTS is a web security policy mechanism that helps to protect websites against man-in-the-middle attacks such as protocol downgrade attacks and cookie hijacking. It allows web servers to declare that web browsers (or other complying Dilqam) should automatically interact with it using only HTTPS connections, which provide Transport Layer Security (TLS/SSL).

You can force the Compose for Data Lakes Web UI and/or the Compose for Data Lakes REST API connections to use HSTS (HTTP Strict Transport Security). To do this, run the commands described below.

All commands should be run from as Admin from the product **bin** folder.

Enabling HSTS

Syntax:

```
ComposeCtl.exe configuration set --static_http_headers header_list --rest_http_headers header_list
```

where `--static_http_headers` are the headers required to connect to the Compose for Data Lakes Web UI and `--rest_http_headers` are the headers required to connect using the API.

Headers should be specified using the following format:

```
ComposeCtl.exe configuration set --static_http_headers "header1:value1" "header2:value2" --rest_http_headers "header1:value1" "header2:value2"
```

Example:

The following instructs the browser to treat the domain as an HSTS host for a year (there are approximately 31536000 seconds in a year). The optional includeSubDomains directive means that subdomains (i.e. secure.myhealthcare.example.com) should also be treated as an HSTS domain.

```
ComposeCtl.exe configuration set --static_http_headers "Strict-Transport-Security:max-age=31536000; includeSubDomains;" --rest_http_headers "Strict-Transport-Security":"max-age=31536000; includeSubDomains;"
```

Disabling HSTS

You can revert to regular HTTPS connections by running the following command:

To disable static_http_headers, run:

```
ComposeCtl.exe configuration set --static_http_headers ""
```

To disable rest_http_headers, run:

```
ComposeCtl.exe configuration set --rest_http_headers ""
```

Changing the Master User Password

All passwords are encrypted using a one-time randomly generated master key. The master key is stored automatically in the root repository of Compose for Data Lakes (<product_dir>\data\projects\GlobalRepo.sqlite).

The master key is encrypted by a user key, which in turn, is derived from a master password entered by the user. By default, the Master User Password is randomly generated by Compose for Data Lakes. The best practice, however, is to change the Master User Password, as this will allow Compose for Data Lakes projects and configuration settings to be imported to another machine without needing to re-enter the project credentials.

It may also be convenient to use the same Master User Password within a trusted environment. In other words, if the same administrators control both the production and the testing environments, using the same Master User Password in both environments will facilitate the transfer of projects with credentials between the testing and production environments.

The user key is stored in the muk.dat file located in <product_dir>\data\.

Important: The Master User Password must be a minimum of 32 characters. You can either use your own password or run the `genpassword` utility described below to generate a password for you. Note also that the password can only contain alphanumeric characters (i.e. it cannot contain special keyboard characters such as # or @).

All of the commands listed below must be run from:

<product_dir>\Attunity\Compose for Data Lakes\bin

To generate a random 32 character password

Issue the following command:

```
ComposeCtl.exe utils genpassword
```

To change the randomly generated master user password

1. Issue the following command:

   ```
   ComposeCtl.exe masterukey set --password <new_master_password>
   ```

   > **Note** If you add the `--prompt` parameter to the command and omit the `--password` parameter, the CLI will prompt you for the password. When you enter the password, it will be obfuscated. This is especially useful if you do not want passwords to be retained in the command prompt history.
   >
   > **Syntax:**
   >
   > `ComposeCtl.exe masterukey set --prompt`

2. Restart the Attunity Compose for Data Lakes service.

To change a user-defined master user password:

1. Issue the following command:

   ```
   ComposeCtl.exe masterukey set --current-password <current_master_
   password> --password <new_master_password>
   ```

   > **Note** If you add the `--prompt` parameter to the command and omit the `--password` and `--current-password` parameters, the CLI will prompt you for the required passwords. When you enter the passwords, they will be obfuscated. This is especially useful if you do not want passwords to be retained in the command prompt history.
   >
   > **Syntax:**
   >
   > `ComposeCtl.exe masterukey set --prompt`

2. Restart the Attunity Compose for Data Lakes service.

Determining the Required Number of Storage Zone Connections

As a rule of thumb, the higher the number of database connections opened for Compose for Data Lakes, the more tables Compose for Data Lakes will be able to load in parallel. It is therefore recommended to open as many database connections as possible for Compose for Data Lakes. However, if the number of database connections that can be opened for Compose for Data Lakes is limited, you can calculate the minimum number of required connections as described below.

To determine the number of required connections

1. For each task, you can determine the number of connections it can use during runtime. This value is specified in the **Advanced** tab in the Modifying Task Settings window. When determining the number of required connections, various factors need to be taken into account including the number of tables, the size of the tables, and the volume of data. It is therefore recommended to determine the required number of connections in a Test environment.
2. Calculate the number of connections needed by all tasks that run in parallel. For example, if three Data Lake tasks run in parallel and each task requires 5 connections, then the number of required connections will be 15.

   Similarly, if a workflow contains two Storage Zone tasks that run in parallel and each task requires 5 connections, then the minimum number of required connections will be 10.
3. Factor in the connections required by the Compose for Data Lakes Console. To do this, multiply the maximum number of concurrent Compose for Data Lakes users by three and then add to the sum of Step 2 above. So, if the number of required connections is 20 and the number of concurrent Compose for Data Lakes users is 4, then the total would be:

   ```
   20 + 12 = 32
   ```

Accessing Qlik Compose for Data Lakes

You can use a Web browser to access the Qlik Compose for Data Lakes Console from any computer in your network. For information on supported browsers, see Preparing your System for Compose for Data Lakes.

The person logged in to the computer where you are accessing the Console must be an authorized Qlik Compose for Data Lakes user. For more information, see Setting up User Permissions.

To access the Qlik Compose Console:

1. To access the Qlik Compose for Data Lakes Console from the machine on which it is installed, select **All Programs > Qlik Compose for Data Lakes > Qlik Compose for Data Lakes Console** from the Windows **Start** menu. To access the Qlik Compose for Data Lakes Console from a remote browser, type the following address in the address bar of your Web browser

   ```
   https://<computer name>/attunitycompose_datalakes
   ```

where <`computer name`> is the name or IP address of the computer where Qlik Compose for Data Lakes is installed.

2. If no server certificate is installed on the Compose for Data Lakes machine, a page stating that the connection is untrusted will be displayed. This is because when Compose for Data Lakes detects that no server certificate is installed, it installs a self-signed certificate. Since the browser has no way of knowing whether the certificate is safe, it displays this page. For more information, see Setting Up HTTPS for the Compose for Data Lakes Console.
3. When prompted for your password, enter your domain username and password.

3 Getting Started with Qlik Compose for Data Lakes

This section provides an overview of the Qlik Compose for Data Lakes architecture, familiarizes you with its interface and ends with a short tutorial.

In this chapter:

The Qlik Compose for Data Lakes Workflow

A Qlik Compose for Data Lakes workflow is typically set up as follows (simplified):

1. In Replicate, define a task that replicates the source tables to a specific target. The target should be defined as the Landing Zone in your Qlik Compose for Data Lakes project.
2. In Compose for Data Lakes:
    a. Configure access to your Storage Zone and your Landing Zone(s).
    b. Use the "Discover" option to auto-generate the metadata from the source tables located in the Landing Zone(s). You can even create the Metadata manually if you prefer.
    c. Optionally, create the Storage Zone tables and then generate the ETL commands that will be executed when the task runs.
    d. Run the tasks to move the data from the Landing Zone to the Storage Zone as follows:
        i. **In an Apache Spark project:** Run the single Full Load and CDC task that was automatically created when the source tables were discovered.
        ii. **In a Apache Hive project:** Run the separate Full Load and CDC tasks (in that order) that were automatically created when the source tables were discovered.
    e. In an Apache Spark project, define a Provisioning task that moves selected data from the Storage Zone to the Provisioning Zone.

See also Introduction .

Introducing the Qlik Compose for Data Lakes Interface

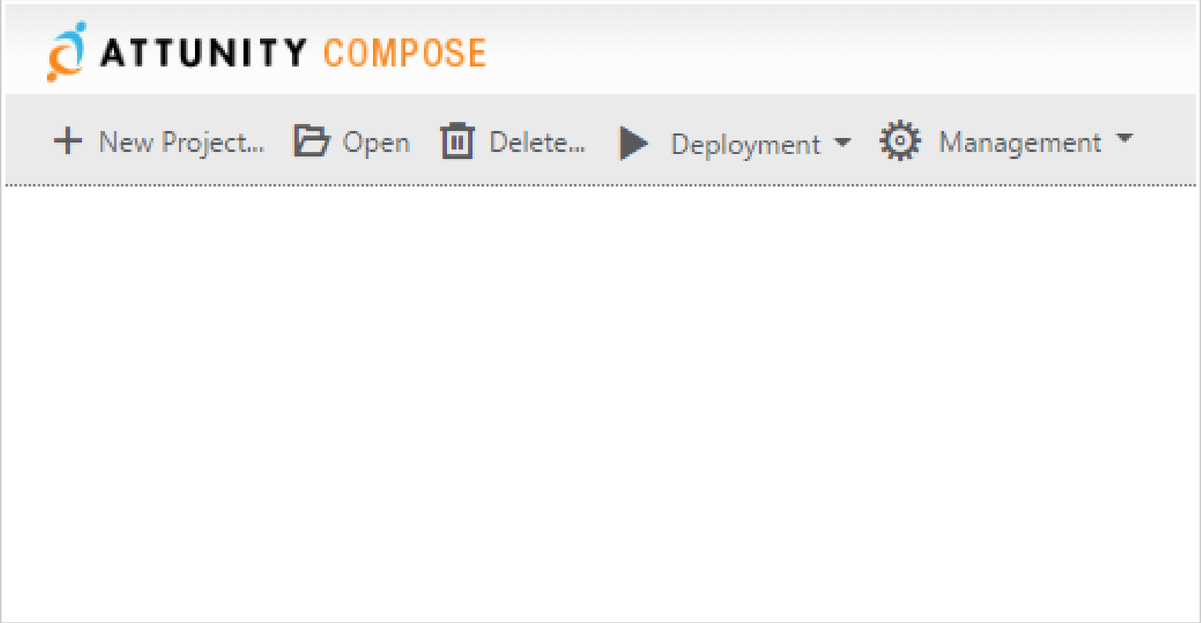
This section will familiarize you with the elements that comprise the Qlik Compose for Data Lakes UI.

To open Qlik Compose for Data Lakes:

From the Windows **Start** menu, select **All Programs > Qlik Compose for Data Lakes > Qlik Compose for Data Lakes Console**.

The Qlik Compose for Data Lakes Console opens in Management view.

Figure 4.1 | Qlik Compose for Data Lakes Console - Management View



Management View

In **Management** view, you can manage the following:

- Qlik Compose for Data Lakes projects

  For more information, see Adding and Managing Projects .
- The product license
- Replicate Server connections
- Compose Agent connection
- Log levels and cleanup options

- Email settings
- User permissions

For more information, see Managing Compose for Data Lakes .

Designer View

When you add a new project or open an existing project, the console switches to **Designer** view. You can switch back and forth between **Designer** view and **Monitor** view by clicking the **Designer** and **Monitor** tabs in the top right of the console.

Designer view comprises the following panels:

- **Landing and Storage Connections** - Configure access to your Landing Zone(s) and Storage Zone.

  For more information, see Defining Landing Zones and Defining a Connection to the Storage Zonerespectively.

- **Storage Zone** - In the Storage Zone, you can:
  - Discover and manage the source table metadata.
  - Define data storage tasks that move the data from the Landing Zone(s) to the Storage Zone.

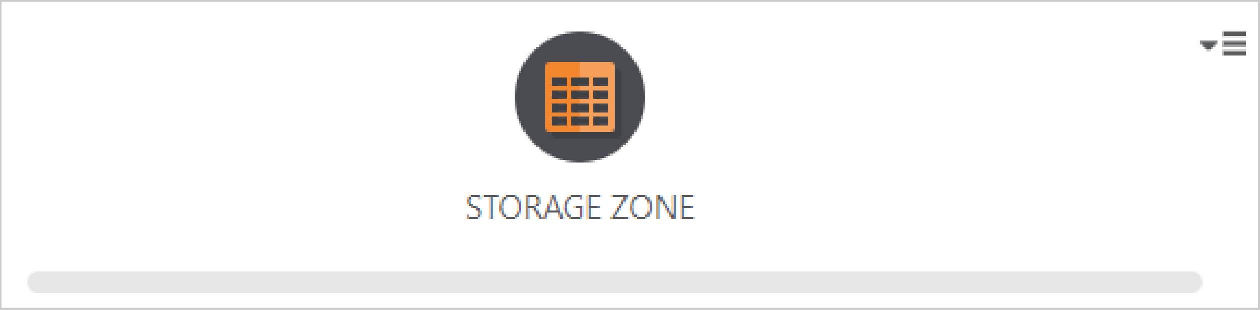
  For more information, see Selecting Source Tables and Managing Metadata and Creating and Managing Storage Zone Tasks .

- **Provisioning Zone** - Only relevant in a Qlik Compose for Data Lakes for Spark project. Define provisioning tasks that move selected data from the Storage Zone to one of the available Provisioning Zone.

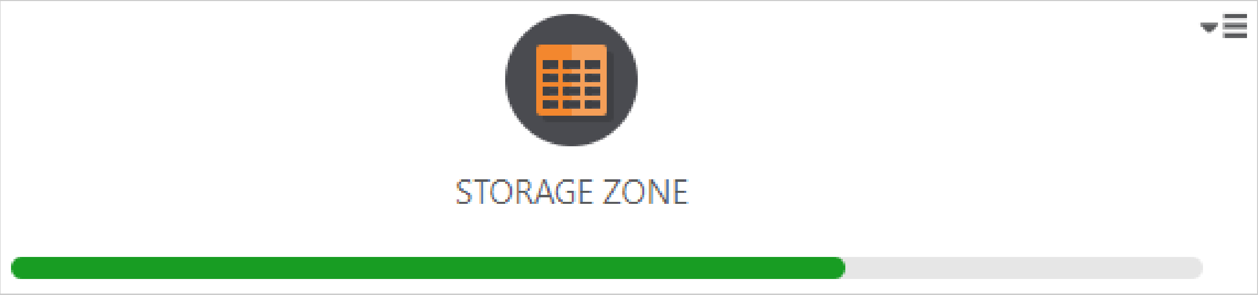
  For more information, see Creating and Managing Provisioning Tasks.

In Designer view, each of the panels has a bar below the panel name. The bar can be empty, half-filled or completely filled, according to the current configuration status of the panel properties, as follows:

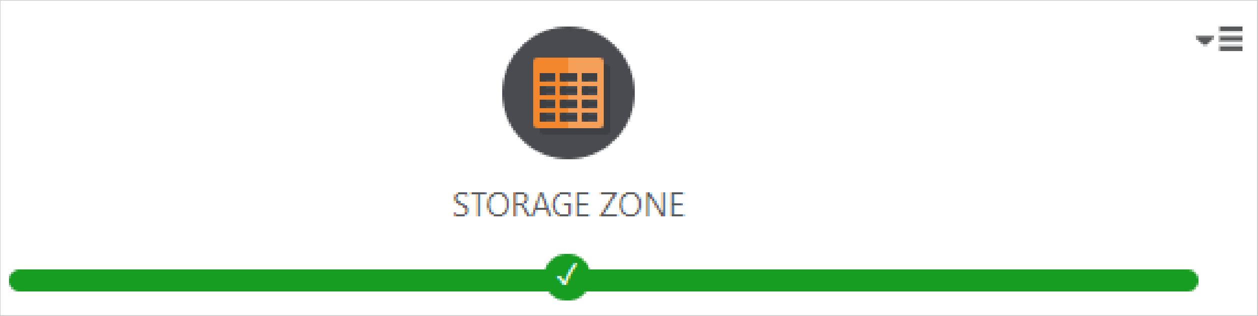
No fill (gray) - Not configured



Half filled - Configuration is not complete



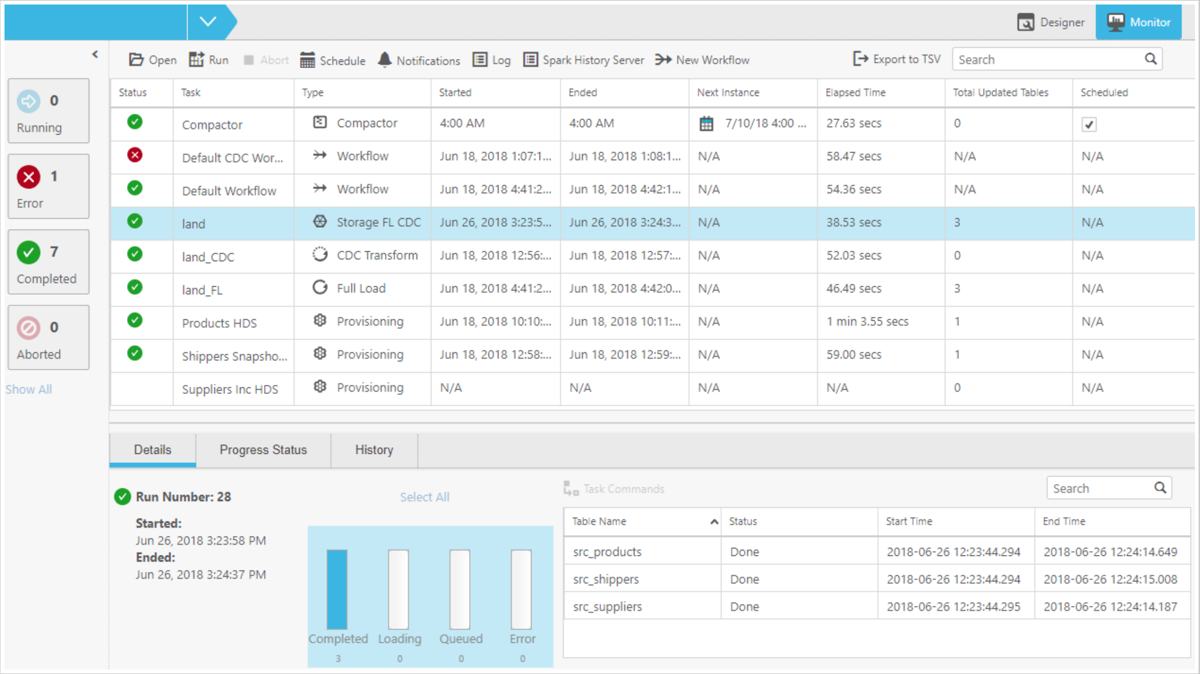
Completely filled - Fully configured



Monitor View

To switch to Monitor view, click the **Monitor** tab in the top right of the console.

Figure 4.2 | Monitor View



Status	Task	Type	Started	Ended	Next Instance	Elapsed Time	Total Updated Tables	Scheduled
✓	Compactor	Compactor	4:00 AM	4:00 AM	7/10/18 4:00 ...	27.63 secs	0	☑
✗	Default CDC Wor...	Workflow	Jun 18, 2018 1:07:1...	Jun 18, 2018 1:08:1...	N/A	58.47 secs	N/A	N/A
✓	Default Workflow	Workflow	Jun 18, 2018 4:41:2...	Jun 18, 2018 4:42:1...	N/A	54.36 secs	N/A	N/A
✓	land	Storage FL CDC	Jun 26, 2018 3:23:5...	Jun 26, 2018 3:24:3...	N/A	38.53 secs	3	N/A
✓	land_CDC	CDC Transform	Jun 18, 2018 12:56:...	Jun 18, 2018 12:57:...	N/A	52.03 secs	0	N/A
✓	land_FL	Full Load	Jun 18, 2018 4:41:2...	Jun 18, 2018 4:42:0...	N/A	46.49 secs	3	N/A
✓	Products HDS	Provisioning	Jun 18, 2018 10:10:...	Jun 18, 2018 10:11:...	N/A	1 min 3.55 secs	1	N/A
✓	Shippers Snapsho...	Provisioning	Jun 18, 2018 12:58:...	Jun 18, 2018 12:59:...	N/A	59.00 secs	1	N/A
	Suppliers Inc HDS	Provisioning	N/A	N/A	N/A	N/A	0	N/A

Run Number: 28

Started:
Jun 26, 2018 3:23:58 PM
Ended:
Jun 26, 2018 3:24:37 PM

Table Name	Status	Start Time	End Time
src_products	Done	2018-06-26 12:23:44.294	2018-06-26 12:24:14.649
src_shippers	Done	2018-06-26 12:23:44.294	2018-06-26 12:24:15.008
src_suppliers	Done	2018-06-26 12:23:44.295	2018-06-26 12:24:14.187

In Monitor view, you can view the status of Qlik Compose for Data Lakes tasks, schedule their execution (either individually or as a workflow), view logs, and create notifications.

For more information, see Controlling and Monitoring Tasks and Workflows .

Defining a Qlik Replicate Task

In order to work with Compose for Data Lakes, you first need to define a Qlik Replicate task that replicates the source tables from the source endpoint to a target endpoint (i.e. the Landing Zone). A connection to the Landing Zone should then be defined in the Compose for Data Lakes project.

- Supported target endpoint types include Hadoop, Amazon EMR, Microsoft Azure HDInsight, Google DataProc, and HortonWorks Data Platform.
- For supported Replicate versions, see Supported Replicate Versions.

The steps below highlight the settings that are required when using Qlik Replicate with Compose for Data Lakes. For a full description of setting up tasks in Qlik Replicate, please refer to the *Qlik Replicate Setup and User Guide*.

- If the Landing Zone supports append, it is recommended to select **Sequence** as the file format in the Replicate target endpoint settings and to set the Control Tables format (if available) to **Text**. This will improve performance by allowing Replicate to append to the file instead of creating a new file for every Change Data Partition.

  If the above is not possible, then it is recommended to periodically delete files that are no longer required from the target directory. This will prevent files from amassing and degrading performance.
- When Microsoft Azure HDInsight is defined as the Replicate target endpoint, you must set the **Target storage format** to **Sequence**.
- When Oracle is defined as the source endpoint in the Replicate task, full supplemental logging should be defined for all source table columns that exist on the target and any source columns referenced in expressions.
- When Hadoop is defined as the source endpoint in the Replicate task, the Hadoop user name must have the same case as the Compose user name (preferably lower case); otherwise, Compose won’t be able to see the table in Hadoop.

- Replicate allows you to define global transformations that are applied to source/Change tables during task runtime. The following global transformations, however, should not be defined (as they are not compatible with Compose for Data Lakes tasks):
    - Rename Change Table
    - Rename Change Table schema

To define the task:

1. Open Qlik Replicate and define a new task.
    - To enable Full Load and Change Processing replication, enable the **Full Load** and **Store Changes** options (the **Apply Changes** option should not be enabled).
    - To enable Full Load replication only, make sure that only the **Full Load** option is enabled.
    - To enable Change Processing replication only, make sure that only the **Store Changes** option is enabled. Note that this option should only be selected if the Full Load tables and data already exist in the Landing Zone.
    - To enable change processing for lookup tables that already exist in the Landing Zone and are not part of the Compose for Data Lakes Metadata, it is recommended to create an Apply Changes only Data Storage task within Compose for Data Lakes. Note that such a task should be defined in addition to the **Full Load** and **Store Changes** replication task described above.
2. Open the **Manage Endpoint Connections** window and define a source and target endpoint. The target endpoint must be the database where you want Compose for Data Lakes to create the Storage Zone tables.
3. Add the endpoints to the Replicate task and then select which source tables to replicate.
4. To facilitate Automatic Schema Evolution in Compose for Data Lakes:
    - If you want Hive to be updated with any new source tables that are added during the Replicate task, you must define **Table Selection Patterns** in the **Select Tables** window.
    - Enable the **DDL History** Control Table in the Task Settings' **Metadata|Control Tables** tab.
5. In the Task Settings' **Change Processing|Store Changes Settings** tab, enable **Change Data Partitioning**.
6. In the Task Settings' **Metadata|Control Tables** tab, select the **Change Data Partitioning** Control Table.
7. Run the task.

Wait for the Full Load replication to complete and then continue the workflow in Compose for Data Lakes as described in Adding and Managing Projects .

4 Adding and Managing Projects

This chapter describes how to add and manage Compose for Data Lakes projects.

In this chapter:

Adding Projects

Adding a new project is the first task you need to undertake in order to work with Qlik Compose for Data Lakes. You can set up as many projects as you need.

> **IMPORTANT** To prevent unpredictable behavior, each project *must* be defined with a dedicated Storage Zone.

There are three project types:

- **Apache Spark** - Uses Spark to move data from the Landing Zone to the Storage Zone and provision selected data further downstream.

  Before you define a Qlik Compose for Data Lakes for Spark project, you first need to install the Qlik Compose for Data Lakes Agent.

- **Apache Hive** - Uses HiveQL to move data from the Landing Zone to the Storage Zone.
- **Databricks** - Uses Databricks Delta on Azure to move data from the Landing Zone to the Storage Zone.

To create a new project:

1. Click the **New Project** toolbar button.

   The **New Project** window opens.

2. Specify a name for your project.

   > **Note** The following names are reserved system names and cannot be used as project names: `CON, PRN, AUX, CLOCK$, NUL, COM1, COM2, COM3, COM4, COM5, COM6, COM7, COM8, COM9, LPT1, LPT2, LPT3, LPT4, LPT5, LPT6, LPT7, LPT8` and `LPT9`.

3. Select **Apache Hive** (the default), **Apache Spark**, or **Databricks** as the project type.

   > **Note** You must have a registered license to the project type in order to select it.

   - If you selected Apache Spark, the **New Project** wizard opens. Continue from Adding an Apache Spark Project.
   - If you selected Apache Hive, the **New Project** window opens. Continue from Adding an Apache Hive Project.
   - If you selected Databricks, the **New Project** window opens. Continue from Adding a Databricks Project.

Adding an Apache Spark Project

> **Note** ACID Transactions can be used only when supported by the underlying distribution.

1. In the **Storage** screen:
   a. From the **Type** drop-down list, select the desired storage type.
   b. If you selected **Amazon S3** or **Google Storage**, specify the target **Bucket name**.
   c. If you selected **Microsoft Azure HDInsight**, specify the **ADLS URL** for accessing your ADLS storage (Microsoft Azure Data Lake Storage Gen1 or Microsoft Azure Data Lake Storage Gen2).
   d. Specify the **Target folder** for the files.
2. Click **Next**.
3. In the **Hive connection** screen, the following settings can be configured:
   - **Name** - The display name of your storage definitions.
   - Use SSL - Select to connect using SSL.
     - Use self-signed certificate - Select to connect using a self-signed certificate.
     - Trusted store full path - Enter the full path to the store containing your trusted certificates.
     - Trusted store password - Enter the password for your trusted certificate store.
   - Authentication Type - Choose one of the following:
     - **Azure HDInsight** - Select if your Hadoop cluster is located on Azure HDInsight. Then, in the **User name** and **Password** fields, specify the name and password of a user authorized to access the Hadoop cluster.
     - **Knox** - Select this option if you need to access the Hortonworks Hadoop distribution through a Knox Gateway. Then, provide the following information:
       - **Host** - The FQDN (Fully Qualified Domain Name) of the Knox Gateway host.
       - **Knox port** - The port number to use to access the host. The default is "8443".
       - **Knox Gateway path** - The context path for the gateway. The default is "gateway".

        - The port and path values are set in the gateway-site.xml file. If you are unsure whether the default values have been changed, contact your IT department.
        - As Compose for Data Lakes automatically appends “/hive” to the specified path, do not include "/hive" in the specified path. Doing so will cause the connection to fail.
    - **Cluster name** - The cluster name as configured in Knox. The default is "Default".
    - **User name** - Enter your user name for accessing the Knox gateway.
    - **Password** - Enter your password for accessing the Knox gateway.
- **Kerberos** - Select to authenticate against the Hadoop cluster using Kerberos. Then, provide the following information:
    - **Realm:** The name of the realm in which your Hadoop cluster resides.

      For example, if the full principal name is john.doe@EXAMPLE.COM, then EXAMPLE.COM is the realm.
    - **Principal:** The user name to use for authentication. The principal must be a member of the realm entered above. For example, if the full principal name is john.doe@EXAMPLE.COM, then john.doe is the principal.
    - **Keytab file:** The full path of the Keytab file. The Keytab file should contain the key of the Principal specified above.

      > **Note** The keytab file should be created by running the Windows “ktpass” command. For a full description of this command, refer to the Microsoft online help for Windows Server commands.

      The krb5.ini file should be located in **C:\Windows** (according to the Java default). However, if Replicate is installed on the same machine as Compose for Data Lakes, the file might be in **C:\Program Files\MIT\Kerberos**. In such a case, simply copy the file to **C:\Windows**.
    - **Host:** The FQDN that will be used to locate the correct Principal in Kerberos. This is only required if the IP address of the Hive machine is not known to Kerberos.
    - Service name: The default is "hive". You should only change this if

you are sure that the service name is different.

If you encounter an issue with Kerberos authentication, do the following:

- Test the connection to the Hive machine with Kerberos.
- Check the Kerberos configuration on HDFS.
- Check the configuration on the Spark machine with Kerberos.
- Validate the `kinit` and `klist` commands on the Compose Agent machine.

- **User name** - Select to connect to the Hadoop cluster with only a user name. Then, in the **User name** field, specify the name of a user authorized to access the Hadoop cluster.
- **User name and password** - Select to connect to the Hadoop cluster with a user name and password. Then, in the **User name** and **Password** fields, specify the name and password of a user authorized to access the Hadoop cluster.

  If you are unsure about any of the above, consult your IT administrator.

- Type - Choose one of the following data store types:
  - HDFS
  - Amazon S3
  - Azure Data Lake Storage Gen1
  - Azure Data Lake Storage Gen2
  - Google Cloud Storage
- Bucket name - If you chose **Amazon S3** or **Google Cloud Storage** as your data store, specify the target bucket name.
- ADLS URL - f you chose **Azure Data Lake Storage Gen1** as your data store, specify the URL for your ADLS storage.
- If you chose **Azure Data Lake Storage Gen2** as your data store, specify:
  - Storage account - The account for your ADLS storage.
  - File system - A file system for the ADLS storage.
  - Target folder - The target folder of the ADLS storage files.
- Target folder - Specify the target folder for the files.
- Use ZooKeeper - Select if your Hive machines are managed by Apache ZooKeeper.
- ZooKeeper hosts - The machines that make up the ZooKeeper ensemble

(cluster). These should be specified in the following format:

host1:port1,host2:port2,host3:port3

- ZooKeeper namespace - The namespace on ZooKeeper under which the HiveServer2 znodes are located.
- Host - If you are not using ZooKeeper, specify the IP address of the Hive machine. This should be the same as the host name or IP address specified in the target endpoint settings in the Replicate task.

  Supported target endpoint types include Hadoop, Amazon EMR, Microsoft Azure HDInsight, Google DataProc, and HortonWorks.
- Port - If you are not using ZooKeeper, optionally change the default port.
- Database name - Specify the name of the Hive target database. This must be different from the database specified in the Landing Zone settings.

  To prevent table name conflicts:
  - In a Compose for Data Lakes with Spark project, the Landing Zone, Storage Zone, and Provisioning Zone databases should be different.
  - In a Compose for Data Lakes with Hive project, the Landing Zone and Storage Zone databases should be different.
- JDBC parameters - Additional parameters to add to the default Simba JDBC connection string. These should be key values separated by a semi-colon.

  Example:

  KEY=VALUE;KEY1=VALUE1

  You can set Hive parameters in the JDBC parameters. For example:
  - `mapred.job.queue.name=<queuename>`
  - `hive.execution.engine=<enginename>`
- File format - With Compose for Data Lakes Spark projects, the file format is set to **Parquet**. With Compose for Data Lakes Hive projects, the default is **ORC**.
- Use buckets - This option is not relevant for Compose for Data Lakes Spark projects. With Compose for Data Lakes Hive projects, this option must be enabled for ACID support. Increasing the number of buckets may improve performance in certain situations.

  For more information, visit:

  https://community.hortonworks.com/questions/23103/hive-deciding-the-number-of-buckets.html

4. Click **Finish**.

   The newly added project will open in **Designer** view.

Adding an Apache Hive Project

1. Optionally, provide a **Description**.
2. Select one of the following options under **ACID Transactions**, according to whether your Hive distribution type supports ACID transactions:
   - Create tables with ACID transactions
   - Create tables without ACID transactions
3. Select one of the following **Project Types**:
   - **Operational Data Store** - Maintains a replica of the source data. Utilizes ACID for updates and optimization.
   - **Historical Data Store with ACID** - Maintains a replica of the source data with history. Utilizes ACID to optimize reads.
   - **Historical Data Store** - Maintains a replica of the source data with history.
4. Select what action Compose for Data Lakes should perform in the Storage Zone when DELETE operations are performed on the source tables.

   When a record is marked as deleted, Compose for Data Lakes performs a soft delete.

   Choose one of the following:
   - Mark the matching Storage record as deleted
   - Mark the matching Storage record as deleted in history tables, but delete the record from other tables.

     Note Records with expressions, lookup, or derived attributes on Primary Keys will not be deleted.
   - Do nothing
5. Click **OK**.

   The newly added project will open in **Designer** view.

Adding a Databricks Project

Databricks is currently supported on Azure. Delta is required by Compose in order to create a replica of the source data (Operational Data Store - ODS). Replicate first creates external tables on the metadata store, and when running Full Load and Store Changes tasks, it writes the files to Azure storage. Similar to other endpoints, Replicate creates change data partitions in the Partition Control Table and in the metadata store which is read by Compose. Compose creates delta tables for the ODS which includes inserted, modified and deleted data that is updated on every Compose task run.

Prerequisites

- **Driver** - Before creating a Databricks project, you first need to download and install the Databricks JDBC driver as follows:
    1. Download Databricks JDBC driver **SparkJDBC41.jar** from the Databricks website.
    2. Copy the downloaded file to <product_dir>\java\jdbc.
    3. Restart the **Attunity Compose for Data Lakes** service.
- **Permissions** - Databricks users must have privileges to SELECT, CREATE, MODIFY, and READ_METADATA from the relevant database.

To add a Databricks project:

1. In the **New Project** window, provide a **Description** (optional).
2. Select the **Operational Data Store** project type.

   This project type maintains a replica of the source data. It utilizes Delta for updates and optimization.
3. Select what action Compose for Data Lakes should perform in the Storage Zone when DELETE operations are performed on the source tables.

   Choose one of the following:
    - Mark the matching Storage record as deleted
    - Delete the matching Storage record.
4. Click **OK**.

   The newly added project will open in **Designer** view.

Managing Projects

The table below describes the available shortcut options for managing projects.

Project management operations are performed in the main Compose for Data Lakes window. To switch to the main window from the project window, either click the Compose for Data Lakes logo or click the downward arrow to the right of the project name and select **All Projects** from the drop-down menu.

To	Do this
Open a project for viewing or editing	Any of the following: » Double-click the project. » Right-click the project and select **Designer**. » Select the project and then click the **Open** toolbar button.
Delete a project	Any of the following: » Right-click the project and select **Delete**. » Select the project and then click the **Delete** toolbar button.
Monitor a project's tasks	Right-click the project and select Monitor.
Create a Deployment Package	Right-click the project and select Create Deployment Package.

Editing the Project Settings

You can change the project settings according to your needs.

To open the project settings window

1. Open your project as described in Managing Projects.
2. Click the downward arrow to the right of the project name and select **Settings** from the drop-down menu.

   The **Settings** window opens.

   The project settings window is divided into the following tabs:

   - General Tab
   - Naming Tab
   - Defaults Tab

» Variables Tab

Note The **Variables** tab is not available for Apache Spark projects.

General Tab

The **General** tab contains the following sections:

- **Project Details** - Displays the properties you set when you created your Compose for Data Lakes project. These settings cannot be changed. For more information about these settings, see Adding Projects.

  The **Project Details** section is not available for Apache Spark projects.
- **Miscellaneous** - Various project-related settings that you can enable as required. These are described in **Miscellaneous Settings** below.

Miscellaneous Settings

- **Generate DDL scripts but do not run them** - By default, Compose for Data Lakes executes the CREATE, ADJUST and DROP statements immediately upon user request. When you select this option, Compose for Data Lakes will only generate the scripts but not execute them. This allows you to review and edit the scripts before they are executed.

  This option is not relevant for Apache Spark projects.

  For example, if you want to apply custom sorting or special formatting, you will need to edit the CREATE statement.

  Note that if you select this option, you will need to copy the scripts to your Storage Zone and run them manually. You can view, copy and download the DDL scripts as described in Viewing and Downloading DDL Scripts.

  > **Note** The UI is not refreshed after manually running the DDL scripts (when the "Generate DDL scripts but do not run them" option is selected). For example, if you run a script that creates the Storage Zone tables, the UI will not show the tables until the display is refreshed.
  >
  > **Workaround:**
  >
  > Press F5 to refresh the browser display.
- **Ignore Mapping Data Type Validation** - By default, Compose for Data Lakes issues a validation error when a Landing Zone table is mapped to a Storage Zone table with a different data type. You can select this option to allow the mapping of different data types. Note that you should only select this option if you need to map Landing Zone table data types to compatible (though not identical) Storage Zone table data types.
- **Do not display the default workflows in the monitor** - Select this option if

you want to prevent the default workflows from being displayed.

- **Exclude the "To Date" column from tables with history** - Select this option if you do not want the Compose for Data Lakes "To Date" column to be included in tables with history. Changing this option also requires you to recreate any existing tables in your Storage Zone.

Naming Tab

Optionally, change the settings in the **Naming** tab according to the descriptions in the table below.

Note If you change the prefix or suffix of existing tables, you need to drop and create the tables.

Table 5.1 | Naming Tab

Name	Description
Prefix for Compose for Data Lakes Columns	Compose for Data Lakes adds its own columns to the Data Lake database according to the project settings. Examples of such columns include the **From Date** and **To Date** columns described below. You can change the default prefix (header__) of these columns.
header__ FROM_DATE Column Name	The name of the "From Date" column. This column is added to tables that contain attributes (columns) with history. The column is used to delimit the range of dates for a given record version. **Notes** - This name cannot be used in other columns. - NULL values are not allowed in this column. If the source "From Date" column contains NULLs, an expression should be created to convert them to non-null values.
To Date Column Name	The name of the "To Date" column. This column is added to tables that contain attributes (columns) with history. The column is used to delimit the range of dates for a given record version. This name cannot be used in other columns.

Table 5.1 | Naming Tab

Name	Description
Replicate Change Table Suffix	The suffix used to identify Replicate Change Tables in the landing area of the Storage Zone.
Archived Change Table Suffix	**Note** This option is not relevant to Apache Spark or Databricks projects. The suffix used to identify archived Change Tables in the specified database. For more information on archiving Change Tables, see After applying changes.
Storage Data Table Prefix	The prefix to add to table names in the Storage Zone. Changing this after the Storage Zone tables have already been created would require you drop and recreate your Storage Zone tables.
Storage Control Table Prefix	**Note** This option is not relevant to Apache Spark projects. The prefix (by default, `attrep_`) to add to the Control Tables in the Storage Zone, to enable reuse of the same database for Landing, Storage and Provisioning Zones.
View Prefix	The prefix to add to view names in the Storage and Provisioning Zones. Note that changing the prefix after the tables have already been created requires you to drop and recreate your tables.
HDS Current View Suffix	The current view consists of the Storage and Provisioning Zone tables without historical records. Note that changing the suffix after the tables have already been created requires you to drop and recreate your tables.
Column Indicating a Soft Delete	This column is selected in the Storage Zone when the Mark the matching target record as deleted (Soft Delete) option is selected and the corresponding source record has been deleted.

Table 5.1 | Naming Tab

Name	Description
Dropped Column Suffix	**Note** This option is not relevant to Apache Spark projects. You can change the default suffix (__dropped) of columns that are dropped.

Defaults Tab

Optionally, change the settings in the **Defaults** tab according to the descriptions in the table below.

Table 5.2 | Defaults Tab

Name	Description
Lowest Date	The value stored in the "From Date" column. This is the default value for records without any other indication (like CDC info). It can be changed to use Now instead.
Highest Date	The value stored in the "To Date" column. This value is the +infinity used for new records that are known to exist in the Storage Zone.
Provisioning Root	
HDFS	The root directory on HDFS where the provisioned files will be stored. A subdirectory with the provisioning task name will be created under the root directory.
Amazon S3 Bucket	The name of the Amazon S3 bucket to which the provisioned files will be uploaded.
Bucket root folder	The root folder in the Amazon S3 bucket to which the provisioned files will be uploaded. A subfolder with the provisioning task name will be created under the root folder.

Variables Tab

Note This tab is not available for Apache Spark projects.

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 Data Lake table, you could setup a project that replaces the variables with the location of each of the factories. For an example, see Consolidation Implementation Example.

In the **Variables** tab, you can create a list of variables for use in your project. Any variables that you create will be displayed in the Source Landing Zone definitions, allowing you to provide values (i.e. data) for each of the variables. You can then create a new attribute in the Data Lake table and define an expression that replaces the variables with the specified data during data ingestion.

Manage variables as described in the table below.

Table 5.3 | Variable Management

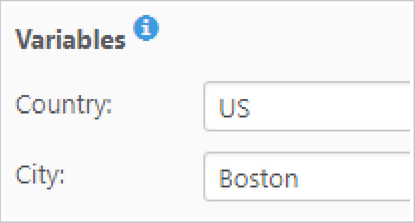
To	Do This
Add a variable	Click the **New** button and then type the name of the variable in the edit field.
Edit a variable	Click the variable and edit as required.
Delete a variable	Select the unwanted variable and then click **Delete**.

Consolidation Implementation Example

A multi-national enterprise wishes to consolidate data from its factories located in five different cities, three in the US and three in the UK. To accomplish this with Compose for Data Lakes, the project would need to be defined as follows:

1. Define two variables as described above: **Country** and **City**.
2. Create five different Source Landing Zones, one for each city.
3. The variables that you defined in Step 1 will be displayed in the **Variables** section of the Source Landing Zone settings. Enter the name of the country and the city where the data is located.

   For example:

   Variables

   Country: US

   City: Boston

4. In the **Manage Metadata** window, create two attributes (a **Country** and a **City**

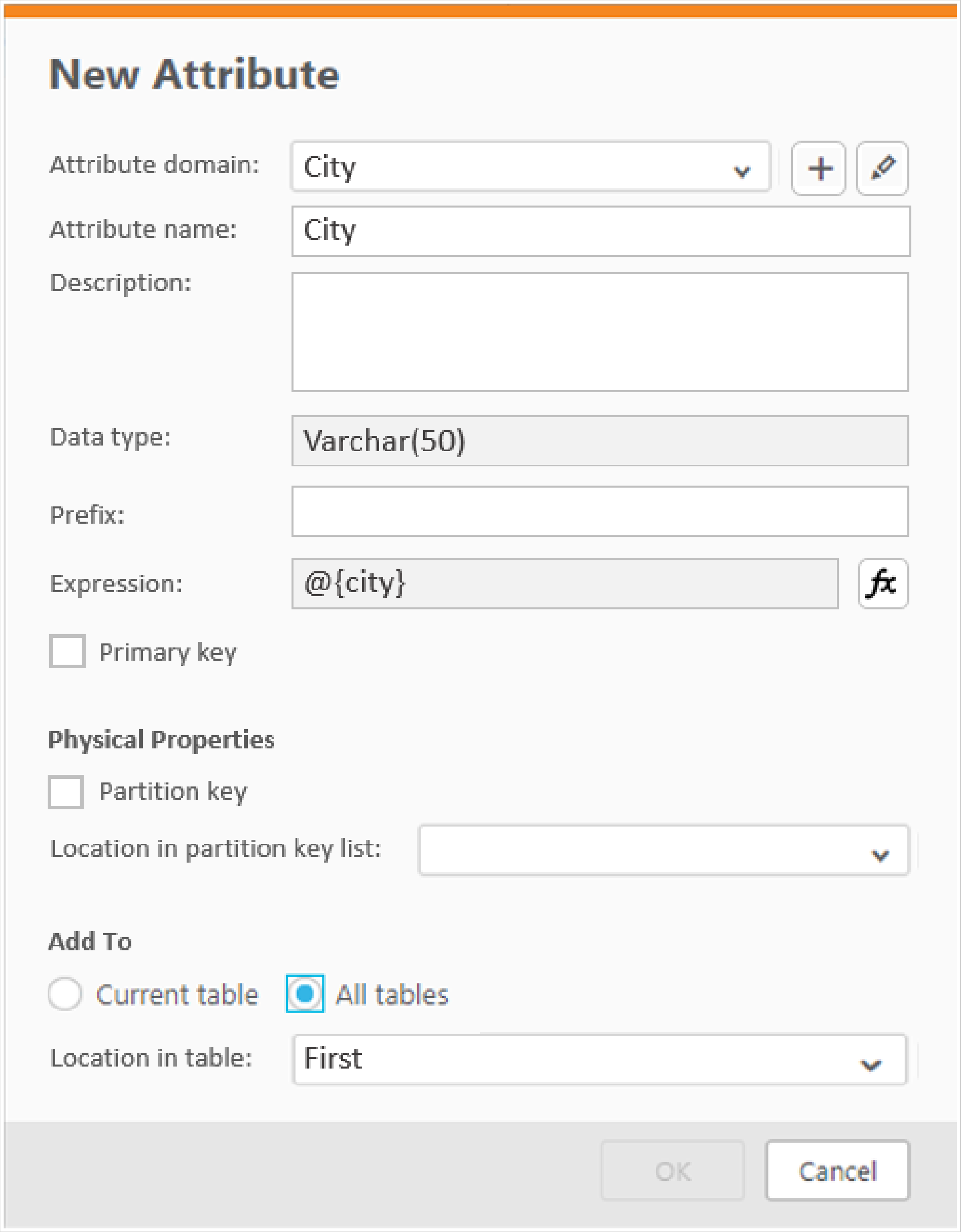
attribute) as follows:

a. When you add each attribute, manually create an expression for the variable in the following format: `@{variable}`

   Example: `@{City}`

   Note that if the attribute does not exist, you will first need to add it to the Attributes Domain. Note also that the attribute name is case-sensitive.

   For information on adding attributes to the metadata and adding attributes to the attributes domain, see Managing Attributes.



New Attribute

Attribute domain: City

Attribute name: City

Description:

Data type: Varchar(50)

Prefix:

Expression: @{city}

□ Primary key

Physical Properties

□ Partition key

Location in partition key list:

Add To

○ Current table ● All tables

Location in table: First

b. In the **Apply to** field, select **All tables**.

c. From the **Location in table** drop-down list, select **First**.

5. Create and generate the Storage Zone tables as described in Creating and Managing the Storage Zone.

Resetting Projects

You can reset projects as required. This can be useful in the project development stage as it allows you to easily delete unwanted project elements. Be careful not to reset a project and delete data in a production environment!

To reset a project:

1. Open your project as described in Managing Projects.
2. Click the downward arrow to the right of the project name and select **Reset Project** from the drop-down menu.

   The **Reset Project** window opens.
3. Choose to reset any of the following:
   - Metadata, mappings, data storage and provisioning tasks

     For information on mappings, data storage tasks and metadata, see Selecting Source Tables and Managing Metadata.

     For information on provisioning tasks, see Selecting Source Tables and Managing Metadata
   - Storage tables and files

     For information on the Storage Zone, seeCreating and Managing Storage Zone Tasks .
   - Provisioning tables and files

     Relevant to Apache Spark projects only.

     For information on provisioning tasks, see Selecting Source Tables and Managing Metadata
   - Command tasks

     For more information, see Creating and Managing Command Tasks .
   - DDL scripts

     For more information on DDL scripts, see Editing the Project Settings and Viewing and Downloading DDL Scripts.
4. Type "confirm" to enable the **Yes** button and then click **Yes**.

Project Deployment

Project deployment packages can be used to back up projects or migrate projects between different environments (e.g. testing to production). As a deployment package is intended to be deployed in a new environment, it contains the Storage Zone and data source definitions, but without any passwords. The deployment package also does not contain any data from the Storage Zone, only the metadata. The deployment package also contains the project metadata and mapping information, which should be consistent with the Landing Zone tables in the new environment.

For a complete list of objects contained in the deployment package, see Exporting a Project.

Creating Deployment Packages

This section explains how to create a project deployment package.

To create a deployment package

1. Choose one of the following methods:
   - In the main Compose for Data Lakes window, right-click the desired project and select **Create Deployment Package** from the context menu.
   - In the main Compose for Data Lakes window, select the desired project. Then, click the **Deployment** toolbar button and select **Create Deployment Package** from the drop-down menu.
   - In the project window, select **Deployment** > **Create Deployment Package** from the project drop-down 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**.

   A ZIP file containing a JSON file (i.e. the project settings) and a **readme.txt** file will be saved to your browser's default download location.

   The ZIP file name is in the following format:

   ```
   <Project_Name>_deployment_<Date>__<Time>.zip
   ```

   The **readme.txt** file contains the following information about the deployment package: project name, export date, exporter user name, deployment version, and description.

Deploying Packages

This section explains how to deploy a project deployment package. You can only deploy packages to an existing project. Therefore, before deploying a project, create a new project with the user name and password required for connecting to the Storage Zone in the new environment.

When deploying, Compose for Data Lakes does not override existing connection parameters, assuming they are environment-local. This enables you to easily migrate projects from test to production, for example, without the need to change user names, passwords or IP addresses.

If preferred, you can create an empty project - by clicking **New Project**, entering a project name, clicking **Create New Project**, and then clicking **Cancel** - and provide the required credentials after the deployment completes. In this case, an error message prompting you for the missing credentials will be displayed after the deployment completes.

To deploy a project deployment package

1. Copy the ZIP file created in Creating Deployment Packages to a location that is accessible from the Compose for Data Lakes machine.
2. Open Compose for Data Lakes and choose one of the following methods:
   - In the main Compose for Data Lakes window, select the desired project. Then, click the **Deployment** toolbar button and select **Deploy** from the drop-down menu.
   - In the project window, select **Deployment** > **Deploy** from the project drop-down menu.

   The **Deploy** window opens.
3. Either drag and drop the file on the window.

   -OR-

   Click **Select** and browse to the location of the deployment package. 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.
4. Click **Deploy** to deploy the package. When prompted to replace the existing project, confirm the operation.

   The project will be deployed.

Exporting and Importing Projects using the CLI

Important: Compose for Data Lakes CLI requires Administrator permission. To grant Administrator permission, select "Run as administrator" when opening the command prompt.

Under normal circumstances, use the deployment options described in Project Deployment to export and import projects. For deployment automation or control by another tool, you can use the command line interface (CLI) to perform the following tasks:

- Exporting a Project
- Importing a Project
- Exporting the Project Configuration
- Importing the Project Configuration

To export or import a project or project configuration including passwords, you first need to change the default Master User Password.

For more information on changing the master user password, see Changing the Master User Password.

See also: Moving Projects from the Test Environment to the Production Environment and Import/Export Scenarios - When is a Password Required?

Before running any command, you must run the Connecting to the Qlik Compose for Data Lakes Server command.

To get help when using the command line, you can run the Help command. For example, for help about exporting a project, issue the following command:

```
ComposeCli.exe export_project_repository --help
```

This brings up a list of help parameters.

Connecting to the Qlik Compose for Data Lakes Server

Run the `Connect` command to establish a connection to the Qlik Compose for Data Lakes Server. You must run this command before running any other command:

Syntax:

```
ComposeCli.exe connect [--url connection-url]
```

where:

- `url` is the connection URL to the system where the server is running, such as https://machine.domain/attunitycompose_datalakes. This is only required if the server is running on a remote machine.

Example:

```
ComposeCli.exe connect --url https://mymachine.mydomain/attunitycompose_
datalakes

Compose Control Program started...

Compose Control Program completed successfully.
```

Exporting a Project

You can use the Compose for Data Lakes CLI to export a project.

Exported projects include the following:

- Data zone connections
- Metadata definitions (entities and attributes)
- Mappings between Landing Zone and Storage Zone table columns.
- Storage Zone ETL tasks
- Provisioning tasks on Spark projects
- Project settings

Existing Storage Zone tables and generated task instructions are not exported. Notifications and schedules are also not exported as they are considered to be environment-specific.

Syntax:

```
ComposeCli.exe export_project_repository --project project_name --outfile
output-file [--is_without_credentials] [--password password] [--master_
user_password master_user_password]
```

where:

- **project** is the name of the project you want to export.
- **outfile** specifies the path to and name of the output file. This file is in JSON format. For example: C:\file.json.
- **is_without_credentials** specifies to export the project settings without the encrypted fields. When importing to a new project, you will need to manually enter the project passwords (in the Compose for Data Lakes database connection settings) after the import completes. In addition to eliminating the need to specify a password when exporting or importing the project, the **is_without_credentials** parameter also allows the project to be used in every Compose for Data Lakes installation, regardless of its master user password. It is also useful in the event that you would like to keep the existing passwords in the target environment (e.g.

when exporting from a testing environment to an existing project in the production environment).

- **`password`** specifies the password for encrypting the credentials in the exported project. The **`password`** qualifier must be used together with the **`master_user_password`** qualifier described below. Use the **`password`** qualifier if you want to encrypt the credentials in the exported project, but do not want the source master password to be used in a different environment. The specified password must be at least 32 characters in length and can either be user-devised or generated using the **`genpassword`** utility described in Changing the Master User Password.
- **`master_user_password`** the master user password defined for the source machine. This must be used together with the **`password`** qualifier. Use the **`master_user_password`** qualifier if you want to encrypt the credentials in the exported project, but do not want the source master password to be used in a different environment. In such a case, when you import the project to an environment that has a different master password, you will only need to specify the **`password`** qualifier.

  For instructions on changing the master user password, see Changing the Master User Password.

  See also: Moving Projects from the Test Environment to the Production Environment and Import/Export Scenarios - When is a Password Required?

Importing a Project

You can use the Compose for Data Lakes CLI to import a project. If you import to an existing project, all of the project settings, except the project configuration items will be overridden. For information on the project configuration items, see Exporting the Project Configuration.

Imported projects include the following:

- Data zone connections
- Metadata definitions (entities and attributes)
- Mappings between Landing Zone and Storage Zone table columns.
- Storage Zone ETL tasks

Syntax:

```
ComposeCli.exe import_project_repository --project project_name --infile
input-file [--password password] [--is_without_credentials] [--override_
configuration] [--dont_backup_existing_project]
```

where:

- **`project`** is the name of the project you want to import.
- **`infile`** specifies the full path to the input file (including the file name). This file is in JSON format. For example: C:\file.json
- **`override_configuration`** overrides the existing project configuration. When importing a project, the default is *not* to override the existing project configuration.
- **`dont_backup_existing_project`** specifies not to back up the existing project. By default, existing projects are backed up to the following location (and automatically restored if the import fails):

  `<product_dir>\data\projects\<project_name>_backup_<timestamp>`
- **`is_without_credentials`** specifies to import the project settings without the encrypted fields. In this case, you will need to manually enter the project passwords (in the Compose for Data Lakes database connection settings).
- **`password`** the password specified with the password qualifier during export.

  For instructions on changing the master user password, see Changing the Master User Password.

Existing Storage Zone tables and generated task instructions are not imported. After the import completes, you must perform step 3 below. You may also need to perform step 1 or 2, depending on whether you changed the Storage Zone connection settings (step 1) or kept the existing connection settings (step 2).

1. If you changed the Storage Zone connection settings after importing the project, then you need to create the tables in the new Storage Zone.
2. If you edited the Metadata in a testing environment and then imported the project into a production environment, you need to validate and adjust the Storage Zone.
3. Generate the Data Storage task instructions.

For information on validating the Storage Zone and generating the task instructions, see Creating and Managing Storage Zone Tasks .

See also: Moving Projects from the Test Environment to the Production Environment and Import/Export Scenarios - When is a Password Required?

Exporting the Project Configuration

You can use the Compose for Data Lakes CLI to export the configuration settings of an existing project. This includes Landing and Storage Connections, scheduling jobs, and notifications. This is helpful, for example, when you need to migrate configuration settings from a test environment to the production environment.

For information about migrating projects, see Moving Projects from the Test Environment to the Production Environment.

Syntax:

```
ComposeCli.exe export_project_repository_config --project project_name --
outfile output file [--is_without_credentials] [--password password] [--
master_user_password master_user_password]
```

where:

- **project** is the name of the project you want to export.
- **outfile** specifies the path to and name of the output file. This file is in JSON format. For example: C:\file.json.
- **is_without_credentials** specifies to export the project configuration without the encrypted fields. When importing to a new project, you will need to manually enter the Landing Zone(s) and Storage Zone passwords (in the Connections panel) after the import completes. In addition to eliminating the need to specify a password when exporting or importing the project, the **is_without_credentials** parameter also allows the project configuration to be used in every Compose for Data Lakes installation, regardless of its Master User Password. It is also useful in the event that you would like to keep the existing passwords in the target environment (e.g. when exporting from a testing environment to an existing project in the production environment).
- **password** specifies the password for encrypting the credentials in the exported project configuration. The **password** qualifier must be used together with the **master_user_password** qualifier described below. Use the **password** qualifier if you want to encrypt the credentials in the exported project configuration, but do not want the source master password to be used in a different environment. The specified password must be at least 32 characters in length and can either be user-devised or generated using the **genpassword** utility described in Changing the Master User Password.
- **master_user_password** the master user password defined for the source machine. This must be used together with the **password** qualifier. Use the **master_user_password** qualifier if you want to encrypt the credentials in the exported project configuration, but do not want the source Master User Password to be used in a different environment. In such a case, when you import the project configuration to an environment that has a different Master User Password, you will only need to specify the **password** qualifier.

  For instructions on changing the Master User Password, see Changing the Master User Password.

See also: Moving Projects from the Test Environment to the Production Environment and Import/Export Scenarios - When is a Password Required?

Importing the Project Configuration

You can use the Compose for Data Lakes CLI to import the configuration settings of an existing project. This includes Data Zone definitions, scheduling jobs, and notifications. This is helpful, for example, when you need to migrate configuration settings from a test environment to the production environment. For information about migrating projects, see Moving Projects from the Test Environment to the Production Environment.

Before you can import the project configuration, you must first run the `import_project_repository` command described in Importing a Project.

Syntax:

```
ComposeCli.exe import_project_repository_config --project project_name --
infile input-file [--password password] [--is_without_credentials]
```

where:

- **project** is the name of the project you want to export.
- **infile** specifies the path to and name of the input file. This file is in JSON format. For example: C:\file.json.
- **is_without_credentials** specifies to import the project configuration without the encrypted fields. In this case, you will need to manually enter the project's Landing Zone and Storage Zone passwords (in the Data Zone Connection settings).
- **password** the password specified with the password qualifier when the project configuration was exported.

  For instructions on changing the Master User Password, see Changing the Master User Password.

  See also: Moving Projects from the Test Environment to the Production Environment and Import/Export Scenarios - When is a Password Required?

Moving Projects from the Test Environment to the Production Environment

After successfully creating and testing projects in the test environment, you now want to move those projects to the production environment. You also need to propagate updates from the testing environment to the production environment as necessary. Although it sounds complicated, moving new and updated projects from the test environment to the production environment is actually quite straightforward, as explained below.

See also Import/Export Scenarios - When is a Password Required?.

Landing and Storage Connections (landing, storage and provisioning) will not be overridden when moving to a production environment. This also includes the file format set in the provisioning task.

The Landing Zone and Storage Zone display names must be identical in both the testing and the production environments.

To perform the initial migration from the testing environment to the production environment:

1. Export the project from the test environment as described in Exporting a Project.
2. Import the test project to the production environment as described in Importing a Project.
3. Edit the connection settings to point to the production Landing Zone and Storage Zone.

   For more information, see Defining Landing Zones and Defining a Connection to the Storage Zonerespectively.
4. Configure notifications and scheduling as needed.

   For more information, see Scheduling Tasks and Defining Notifications Rules respectively.

To propagate updates from the testing environment to the production environment:

1. Export the project from the test environment as described in Exporting a Project.
2. Import the test project to the production environment as described in Importing a Project.

Import/Export Scenarios - When is a Password Required?

The following section describes which of the various export/import scenarios require a password to be specified.

In all scenarios, if you import a project to an existing project, the credentials of the existing projects are preserved (as they are part of the project configuration).

Scenario 1: Moving a project or project configuration between two Compose for Data Lakes machines without retaining the project credentials. This is useful when importing to a new project that will have different project credentials.

In such a scenario, simply add the `is_without_credentials` parameter to either the export or the import command.

Scenario 2: Moving a project or project configuration between two Compose for Data Lakes machines that have the same Master User Password.

In such a scenario, neither the export command nor the import command need to include a password. If you do not want the source and target projects to have the same credentials (for Data Zone connectivity, etc.), then you also need to specify the `is_without_credentials` parameter in either the export or the import command.

Scenario 3: Moving a project or project configuration between two Compose for Data Lakes machines that have a different Master User Password, but without revealing the Master User Password of the source machine.

In such a scenario, the export command must include the `password` and `master_user_password` parameters while the import command must include the `password` parameter. The same password (specified with the `password` parameter) must be used for both export and import.

Scenario 4: Moving a project or project configuration between two Compose for Data Lakes machines that have a different Master User Password.

In such a scenario, the export command does not need to include a password, but the import command should specify the Master User Password of the source machine (using the `password` parameter).

Viewing and Downloading DDL Scripts

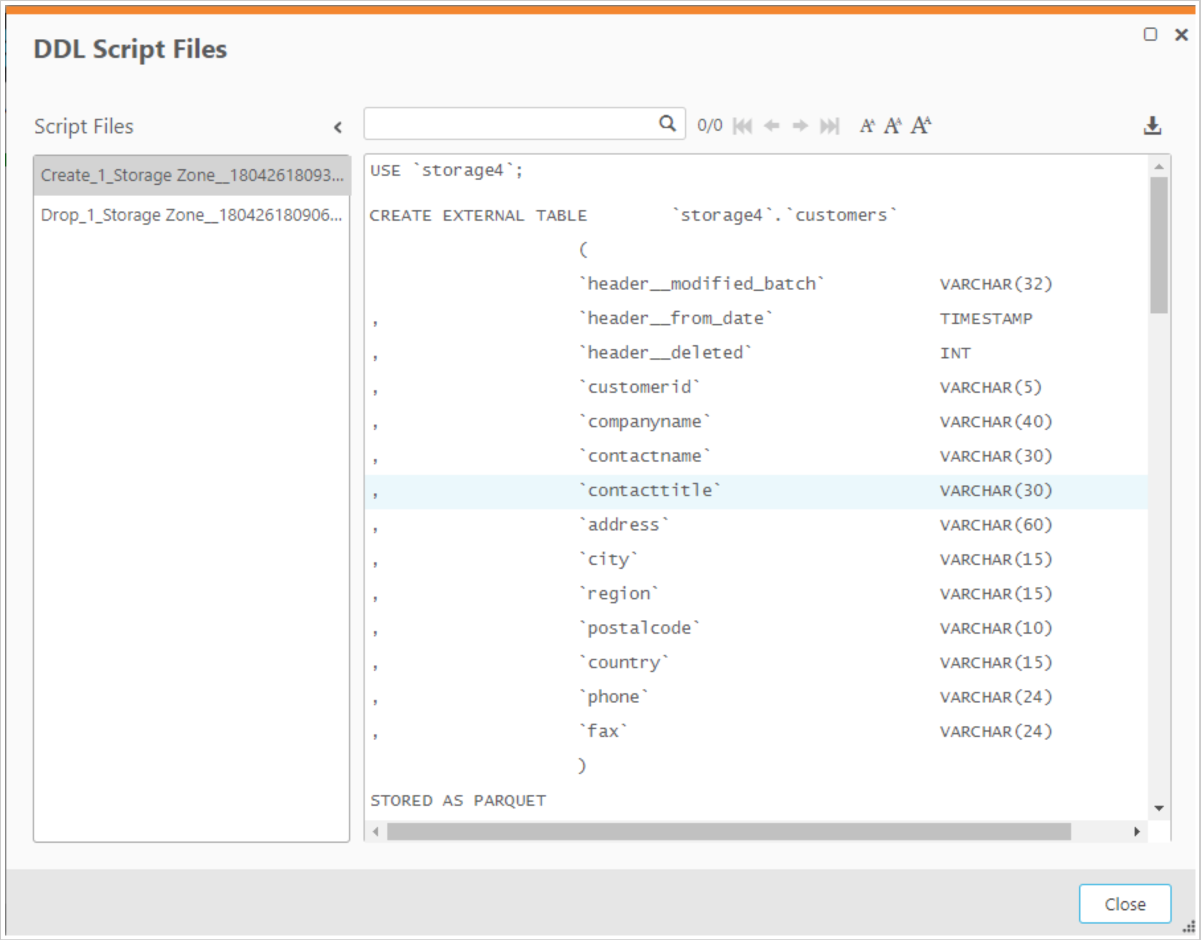
In the **DDL Script Files** window, you can view and download the Storage Zone DDL script files. By default, Compose for Data Lakes executes the **Create**, **Adjust** and **Drop** statements immediately upon user request. However, when the **Generate DDL scripts but do not run them** option is enabled, Compose for Data Lakes will only generate the scripts but not execute them.

For more information on the **Create DDL scripts only** option, see Editing the Project Settings.

To open the DDL Script Files window:

1. Open your project as described in Managing Projects.
2. Click the downward arrow to the right of the project name and select **Show DDL Scripts** from the drop-down menu.

   The **DDL Script Files** window opens.


DDL Script Files
Script Files
Create_1_Storage Zone__18042618093...
Drop_1_Storage Zone__180426180906...
0/0
Close

```
USE `storage4`;

CREATE EXTERNAL TABLE          `storage4`.`customers`
                        (
                        `header__modified_batch`            VARCHAR(32)
,                       `header__from_date`                 TIMESTAMP
,                       `header__deleted`                   INT
,                       `customerid`                        VARCHAR(5)
,                       `companyname`                       VARCHAR(40)
,                       `contactname`                       VARCHAR(30)
,                       `contacttitle`                      VARCHAR(30)
,                       `address`                           VARCHAR(60)
,                       `city`                              VARCHAR(15)
,                       `region`                            VARCHAR(15)
,                       `postalcode`                        VARCHAR(10)
,                       `country`                           VARCHAR(15)
,                       `phone`                             VARCHAR(24)
,                       `fax`                               VARCHAR(24)
                        )
STORED AS PARQUET
```

3. To view a script, select the desired script in the **Script Files** pane on the left. The script will be displayed on the right.
4. To download a script, select the desired script in the **Script Files** pane on the left. Then click the download button in the top right of the window.
5. To search for an element in the script, start to type in the search box. All strings that match the search query will be highlighted blue.

   You can navigate between search query matches using the arrows to the right of the search box. Use the right and left single arrows to navigate matches sequentially. Use the right and left double arrows to jump to the last and first match respectively.
6. To reset the search, either delete the search query or click the "x" in the right of the search box.

Project Versioning

Compose for Data Lakes provides built-in project version control using the Git engine. Version control enables Compose for Data Lakes developers to commit project revisions to both a local and a remote Git repository. If a mistake is made, Compose for Data Lakes developers can easily roll back to earlier versions of the project while minimizing disruption to all team members.

Revisions only store metadata and mapping information. After you revert to a saved revision, you will need to recreate the Storage Zone tables.

Configuring Version Control Settings

To define Version Control Settings

1. From the project drop-down menu, select **Version Control** > **Settings**.

   The **Version Control Settings - Git** window opens.

   The **Local Commits** area shows the local root folder where project revisions are committed. The first time a project revision is committed, Compose for Data Lakes creates a JSON file with the current project settings. The `<project_name>.json` file is archived to a ZIP file (`<project_name>_deployment.zip`), which is located in a project-specific folder under the **source-control** folder.

2. To enable commits to a remote Git database, select **Enable remote commits** and then provide the following information:

   - **URL** - The address of the remote Git database.
   - **User name** - Your user name for accessing the remote Git database.
   - **Password** - Your password for accessing the remote Git database.

Committing Projects

You can commit a project using the console or using the CLI:

To commit a project to Version Control using the UI:

1. From the project drop-down menu, select **Version Control** > **Commit**.

   The **Commit - <Project_Name>** window opens.

2. Enter a message in the **Message** box and optionally select the **Remote push** check box. Note that the **Remote push** check box will be disabled if the **Enable remote commits** option described above is not selected.

To commit a project to Version Control using the CLI:

Run the following command from the Compose for Data Lakes **bin** directory:

```
ComposeCli.exe commit --project project_name [--message message] [--
remote]
```

Where:

- `commit` is the verb.
- `project_name` is the name of the project you want to commit.
- `message` is an optional message to accompany the commit.
- `remote` is required if you want to commit the project to a remote Git repository (see above). By default, the project will be committed locally to **<product_dir>\data\source-control**.

Saved Revisions

To revert to a saved revision

1. From the project drop-down menu, select **Version Control** > **Revisions history**.

   The **Revision History- <Project_Name>** window opens.

   By default, the last 10 revisions are shown. You can change this number by selecting one of the available options from the **Show** drop-down list.
2. Optionally, use the **Search** box to find a specific revision.
3. Select the desired revision and then click the **Deploy to Revision** toolbar button.
4. When prompted to confirm the operation, click **Yes**.

   The existing project will be replaced.
5. Click **Close** to close the **Revision History- <Project_Name>** window.

To download a saved revision

1. From the project drop-down menu, select **Version Control** > **Revisions history**.

   The **Revision History- <Project_Name>** window opens.

   By default, the last 10 revisions are shown. You can change this number by selecting one of the available options from the **Show** drop-down list.
2. Optionally, use the **Search** box to find a specific revision.
3. Select the desired revision and then click the **Download Revision as Package** toolbar button.

   The package will be saved as a ZIP file in your browser's default download location.

5 Setting up Landing and Storage Connections

This chapter explains how to configure connectivity to your Storage Zone and Landing Zone(s).

In this chapter:

Defining a Storage Zone

This section explains how to set up Storage Zone connectivity in a Qlik Compose for Data Lakes project.

In this section:

Defining a Connection to the Storage Zone

As the server connection settings for the Landing Zone are derived from the Storage Zone settings, you must define a Storage Zone first.

For more information on adding data sources, see Defining Landing Zones .

To define the Storage Zone connection:

1. Open your project and click the **Connections** button in the bottom left of the **Landing and Storage Connections** panel.

   The **Manage Landing and Storage Connections** window opens.

2. Either, click the **Add New Storage** link in the middle of the window.

   -OR-

   Click the **New** toolbar button and then select **Storage** from the drop-down menu.

   The **New Storage Zone** window opens.

3. Enter the information as described in the table below.

Field	Description
Name	The display name of your storage definitions.
Security	
Use SSL	Select to connect using SSL.
Use self-signed certificate	Select to connect using a self-signed certificate.
Trusted store full path	Enter the full path to the store containing your trusted certificates.
Trusted store password	Enter the password for your trusted certificate store.

Field	Description
Authentication Type	» **Azure HDInsight** - Select if your Hadoop cluster is located on Azure HDInsight. Then, in the **User name** and **Password** fields, specify the name and password of a user authorized to access the Hadoop cluster. For Databricks projects, authentication requires a user name, password and HTTP path. Token authentication is supported by specifying "token" as the user name (the default) and the token value as the password. » **Knox** - Select this option if you need to access the Hortonworks Hadoop distribution through a Knox Gateway. Then, provide the following information: » **Host** - The FQDN (Fully Qualified Domain Name) of the Knox Gateway host. » **Knox port** - The port number to use to access the host. The default is "8443". » **Knox Gateway path** - The context path for the gateway. The default is "gateway". The port and path values are set in the gateway-site.xml file. If you are unsure whether the default values have been changed, contact your IT department. » **Cluster name** - The cluster name as configured in Knox. The default is "Default". » **User name** - Enter your user name for accessing the Knox gateway. » **Password** - Enter your password for accessing the Knox gateway. » **Kerberos** - Select to authenticate against the Hadoop cluster using Kerberos. Then, provide the following information: » **Realm:** The name of the realm in which your Hadoop cluster resides. For example, if the full principal name is john.doe@EXAMPLE.COM, then EXAMPLE.COM is the realm.

Field	Description
	- **Principal:** The user name to use for authentication. The principal must be a member of the realm entered above. For example, if the full principal name is john.doe@EXAMPLE.COM, then john.doe is the principal. - **Keytab file:** The full path of the Keytab file. The Keytab file should contain the key of the Principal specified above. **Note** The keytab file should be created by running the Windows “ktpass” command. For a full description of this command, refer to the Microsoft online help for Windows Server commands. The krb5.ini file should be located in **C:\Windows** (according to the Java default). However, if Replicate is installed on the same machine as Compose for Data Lakes, the file might be in **C:\Program Files\MIT\Kerberos**. In such a case, simply copy the file to **C:\Windows**. - **Host:** The FQDN that will be used to locate the correct Principal in Kerberos. This is only required if the IP address of the Hive machine is not known to Kerberos. - **Service name:** The default is "hive". You should only change this if you are sure that the service name is different. In case of an issue with the Kerberos authentication, do the following: 1. Test the connection to the Hive machine with Kerberos. 2. Check the Kerberos configuration on HDFS. 3. Check the configuration on the Spark machine with Kerberos. 4. Validate the `kinit` and `klist` commands on the Compose Agent machine.

Field	Description
	» **User name** - Select to connect to the Hadoop cluster with only a user name. Then, in the **User name** field, specify the name of a user authorized to access the Hadoop cluster. » **User name and password** - Select to connect to the Hadoop cluster with a user name and password. Then, in the **User name** and **Password** fields, specify the name and password of a user authorized to access the Hadoop cluster. If you are unsure about any of the above, consult your IT administrator.
Data Store - Only relevant for Apache Spark projects.	
Type	Choose one of the following types: » HDFS » Amazon S3 » Azure Data Lake Storage Gen1 » Azure Data Lake Storage Gen2 » Google Cloud Storage
Bucket name	If you chose **Amazon S3** or **Google Cloud Storage** as your data store, specify the target bucket name.
ADLS URL	If you chose **Azure Data Lake Storage Gen1** as your data store, specify the URL for your ADLS storage.
If you chose **Azure Data Lake Storage Gen2** as your data store, specify:	

Field	Description
Storage account	The account for your ADLS storage.
File system	A file system for the ADLS storage.
Target folder	The target folder of the ADLS storage files.
Target folder	Specify the target folder for the files.
Hive Access	
Use ZooKeeper	This is not relevant for Databricks. Select if your Hive machines are managed by Apache ZooKeeper.
ZooKeeper hosts	The machines that make up the ZooKeeper ensemble (cluster). These should be specified in the following format: `host1:port1,host2:port2,host3:port3`
ZooKeeper namespace	The namespace on ZooKeeper under which the HiveServer2 znodes are located.
Host	If you are not using ZooKeeper, specify the IP address of the Hive machine. This should be the same as the host name or IP address specified in the target endpoint settings in the Replicate task. Supported target endpoint types include Hadoop, Amazon EMR, Microsoft Azure HDInsight, Google DataProc, and HortonWorks.
Port	If you are not using ZooKeeper, optionally change the default port.
Database name	Specify the name of the Hive target database. This must be different from the database specified in the Landing Zone settings. To prevent table name conflicts: » In a Compose for Data Lakes with Spark project, the Landing Zone, Storage Zone, and Provisioning Zone databases should be different. » In a Compose for Data Lakes with Hive or Databricks project, the Landing Zone and Storage Zone databases should be different.

Field	Description
JDBC parameters	Additional parameters to add to the default Simba JDBC connection string. These should be key values separated by a semi-colon. **Example:** `KEY=VALUE;KEY1=VALUE1` **Notes** » You can set Hive parameters in the JDBC parameters. For example: » `mapred.job.queue.name=<queuename>` » `hive.execution.engine=<enginename>` » To distinguish Compose Hive sessions from other Hive Sessions, if Tez is being used, you can define a JDBC parameter to change the query description, as follows: `hive.query.name=my_description`
Hive metadata storage type	Select how you want to store your Hive metadata.
Hive Metastore	This is the default metadata storage type.
AWS Glue Data Catalog	When Amazon S3 is your data type, you can choose to store Hive metadata using the AWS Glue Data Catalog. AWS Glue allows you to store and share metadata in the AWS Cloud in the same way as in a Hive metastore. **Notes** » If a change is made to the data store type in an existing Data Lake, you should either **Drop and Recreate** the existing tables or make sure that the tables exist in the Glue storage at the time of the change. » When using AWS Glue Data Catalog for metadata storage, Compose for Data Lakes control tables will be created with the data type STRING instead of VARCHAR (LENGTH).
Target Table Parameters	

Field	Description
File format	With Compose for Data Lakes Spark projects, the file format is set to Parquet. With Compose for Data Lakes Hive projects, the default file format is ORC. Renaming a column in Parquet or Avro format will cause the loss of all data in that column.
Use buckets	This option is not relevant for Compose for Data Lakes Spark or Databricks projects. With Compose for Data Lakes Hive projects, this option must be enabled for ACID support. Increasing the number of buckets may improve performance in certain situations. For more information, visit: https://community.hortonworks.com/questions/23103/hive-deciding-the-number-of-buckets.html

4. Click **Test Connection** to verify that Compose for Data Lakes is able to establish a connection with the specified database.
5. Click **OK** to save your settings.

   The database is added to the list on the left side of the **Manage Connections** window.

Data Types

The following table shows the default mapping from Compose for Data Lakes data types to Apache Hive and Databricks data types.

Qlik Compose for Data Lakes Data Types	Hive Data Types	Databricks Data Types
INTEGER	INT	INT
DATETIME	TIMESTAMP	TIMESTAMP
TIME	TIMESTAMP	TIMESTAMP
DATE	DATE	DATE
BIGINT	REAL	BIGINT

Qlik Compose for Data Lakes Data Types	Hive Data Types	Databricks Data Types
BYTE ARRAY	STRING	STRING
DECIMAL	DECIMAL (P,S)	DECIMAL (P,S)
GUID	VARCHAR (38)	STRING
VARCHAR	VARCHAR (LENGTH)	VARCHAR (LENGTH)
STRING	STRING	STRING

Required Permissions

The following permissions are required:

- **Metadata:** Read and Write
- **Tables:** Insert and Update, and Delete.

Permissions that are different when the Compose for Data Lakes project type is Apache Spark and ODS is set as the provisioning type:

- **Tables:** INSERT and UPDATE

Defining Landing Zones

This section explains how to set up Landing Zone connectivity in a Qlik Compose for Data Lakes project.

In this section:

- Landing Zone Permissions
- Defining Landing Zones Connections

Landing Zone Permissions

For proper operation, the Landing Zone database must be granted the following permissions:

- Read metadata
- Select from tables

For information on configuring the Landing Zone, see Defining Landing Zones Connections.

Defining Landing Zones Connections

In a Qlik Compose for Data Lakes project, you can define any number of Landing Zone connections. Defining multiple Landing Zone connections is necessary if the data that you eventually want to be available in your Storage Zone is located in several Landing Zones.

> **IMPORTANT** Databricks projects can have only one Landing Zone per project.

The Landing Zone connection settings tell Compose for Data Lakeswhere the replica source tables are located. Since theLanding Zoneis always located on the Storage ZoneServer and the Storage Zoneconnection details have already been defined, you do not need to provide them again.

- Before you can define a Landing Zone connection in Qlik Compose for Data Lakes, you first need to define a Storage Zone connection.
- If several tasks are reading from the same Replicate Landing Zone, you must specify the associated Replication tasks.

For more information on defining a Storage Zone connection, see Defining a Connection to the Storage Zone.

To define a Landing Zone connection:

1. Open your project and click **Connections** in the **Landing and Storage Connections** panel.

   The **Manage Landing and Storage Connections** window opens.

2. Click the **New** toolbar button and then select **Landing Connection**.

   If you prefer to import the connection details from the target endpoint of the Replicate task, select **Landing from Replicate Task** instead and then continue from Importing Landing Zone Connection Settings from a Replicate Task below.

3. In the **Name** field, specify a display name for your Landing Zone.

4. From the Content type drop-down list, choose whether the content in the landing area is **Full Load**, **Change Processing** or **Full Load and Change Processing** (according to the Qlik Replicate task definition).

   See also **After applying changes** below.

5. In the Database name field, specify the database name.

   This must be the same as the Hive access database defined in the Qlik Replicate target endpoint (in the Qlik Replicate task).

To prevent table name conflicts:

- In a Compose for Data Lakes for Spark project, the Landing Zone, Storage Zone, and Provisioning Zone databases should be different.
- In a Compose for Data Lakes for Hive project, the Landing Zone and Storage Zone databases should be different.

For more information, see Defining a Qlik Replicate Task.

6. After applying changes -

   This setting is not relevant for Apache Spark projects.

   If you selected **Change Processing** or **Full Load and Change Processing** as the **Content Type**, you can determine whether the Change Tables will be deleted or archived after the changes have been applied (to the Storage Zone tables). If you select **Archive the Change Tables**, you also need to specify a **Database name** and storage **Format** in the designated fields.

7. **Associate with Replicate Task** - Select this to associate your Compose for Data Lakes project with the related Replicate task. Replicate tasks replicate the relevant tables from the source database to the Landing Zone. Specifying the Replicate task name will enable you to monitor and control that task from within Compose for Data Lakes.

   Before you can choose a Replicate task, you first need to define the connection settings to the Qlik Replicate Server machine. To do this, click the **Manage Replicate Servers** link below the **Task** field and then configure the settings as described in Managing Replicate Servers.

   To select a Replicate task:

   a. Click **Select Task**.

      The **Select Replicate task** window opens.

   b. From the **Server** drop-down list, select the desired server.

      A list of tasks will be shown in the **Replicate Tasks** box.

   c. Select the desired task and then click **OK**.

   Server - The name of the selected Replicate server.

   Task - The name of the selected task.

8. **Automatic Schema Evolution**

   Select this option to always keep the tables in the Storage Zone up-to-date with the latest changes to the source schema.

- Automatic Schema Evolution applies only to Change Processing data storage tasks and not to Full Load.
- Automatic Schema Evolution requires the Replicate task to be set up accordingly. For details, see Defining a Qlik Replicate Task.

When this option is selected, 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.

When using AWS Glue Data Catalog for metadata storage, Compose for Data Lakes control tables will be created with the data type STRING instead of VARCHAR (LENGTH).

The following changes are supported:

- Create table
- Drop column
- Add column
- Changes will only be applied if there is at least one new record.
- DROP COLUMN - The column isn't actually dropped. Rather, the column name is appended with the `__dropped` suffix in the metadata.
- The DROP COLUMN DDL is not supported with Apache Spark or Databricks projects.
- ADD COLUMN - Compose for Data Lakes adds columns to the end of the table, regardless of their position in the source.
- Any DDL handling limitations that are relevant to Qlik Replicate are also relevant to Compose for Data Lakes. For information on these limitations, refer to the *Qlik Replicate Setup and User Guide*.
- Schema evolution applies only to Change Data Partitions that have already been closed by Replicate.
- If Compose for Data Lakes fails to apply a DDL to the target (e.g. ADD COLUMN), a warning will be written to the log and the task will continue.
  - After resetting a Compose for Data Lakes project without reloading the Replicate task, the Full Load tables will be updated with the new structure even though the DDL History table might already contain the changes. To prevent this, best practice is to delete or back up the Replicate DDL History records which were already applied before running the Schema Evolution.

9. **Variables** - Any variables that you defined for your project will appear in this

section. Provide values for the variables as required.

This setting is only relevant for Apache Hive projects.

For information on defining and implementing variables, see Editing the Project Settings.

10. Click **Test Connection** and then, if the connection is successful, click **OK** to save your settings.

Importing Landing Zone Connection Settings from a Replicate Task

The **Landing from Replicate Task** option saves time by auto-populating the connection fields with the connection settings defined in the target endpoint of the associated Replicate task.

To define a Landing Zone using the Landing from Replicate Task option:

1. Open your project and click **Connections** in the **Landing and Storage Connections** panel.

   The **Manage Landing and Storage Connections** window opens.
2. Click the **New** toolbar button and then select **Landing from Replicate Task**.

   The **New from Replicate Task** window opens.
3. Click the **Select Task** button.

   The **Select Replicate Task** window opens
4. Select a Replicate server and then select a task. If you have not yet set up any Replicate server connections, click the Managing Replicate Servers button to add a Replicate Server.
5. Optionally, change the default **Data Source Name** (which is the display name of the source endpoint in the Replicate task).
6. Click **OK** to add the connection details.

   A new Source Landing Zone will be added to the left pane of the **Manage Connections** window.
7. Optionally enable **Automatic Schema Evolution**, provide variable values and edit other settings as described in the table above.

Managing Landing and Storage Connections

You can edit and delete Landing and Storage Connections as required. The table below describes the available options.

To	Do this
Edit a Data Zone connection	In the left side of the **Manage Landing and Storage Connections** window, select the desired Data Zone (Landing Zone or Storage Zone) and then click the **Edit** toolbar button.
Delete a Data Zone connection	In the left side of the **Manage Landing and Storage Connections** window, select the desired Data Zone (Landing Zone or Storage Zone) and then click the **Delete** toolbar button. Click **Yes** when prompted to confirm the deletion.

6 Selecting Source Tables and Managing Metadata

This section describes how to select source tables and manage metadata. The source tables are the tables that were replicated to the Landing Zone by the Replicate task (i.e. the target tables of the Replicate task).

In this chapter:

Selecting or Adding the Source Tables

This section explains how to select or add the source tables. Note that in the following explanations, "table" refers to the physical database object whereas "entity" refers to the virtual object within Compose for Data Lakes.

You can select the source tables using any of the following methods:

- Use Compose for Data Lakes to discover the Landing Zone
- Import entities from another project
- Create the entities manually in Compose for Data Lakes

Discovering the Landing Zone

To discover the source tables:

1. Open your project.
2. In the **Storage Zone** panel, select **Discover** from the drop-down menu in the top right corner.

   -OR-

In the **Manage Metadata** window, click the **Discover** toolbar button.

The **Discover** window opens.

3. Select the desired source Landing Zone and then click **OK**.

   The **Landing Tables/Views Selection -** ***Name*** window opens.

4. Choose one of the following **Search for** options:
   a. To search for tables only, select **Tables**.
   b. To search for views only, select **Views**.
   c. To search for tables and views, select **All**.
5. To include internal Qlik tables in the search results, select the **Show Internal Qlik Tables** check box. This may be useful for debugging, but is not usually not necessary.
6. If you do not want the primary keys in Hive to be the same as the primary keys in the source endpoint defined for the Replicate task, clear the **Include primary keys from the Replicate source endpoint** option.
7. To only search for tables/views whose names contain a specific string, type the string in the **Name** field.

   For example, entering "ers" will return "customers" and "suppliers" in the search results.

8. Click the **Search** button.

   The resulting tables/views will be displayed in the list in the left of the window.

9. Optionally, click the **Clear Cache** button to clear the Landing Zone's metadata cache (tables and columns). This may be necessary, for example, if tables was added to the Landing Zone or renamed. Such tables will not appear in the table list until the cache is cleared.
10. To add all of the resulting tables/views, click the **>>** button **(Add All)**

    > **Note** You can select multiple tables/views by holding down the [Shift] (sequential selection) or [Ctrl] (non-sequential selection) button.

11. To add specific tables/views, select the desired tables and/or views and then click the **> (Add)** button.

> **Note** If you add a table that already exists in the Metadata with the same name, then the new table is added with the name: ***source_table_name*_DISCOVERED** (or ***source_table_name*_DISCOVERED_02** if the name ***source_table_name*_DISCOVERED** already exists, and so on).
>
> If the table contains attribute domains that differ from existing ones but have the same name, they will also be appended with the ***_01*** suffix.

12. Click **OK** to add the selected tables/views to the project.

    The **Generating Metadata from [Metadata Name]** window opens.

    A progress bar indicates the current metadata generation progress. For each stage of the metadata generation process, a corresponding message appears in the **Messages** list.
13. After the metadata has been generated, click **Close**.
14. Repeat steps 2-12 to discover additional sources.

Clearing or Recreating the Metadata Cache

To improve performance when reading from the Landing Zone or from the Storage Zone, Compose for Data Lakes caches both the Landing Zone metadata and the Storage Zone metadata. However, synchronization issues may sometimes occur if the structure of the Landing Zone or the Storage Zone metadata is altered outside of the Compose for Data Lakes project.

If you aware of external changes to the metadata or if you notice any data synchronization anomalies, Compose enables you to recreate the metadata cache, as described in the following procedures.

Recreating the Landing Zone Metadata Cache

To refresh the Landing Zone cache on the next reading of the metadata, click **Clear Landing Cache** in the **Mappings** tab of the **Storage Zone** panel.

Recreating the Storage Zone Metadata Cache

To recreate the storage zone metadata cache in an Apache Hive or Databricks project:

1. In the Storage Zone panel, select the **Drop and Recreate|Storage Metadata Cache** item from the menu in the top right corner.

A confirmation dialog box opens warning you that this process could take a while depending on the number of tables involved.

2. Click **Yes** to recreate the storage zone metadata.
3. When the storage zone metadata cache is recreated successfully, click **Close** in the dialog box.

To recreate the storage zone metadata cache in an Apache Spark project:

1. In the Storage Zone panel, select the **File and Table Actions|Recreate Storage Metadata Cache** item from the menu in the top right corner.
2. Click **Yes** in the warning dialog box to recreate the storage zone metadata.
3. When the storage zone metadata cache is recreated successfully, click **Close** in the dialog box.

Importing Entities and Mappings from Another Project

You can import entities and mappings from another project with the same Storage Zone type. This can be useful within a development environment, for example, if you need to integrate a private developer's project with the main project.

To import entities and mappings

1. Open the **Manage Metadata** window as described in Managing the Metadata.
2. In the **Entities** toolbar, click the **Import from Project** button.
3. The **Import from Project** wizard opens.
4. In the **Entities** tab:
   - Select a project from the **Import from Project** drop-down list.
   - Optionally, search for specific entities.
   - Select which entities to import or select **Select All** to import all entities.
5. Click **Next** to select which mappings to import.

   To create new entities and mappings if the selected entities and mappings already exist, clear the **Replace existing entities and mappings** check box.

   The new entities/mappings will be named `<existing_name>_IMPORTED` (or `<existing_name>_IMPORTED_<n+>` if the entity/mapping is imported more than once).
6. In the **Mappings** tab:

   Either click **Finish** to import all mappings for the selected entities (the default).

   -OR-

Select which mappings you want to import and then click **Finish** to import the selected entities and mappings.

If you do not wish to import any mappings, clear the **Mappings** check box before clicking **Finish**.

Limitations

The Storage Zone needs to be "adjusted" when deleting an attribute from the Metadata and then adding the same attribute back to the Metadata. However, the "Adjust" operation will also delete the data from the corresponding Storage Zone column.

Validating the Metadata and Storage

Once the table metadata has been generated, to prevent data inconsistency issues, it is strongly recommended to check the validity of the metadata and the Storage Zone. For example, for the metadata to be valid, each of the tables must have a Business Key.

- Validating the metadata does not recalculate expressions for historical data that has changed.
- Validation of the Data Lake does not detect columns that have been renamed.

To validate the metadata:

1. Click the **Validate** button inside the Metadata screen (from the Storage Zone panel).

   Compose for Data Lakes will run validation checks and point to any entities which are not valid.

   If the metadata is valid, the following message will be displayed:

   `Validation tests completed successfully. No issues were detected`.

   If the metadata is not valid, the **Validating Storage Zone** window opens. This window is divided into the following columns:

   - **Severity:** Warning or Error.
   - **Message:** A message indicating why the entity is invalid.
   - **Names:** The names of the affected entities.
   - **Resolve:** To open the **Manage Metadata** window and manually resolve the issue, click the **Edit Entities** button.
2. Resolve the issue (for example, by adding a Business Key) and then click **Close**.

3. Click **Validate** again.

A message will confirm the metadata’s validity. Click **Close**.

To validate the Storage Zone:

1. Do the following:
   - **In Hive projects**, either click the **Validate** button in the bottom right of the **Storage Zone** panel, or select **Validate** from the drop-down menu in the top right of the **Storage Zone** panel.
   - **In Spark projects**, external tables are initially created by running **Create External Tables** from **File and Table Actions** in the **Storage Zone** menu. After the storage external tables are created, click **Validate External Tables** to validate them.
2. Compose for Data Lakes will run a series of validation checks and the **Validating Storage** window opens.

   If the storage metadata is not valid, the following message will be displayed:

   ```
   The metadata is not valid.
   ```

   (To resolve the invalid issues, see instructions above "**To validate the metadata**".)

   If the storage zone is valid, the following message will be displayed:

   ```
   The Storage Zone is valid.
   ```

   If the metadata is not the same as the Storage Zone metadata, the following message will be displayed:

   ```
   The Storage Zone is different from the metadata.
   ```

3. Review the report in the **Validating Storage** window. (Note that this step is only applicable if the Storage Zone is different from the metadata.)
   - In Spark projects, only ADD COLUMN and ADD TABLE are supported. For all supported changes, the **Adjust Automatically** button is displayed in the **Validating Storage** window. The window also displays the **Drop and Recreate Tables** button; clicking this button deletes all the files and recreates empty tables and directories as described in Dropping and Recreating Tables and Deleting Files.
   - In Databricks projects, only ADD COLUMN and ADD ENTITY are supported. For non-supported changes, you should click the **Drop and Recreate Tables** button each time validation is performed, and then reload your data with the history or manually apply the structure changes.

- In Hive projects, if all changes are supported, the **Adjust Automatically** button is available. If you make a change that is not supported, click **Generate Adjust Script** to generate a script with the adjust commands.

The **Adjust Automatically** button will be disabled either if the Generate DDL scripts but do not run them option is selected or if Compose for Data Lakes is unable to automatically adjust the Storage Zone (for example, in cases of data type changes that are not supported by the database or may result in data loss, or a change in an entity’s business key or distribution key). In such cases, you should click **Generate Adjust Script** as described below.

Clicking **Adjust Automatically**:

If you clicked **Adjust Automatically**, the **Adjust Storage Zone** progress window opens.

When the "The Storage Zone was adjusted successfully." message is displayed, close the window.

Clicking **Generate Adjust Script**:

When you click **Generate Adjust Script**, the **Generate DDL Scripts** window opens showing the progress of the script generation.

The generated scripts will be saved to:

<product_dir>\data\projects\<project_name>\ddl-scripts

Once the script(s) have been generated, close the **Generate DDL Scripts** window.

When working with an Apache Hive project, after you close the **Generate DDL Scripts** window, the **DDL Script Files** window opens automatically displaying the generated scripts. The DDL Script Files provides a read-only view that allows you to review the scripts and download them.

The scripts need to be executed directly in your Storage Zone. Make sure that any modifications that you make to the scripts are done prior to executing them.

> **Important:** When you run the adjust scripts, backup tables are created from the existing tables. The backup table names are appended with an "_old" suffix and must be deleted manually after the script completes.

> **Note** Search for "TODO" in the script to locate the part of the script that needs modifying.

4. Click **Close** to close the **Adjust Storage** window.

See also: Supported Characters.

Managing the Metadata

You can add, remove and edit the entities and attributes according to your needs. All management tasks are performed in the **Manage Metadata** window, which you can open using one of the following methods:

- Click the **Metadata** button at the bottom left of the **Storage Zone** panel.
- Click the **Entities** number in the **Storage Zone** panel.
- Select **Metadata** from the drop-down menu in the top right of the **Storage Zone** panel.

The **Manage Metadata** window is split into two tabs: The **Logical Metadata** tab and the **Physical Metadata** tab. The **Logical Metadata** tab shows the entities and attributes as they appear in the Metadata whereas the **Physical Metadata** tab provides a preview of the actual tables (and columns) that will be created in the Storage Zone.

In the **Logical Metadata** tab, you can perform various management tasks such as adding and/or editing entities and attributes. For more information, see Managing Entities, and Managing Attributes

In the **Physical Metadata** tab, you can add partitions to tables as described in Managing Partitions and view fields that Compose for Data Lakes automatically adds, such as **header__modified_batch**. Note that this option is only available when **Apache Hive** is selected as the project type.

Managing Partitions

Partition management is only available when **Apache Hive** is selected as the project type.

Hive organizes tables into partitions. It 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.

Compose for Data Lakes dynamically creates the Hive partitions, 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.

- When a large volume of data needs to be reloaded, it is advisable to create Replicate partitions of a manageable size so that Compose will be able to process them one (or a few) at a time. It is recommended that you first assess the volume of the reloaded data in your test environment before production.
- In order to reduce the size of the `attrep_cdc_partitions` table and speed up processing, Qlik Compose for Data Lakes recommends that you delete, compact, or archive partitions that are no longer required.

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.

To partition a table:

1. In the **Physical Metadata** tab, select a table from the **Tables** list on the left of the **Manage Metadata** window.
2. Select the **Partition Key** tab in the lower right side of the **Manage Metadata** window.
3. Click **Add Partition Key**.

   A row is added to the list.
4. Select a column from the drop-down list in the **Column** column.
5. Repeat the steps above to add more partitions.

   The order of the partition keys affects the directory hierarchy order.

To delete a table partition:

1. In the **Physical Metadata** tab, select the table containing the partition you wish to delete (from the **Tables** list on the left of the **Manage Metadata** window).
2. Select the **Partition Key** tab in the lower right side of the **Manage Metadata** window.
3. Select the unwanted partition and then click **Delete**.
4. Repeat the steps above to delete other partitions.

By default, the **Bucket Key** tab contains all of the table's primary key columns. Any primary key column that you add to the **Partition Key** list will be automatically removed from the **Bucket Key** tab (as the same column cannot be used both as a partition and as a bucket). Since the **Bucket Key** tab must contain at least one primary key column, adding *all* of the primary key columns to the **Partition Key** tab is not permitted. If you do so, an error will be generated during validation or during runtime if you skip the validation stage.

Managing Entities

You can add, edit and remove entities as described in the table below.

Reducing the window size also shortens the toolbar. If the toolbar is too short to contain all the buttons, the toolbar options will be displayed in the drop-down menu instead. The shorter the toolbar, the more options will appear in the drop-down menu.

To	Do This
Add an entity	1. Click the **New Entity** button in the **Entities** toolbar. 2. Provide a name and description (optional) for the entity and then click **OK**.
Edit an entity	1. Select the entity you want to edit and then select **Edit** from the drop-down menu in the **Entities** toolbar. 2. Edit the entity’s name and description (optional) and then click **OK**.
Remove an entity	1. Select the entity (or multiple entities) that you want to remove, and then select **Delete** from the drop-down menu in the **Entities** toolbar. 2. When prompted to confirm the deletion, click **Yes**.
Duplicate an entity	1. Select the entity you want to duplicate and then select **Duplicate** from the drop-down menu in the **Entities** toolbar. 2. Edit the entity’s name and description (optional) and then click **OK**. The duplicated entity is added to the Entities list.
Import entities from another project	See Importing Entities and Mappings from Another Project.

To	Do This
Include historical records	Select the check box in the **Save History** column to the right of the desired entity. Note that if you chose Apache Spark of Apache Hive with Historical Data Store as your project type, all of the **Save History** check boxes will be selected by default.

Managing Attributes

You can add, edit and remove attributes as required. All attributes in the Metadata belong to the Attributes Domain. When adding a new attribute, you can either select an existing attribute from the Attributes Domain or create a new Attributes Domain. Both of these options are described in the table below.

To add an attribute from the attributes domain:

1. Click the **New Attribute** button in the **Attributes** toolbar.

   The **New Attribute** window opens.

2. From the **Attribute domain** drop-down list, select the desired attribute.
3. To edit the selected attribute domain on-the-fly, click the edit button located after the **Attribute domain** drop-down list. This will open the **Edit - *AttributeDomainName*** window. Then, continue from Step 2 in Edit an attribute domain.
4. In the **Attribute name** field, optionally change the default instance name for the attribute domain.

   The name cannot contain any of the following forbidden (by Hive) characters:

   ```
   : ; . , ' "
   ```

   You can create multiple instances of a single Attribute Domain. This is especially useful if you want to use the same Attribute Domain across multiple tables, with each "instance" having its own unique name. This also allows you to edit the properties of each attribute without affecting the other attributes, even though all of the Attribute Domain instances share the same parent Attribute Domain. For example, if the Attribute Domain name is "ID", you could create one instance for it in the "Categories" entity named "CategoryID" and another instance in the "Employees" entity named "EmployeeID". If, however, you edit the parent Attribute Domain attribute, all instances of that attribute will be updated as well.

5. **Data type:** The data type of the Attribute Domain. This can only be edited by editing the Attribute Domain.
6. To add a prefix to the attribute name, enter the desired prefix in the **Prefix** field.

Adding a prefix to an attribute name allows you to add multiple instances of the same attribute domain. For example, the attribute "Employee" could become two different attributes: "ReportsTo_Employee" and "HiredBy_Employee".

7. To create an expression for the attribute, click the **fx** button located after the **Expression** field and then continue from Creating Expressions.

   This may be required, for example, if you need to consolidate data from multiple sources with the same table metadata. In this case, you would need to create an expression with a variable. For more information, see Managing Variables.

8. To make the attribute part of the target table's primary key, select **Primary Key**.

9. This option is only available when the Compose for Data Lakes project type is Apache Hive. To partition the target table according to the attribute's values, select **Partitionkey**. The attribute will be automatically added to the **Partition keys** tab at the bottom right of the **Physical Metadata** tab.

   For more information on partitioning, see Managing Partitions.

10. This option is only available when the Compose for Data Lakes project type is Apache Hive. From the **Location in partition key list** drop-down list, select whether the attribute should be **First** of **Last**.

11. Select whether to add the attribute to **All tables** or only to the **Current table**.

    When adding a column to all tables, best practice is to define a reusable transformation instead of a dedicated expression. This way, if you would like to edit it later, you would only need to edit it once for all attributes.

12. From the **Location in table** drop-down list, select where you want the attribute to be positioned in the target table.

    If you are editing an attribute that is being used in the Data Lake, the column will always be added to the end of the table, regardless of which position you choose. To change the position of the attribute according to your selection, you will need to drop and recreate the table as described in Dropping and Recreating Tables and Deleting Files.

13. Click **OK** to save your settings.

To create a new attribute domain and add it to the Metadata:

1. Click the **New Attribute** button in the **Attributes** toolbar.

   The **New Attribute** window opens.

2. Click the plus sign to the right of the **Attribute domain** drop-down list.

   The **New Attribute Domain** window opens.

a. Specify a **Name** for the attributes domain.

   The name cannot contain any of the following forbidden (by Hive) characters:

   ```
   : ; . , ' " :
   ```

b. From the **Type** drop-down list, select one of the available data types.

c. If the selected data type requires further configuration, additional fields will be displayed. For example, when Decimal is selected, the **Length** and **Scale** fields will be displayed. Set the values as desired.

d. Optionally, specify a **Description**.

e. Click **OK** to add the newly created attribute domain to the **Attribute domain** field and close the **New Attribute Domain** window.

3. Continue from Step 4 in Add an existing attribute domain above.

   You can also add new attribute domains via the **Manage Attribute Domains** window. For more information, see Managing the Attributes Domain

To edit an attribute:

Method 1:

1. Select the attribute you want to edit and then click the **Edit** button in the **Attributes** toolbar.

   The **Edit Attribute *Name*** window opens.

2. Edit the values as required and then click **OK**.

Method 2:

1. Double-click the attribute you want to edit.

   The values in the attribute row become editable.

2. Edit the values as required and then click the tick button at the end of the row.

To remove an attribute:

1. Select the attribute(s) you want to delete.
2. Click the **Delete** button in the **Attributes** toolbar.
3. When prompted to confirm the deletion, click **Yes**.

To change the attribute order:

- Select the attribute you want to move and use the **Move Up**/**Move to Top** and **Move Down** /**Move to Bottom** toolbar buttons to move the attribute

To manage the Attributes Domain:

- See Managing the Attributes Domain.

To create an expression for an attribute:

- See Add an attribute from the attributes domain or Edit an attribute above.

To export the attributes to a TSV file:

- Select an entity from the Entities list on the left of the **Manage Metadata** window and then select **Export to TSV** from the drop-down menu in the Attributes toolbar.

  Depending on your browser settings, you will either be prompted to download the **<entityname>_Attributes.tsv** file or it will be downloaded to your default Downloads location.

Managing the Attributes Domain

The Attributes Domain provides a list of all the attributes available in the Compose metadata, as well as their data type. You can add, edit and delete attributes according to your data warehousing needs. The Attributes Domain also allows you to see which entities each attribute belongs to, as a single attribute may be present in several entities.

To manage the Attributes Domain

1. From the drop-down menu in the top right of the **Storage Zone** panel, select **Manage Attributes Domain**.
2. Add, delete and edit attributes as describe in the table below.

To	Do This
Add an attributes domain	1. Click the **New Attributes Domain** toolbar button. The **New Attribute Domain** window opens. 2. In the **Name** field, specify a name for the attribute. The name cannot contain any of the following forbidden (by Hive) characters: `: ; . , ' " :` 3. From the **Type** drop-down list, select one of the available data types. 4. If the selected data type requires further configuration, additional fields will be displayed. For example, when `Decimal` is selected, the `Length` and `Scale` fields will be displayed. Set the values as desired. 5. Optionally specify a **Description**. 6. Click **OK** to add the attribute and close the **New Attribute Domain** window.
Edit an attribute domain	1. Select the desired attribute and then click the **Edit** toolbar button. The **Edit: *Name*** window opens. 2. Edit the attribute as described in steps 2-6 of Add an attributes domain above. Note that the **Edit: *Name*** window also contains a **Used in Entities** list. Knowing which entities the attribute is used in may affect the type of changes you make, as the planned changes may not be appropriate for all entities.
Remove an attribute	1. Select the attribute you want to delete and then click the **Delete** toolbar button. 2. When prompted to confirm the deletion, click **Yes**.

Creating Expressions

Compose for Data Lakes allows you to transform data using an expression either in Replicate or Compose for Data Lakes, according to your needs. The table below provides further information about creating transformations.

Where the Transformation is Created	Reasons to Create a Transformation There	When the Transformation is Applied
Replicate	- Filtering large amounts of data that is not needed for the Storage Zone (in the present or the future) - Obfuscation due to regulatory reasons or internal policies - Data type conversion (e.g. converting a source data type that is not supported on the Storage Zone platform)	Before the data reaches the landing area.
Metadata	- The default location if you are not sure where to put it - General business logic - Needed for several sources or several data marts	Between the Landing Zone and the Storage Zone.
Storage Zone	- Specific source preparation - Need to preserve the original unfiltered source information in Hadoop - Needed for merging several sources	Between the Landing Zone and the Storage Zone.

See also Defining Reusable Transformations.

The following topics describe the Expression Builder:

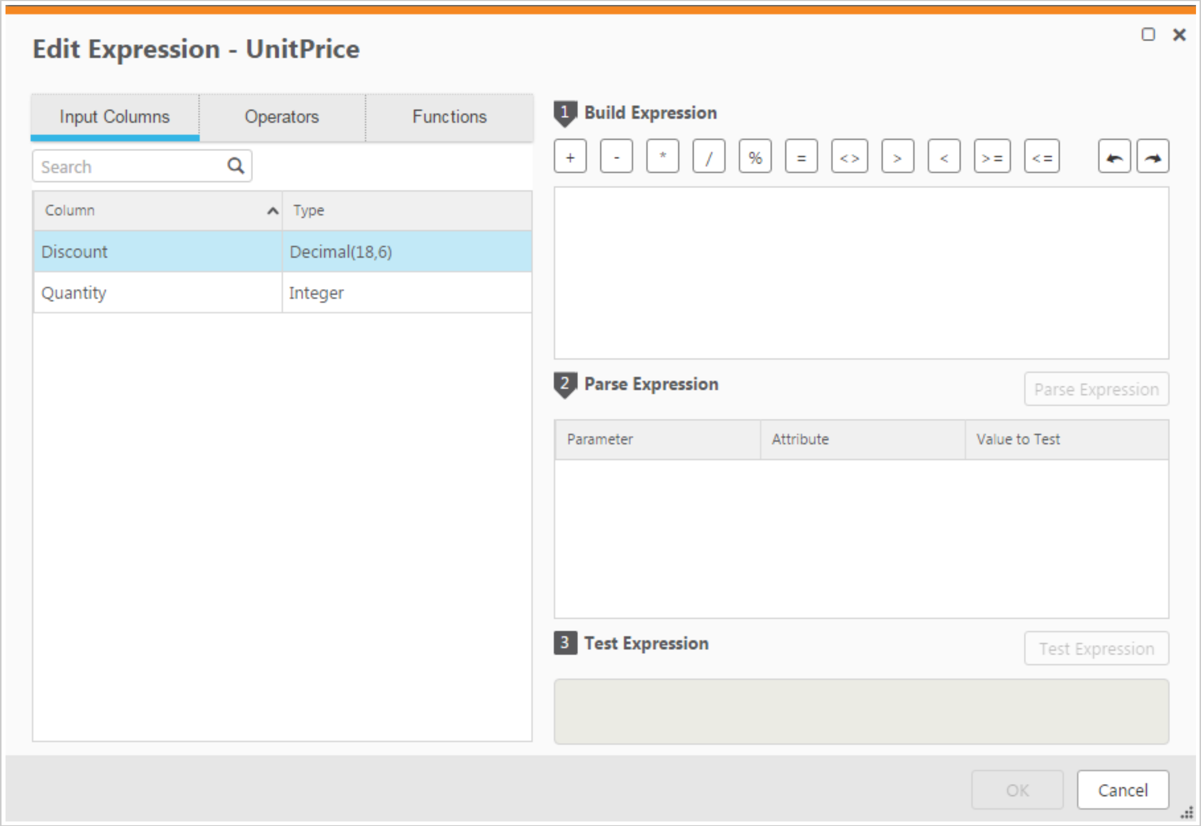
- Opening the Expression Builder
- Expression Builder Overview
- Building Expressions
- Testing Expressions

Opening the Expression Builder

The Expression Builder enables you to create a transformation without needing to type anything manually.

The Expression Builder can be opened in several places, depending on your needs. For more information about where to create a transformation, see the table Creating Expressions.

Figure 7.1 | Expression Builder



Column	Type
Discount	Decimal(18,6)
Quantity	Integer

Expression Builder Overview

The following section provides an overview of the Expression Builder functionality.

The Expression Builder consists of the following panels:

- **Tabs on the left of the Expression Builder**: These tabs contains elements that you can add to an expression. Select elements and add them to the **Build Expression** pane to create an expression. For more information, see Building Expressions.

  The following tabs are available:

  - **Parameters** - Only displayed when opening the Expression Builder from within the **Reusable Transformations > Edit Transformation** window.

    For information on reusable transformations, see Defining Reusable Transformations below.

  - **Input Columns/Input Attributes** - Columns/attributes that can be used to build your expression.

  - **Transformations** - Contains a list of reusable transformations. The tab is not displayed if no reusable transformations have been defined.

For information on reusable transformations, see Defining Reusable Transformations below.

- **Operators** - Operators that can be used to build your expression.
- **Functions** - Functions that can be used to build your expression.

> **Note** The Operators and Functions displayed in the Expression Builder use SQL format. As SQL support and implementation is different for each database type and version, the database being used in your Compose for Data Lakes project will determine which Operators and Functions will be available.
>
> Additionally, the list of Operators and Functions displayed in the Expression Builder is not comprehensive. However, you can use any Operators and Functions supported by the database, even if they are not included in the list.
>
> For an explanation of the available Operators and Functions, refer to the Help for your data lake.

- **Build Expression Pane**: The **Build Expression** pane is where you build your expression. You can add elements, such as columns or operators to the panel as well as type all or part of the expression. For more information, see Building Expressions.
- **Parse Expression Pane**: This pane displays the parameters for the expression. After you build the expression, click **Parse Parameters** to list the expression parameters. You can then edit the parameters, enter a value for each of the parameters and associate attributes with them. For more information, see Parsing Expressions.
- **Test Expression Pane:** This panel displays the results of a test that you can run after you provide values to each of the parameters in your expression. For more information, see Testing Expressions.

Building Expressions

The first step in using the Expression Builder is to build an expression in the **Build Expression** pane.

To build an expression:

1. Hover the mouse cursor over the element that you want to add to your expression (expressions usually start with an **Input Column**) and click the arrow that appears to its right.
2. Add **Operators** additional **Input Columns** and **Functions** as required.

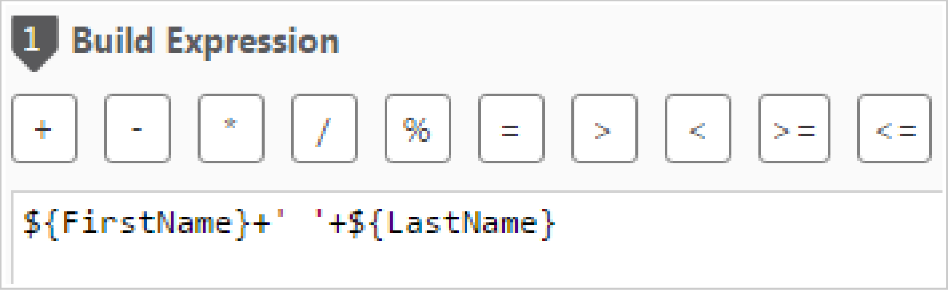
To add operators to your expression, you can use the **Operator** tab on the left or the **Operator** buttons located above the **Build Expression** pane or any combination of these.

Example:

To create an expression that combines the `FirstName` name and `LastName` columns, do the following:

1. Add the `FirstName` **Input Column** to the **Build Expression** pane.
2. In the **Operator** toolbar above the **Build Expression** pane, click the concatenate operator.
3. Then add a space between single quote characters and click the concatenate (`+`) operator again.
4. Add the `LastName` **Input Column** to the **Build Expression** pane.

   The expression would look like this:



```
${FirstName}+' '+${LastName}
```

Parsing Expressions

When you add operators to the expression, the expression’s parameters are usually added automatically to the **Parse Expression** pane. However, when you complete your expression or edit it, you may need to parse the expression see all of the parameters.

To parse the expression parameters:

- Click the **Parse Expression** button below the **Build Expression** pane.

If the expression is *not* valid, a red error message will appear at the bottom of the Expression Builder window.

If the expression is valid, the expression parameters and attributes (Input Columns) will be displayed in the in the **Parse Expression** pane. See the figure Test Expression.

Editing Parameter Names

By default, the parameter name is the same as the input column name. However, you can change the parameter name as needed and then associate it with an input column. This is useful, for instance, when you need to shorten attribute names. For example, `EstimatedTimeOfArrival` can be abbreviated to `ETA`.

To edit a parameter and associate it with an input column:

1. In the **Parse Expression** pane, edit the parameter name as required.
2. From the **Attribute** drop-down list, select the desired input column.

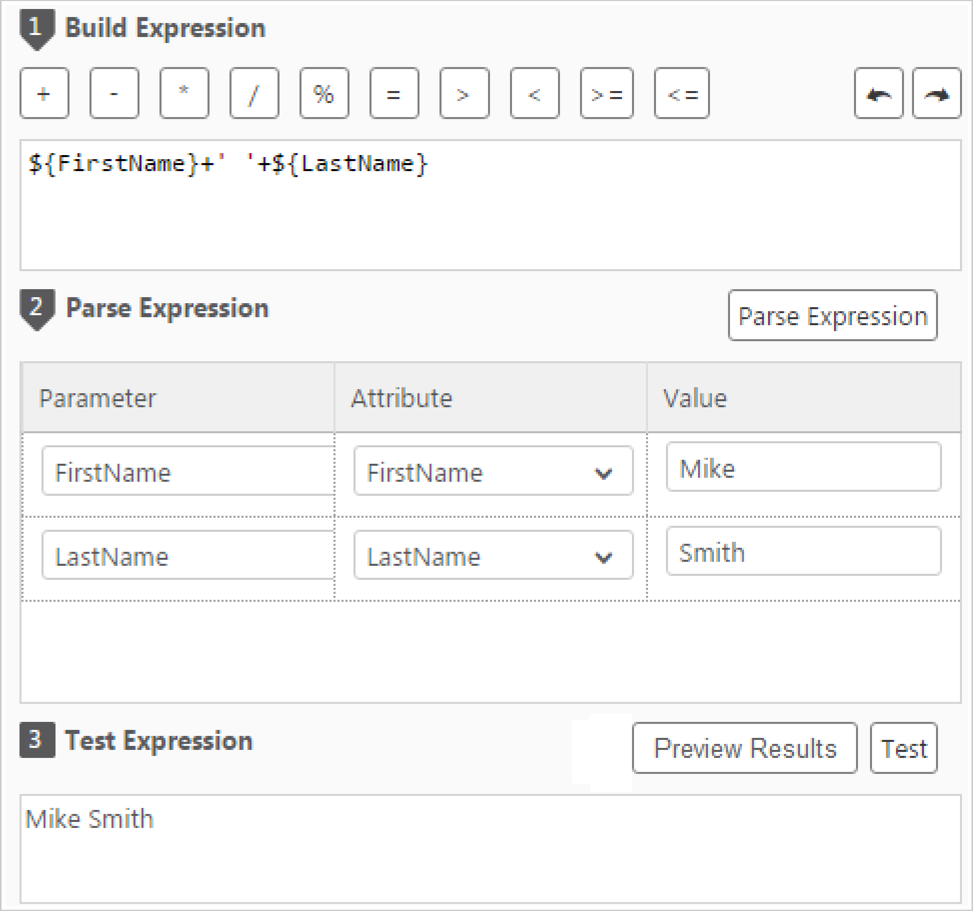
Testing Expressions

You test your expression to check that results are as expected. The following figure is an example of an expression that has been evaluated and tested.

Testing an expression that contains an analytic function will validate the syntax without actually executing the function. Additionally, the test will only be performed on a single record.

Compose does not check the data types of columns used in an expression for compatibility. For example, if a column of type integer is used in an expression for a column of type varchar, the expression will not be executed successfully.

Figure 7.2 | Test Expression



```
${FirstName}+' '+${LastName}
```

Parameter	Attribute	Value
FirstName	FirstName	Mike
LastName	LastName	Smith

Mike Smith

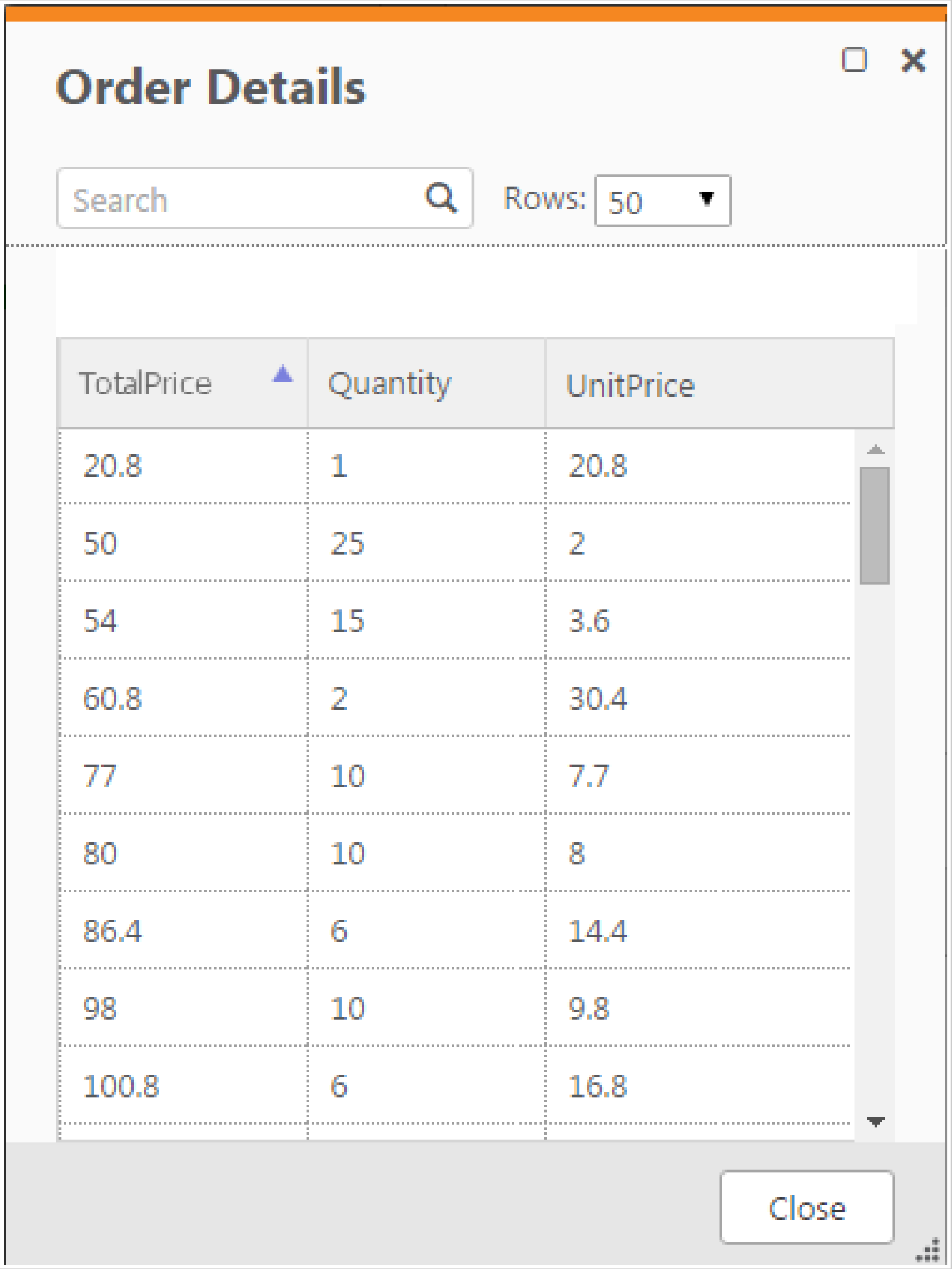
To test an expression:

1. In the Expression Builder window, build an expression as described in Building Expressions.
2. Click **Parse Expression** as described in Parsing Expressions.
3. View the parameters that are displayed. If your expression is not valid, an error message is displayed.
4. Optionally edit the parameters name(s) as described in Editing Parameter Names.
5. Type values for each parameter and then click **Test Expression** to see the expression result.

   For example, using the expression in Test Expression, type `Mike` for `FirstName` and `Smith` for `LastName`. The result displayed is `Mike Smith`.
6. This step is only available for transformations created in the **Edit Mappings** window. When you create a transformation in the **Edit Mappings** window, an additional button called **Show Data** appears to the left of the **Test Expression**

button. You can click this button to see how your expression translates into actual data.

For example, clicking the **Show Data** button for the expression `UnitPrice*Quantity` will open the following window.



Order Details

TotalPrice ▲	Quantity	UnitPrice
20.8	1	20.8
50	25	2
54	15	3.6
60.8	2	30.4
77	10	7.7
80	10	8
86.4	6	14.4
98	10	9.8
100.8	6	16.8

For more information on the **Edit Mappings** window, see Editing Column Mappings in Creating and Managing Storage Zone Tasks .

Defining Reusable Transformations

Note This feature is not available for Apache Spark projects.

In a single Compose project there may be several processes that require similar data transformations. For example a reusable transformation can be defined that concatenates first and last names. This transformation could then be used both in the Customers mapping and in the Employees mapping.

As opposed to stored functions or procedures which are environment dependent, reusable transformations are environment agnostic, meaning that not only can they be used as required within a Compose project, but they can also be used across different environments (using Compose for Data Lakes’s export/import function).

Centrally managed transformations increase efficiency by eliminating unnecessary duplication, while at the same time, enabling the seamless propagation of changes to all transformation instances.

To define a reusable transformation:

1. From the drop-down menu in the top right of the **Storage Zone** panel, select **Reusable Transformations.**

   The **Reusable Transformations** window opens.

   The window is split into the following panes:

   - Upper pane - Lists the reusable transformations that have been defined.
   - Lower pane - Provides additional information about transformation instances such as where they are in use (e.g. mappings, metadata, etc.) and the expression that was created using the transformation.

   Select a transformation to see the additional information.

2. Click the **New Transformation** toolbar button.

   The **New Transformation** window opens.

   a. In the **Name** field, specify a name for the transformation.

   b. In the **Category** field, specify a category name. If the category name already exists it will be displayed below the field when you start to type the name. To group the new transformation in the same category, simply select the existing name (unless of course you wish to create a new category with a similar name).

In the Expression Builder, transformations are grouped according to their category name, making it easier to find the transformation you want to use. Therefore, when specifying a category name, it is recommended to choose a name that reflects the purpose of the transformation. For example, if you create several transformations that concatenate data, it would make sense to group those transformations under a category called "Join".

c. To add a parameter to the transformation, click the **New** button to the right of the **Parameters** heading.

   A new row is added to the **Parameters** list.

d. Specify a name for the parameter, select an appropriate data type, and optionally provide a description.

   > **Note** If you add multiple parameters, you can change a parameter’s position by selecting the parameter and then using the Up/Down arrows (above the Parameters list) to reposition it.

e. Click the **Create Expression** button below the **Parameters** list.

   The **Edit Transformation** window opens.

f. In the **Edit Transformation** window, create an expression using the parameters you defined earlier.

   For information on creating expressions, see Creating Expressions.

g. Click **OK** to save the transformation.

   The transformation is added to the list in the upper pane.

Once a transformation has been defined, it will be available for selection as needed in the Expression Builder’s **Transformations** tab.

For information on creating expressions, see Creating Expressions.

Managing Reusable Transformations

You can manage reusable transformation as described in the table below.

To	Do This
Delete a transformation	Select the transformation and then click the **Delete** toolbar button. When prompted to confirm the action, click **OK**. If the transformation is in use, you first need to delete the transformation instances.
Edit a transformation	Double-click the transformation or select the transformation and then click the **Edit** toolbar button. Continue as described in Defining Reusable Transformations. **Note** Any changes you make to a transformation will be propagated to all instances of that transformation.
Edit a parameter	Open the **Edit Transformation** window as described in Defining Reusable Transformations. Then, select the parameter you want to delete and click the **Delete** button above the **Parameters** list.

7 Creating and Managing Storage Zone Tasks

Once the Metadata has been prepared, the next step in the Compose for Data Lakes workflow is to create the Storage Zone tables (optional), generate the task instructions and run the Storage Zone task. Tasks can either be run manually or scheduled to run periodically or in the future.

In this chapter:

Data Storage Tasks

This section, describes how to create the Storage Zone tables, generate the task instructions and run a Data Storage task.

It contains the following topics:

- Creating the Storage Zone Tables
- Generating the Task Instructions
- Controlling Data Storage Tasks

Creating the Storage Zone Tables

To create the Storage Zone tables:

1. Do one of the following depending on your project type:

   In an Apache Spark project:

   From the drop-down menu in the top right of the **Storage Zone** panel, select **File and Table Actions > Create External Tables**.

   In an Apache Hive project:

   Click the **Create** button in the bottom right of the **Storage Zone** panel.

   The **Creating Storage Zone** window opens.

A progress bar indicates the current progress. For each stage of the Storage Zone generation process, a corresponding message appears in the **Messages** list.

2. When the process completes, click **Close**.

See also: Supported Characters.

Generating the Task Instructions

After the Storage Zone tables have been created, you then need to generate the task instructions that will be used in the Storage Zone task. The task instructions include the Mappings between the Landing Zone tables and the Storage Zone tables. If you need to make changes to the Mappings, continue from Managing Task Definitions.

- Changing a Primary Key in the source record will cause a new record to be inserted in the storage table.
- Regenerating the task instructions after performing a non-supported change in the metadata will appear to succeed without errors or warnings, but the task will fail if run later.
- Defining a single task that ingests data from several Landing Zones is not supported. As a workaround, you can create a separate task for each Landing Zone.

To generate the Storage Zone task instructions:

1. Click the **Data Storage Tasks** button in the bottom left of the **Storage Zone** panel.

   The **Manage Data Storage Tasks** window opens.

2. If there are multiple tasks, in the left pane, select the desired task.
3. Click the **Generate** toolbar button, and select one of the following options:
   - **With validation** - This is the default option for generating storage tasks.

     You can also generate storage tasks with validation (default option) by clicking the **Generate** toolbar button.

   - **Without Validation** - Select this option if you are sure that the storage tables are adjusted properly and the mapping is valid. The generation of storage tasks is much faster. Note that you could have errors later when running the task if something is not valid.

   The **Generating Instructions for Task: *Name*** progress window opens. When the "Generate task finished successfully" message is displayed, close the window.

Only mappings associated with the task in the **Manage Tasks** window will be generated.

Controlling Data Storage Tasks

Once the Storage Zone tables have been created and the task instructions have been generated, you can then proceed to run the Storage Zone task. The Storage Zone task extracts data from the Landing Zone tables and loads it into the Storage Zone tables.

- A storage directory may be used exclusively by only one Compose for Data Lakes project.
- Data storage tasks are optimized to run on relatively large batches of data. It is recommended to specify a partition length in excess of one hour. Although specifying a partition length of less than one hour may improve latency, creating many partitions on the target may also impact (target) performance (especially in systems with large volumes of changes).
- Change Processing creates a new file on every write. This may cause many files to amass and degrade performance. Therefore, it is recommended to monitor the storage directory and periodically consolidate small files into larger ones and move/delete files that are no longer required.
- Storage directories and subdirectories are managed by Compose for Data Lakes; you should not delete files or write to them unless approved by Qlik Support or explicitly recommended in this guide.
- When storing data in an Apache Hive or Spark Compose project, for optimal performance, it is recommended to allocate a dedicated queue to Compose tasks only.
- In an Apache Hive project, in order to see the delta of data changes in the storage tables, you need to define the following commands so that Hive can read the subdirectories:

  ```
  set hive.supports.subdirectories=true;
  set hive.input.dir.recursive=true;
  ```

Storage Zone tasks can be run manually, scheduled to run periodically or run as part of a workflow. The section below describes how to run a Storage Zone task manually. For information on scheduling Storage Zone tasks or including them in a workflow, see Controlling and Monitoring Tasks and Workflows .

If there is a Replicate source table with data, that:

- Was not originally selected in the Replicate Full Load and Apply Changes task (i.e. was added later).

  -OR-

- Was selected in a Replicate Full Load and Apply Changes task, but was not selected in the mappings of the Compose for Data Lakes Full Load and Change Processing data storage tasks, and the tasks have already been run.

In any of the above scenarios, in order to get the data that was added later, you need to:

1. Duplicate the Compose for Data Lakes Full Load and Change Processing tasks associated with that table.
2. Run the duplicated Full Load task.
3. Run the duplicated Change Processing task.

Note that after running these tasks, duplicate records may exist in the Storage Zone, but they will be removed when any of the following processes occur:

- Loading to the Provisioning Zone
- Compacting the Storage Zone
- Reading from the Storage Zone views

To run a Storage Zone task:

1. Click the **Metadata** button in the bottom right of the **Storage Zone** panel.

   The **Manage Data Storage Tasks** window opens.

2. If multiple tasks have been defined, in the left pane, select the task that you want to run.
3. Click the **Run** toolbar button. The window switches to **Monitor** view and the following status bars are displayed:
   - **Completed** - Shows the tables that have already been loaded into Hive
   - **Loading** - Shows the tables currently being loaded into Hive
   - **Queued** - Shows the tables waiting to be loaded into Hive
   - **Error** - Shows the tables that could not be loaded into Hive due to error. Click the **Show Details** link below the bar to see more information about the statement(s) that resulted in the error.
   - **Canceled** - The number of canceled tables (tables that were not processed due to the task being aborted) does not appear as a separate status bar. To view the number of canceled tables, click the **Select All** link above the status bars.

To see more information about tables in a particular status, click the relevant status bar. A list of tables in the selected status will be shown.

When the **Completed** bar reaches 100% completed, close the **Manage Data Storage Tasks** window.

You can stop the task at any time by clicking the **Abort** toolbar button. This may be necessary if you need to urgently edit the task settings due to some unforeseen development. After editing the task settings, simply click the **Run** button again to restart the task.

You can also access the task log files by clicking the **View Log** button.

> **Note** Aborting a task may leave the Storage Zone tables in an inconsistent state. Consistency will be restored the next time the task is run.

With Apache Spark projects, you can click the **Spark History Server** button for information about completed Spark applications.

> **Note** If you encounter an access error when clicking the **Spark History Server** button, try one or both of the following:
>
> - Add an entry to the client host file that maps the Spark History Server host name to its externally accessible IP address.
> - Open the necessary firewall ports to allow Compose for Data Lakes to access the Spark History Server.

Reloading Data Storage from the Source Tables

This topic explains how to reload data from the Storage Tables in cases of inconsistencies or when metadata from the Source Tables is not replicated to the Landing Zone.

To reload data from the Source Tables:

1. Reload the Source Tables data and run the Qlik Replicate Full Load replication task again.
2. Create another Compose for Data Lakes Full load storage task.
3. Run the Compose for Data Lakes Full Load storage task.

Limitations

These limitations are not relevant to Apache Spark or Databricks projects.

- Running the Full Load data task again may result in duplication of records. Although Compose for Data Lakes excludes duplicates from the Storage Zone views so that they do not appear in any provisioned data, rerunning the Full Load uses considerable storage space. To minimize storage space, you should limit the Qlik Replicate Full Load to the minimum required time or run a script on the storage that merges the duplicated records with the new files.
- Records that should be deleted in the new Full Load task are not deleted nor marked for deletion (soft delete). To add records that are marked for deletion, you need to write a script that compares the new Full Load with the existing storage and add a soft delete record for every record that is missing from the new Full Load task.

Managing Task Definitions

Task definitions contain the mappings between the columns in the Landing Zone tables and the columns in the Storage Zone tables (and any transformations applied to those mappings). The same mappings can be used by several tasks. You can create new tasks, duplicate tasks and edit existing tasks as required.

The following options are available:

- Adding and Duplicating Tasks
- Editing Column Mappings
  - For each Compose for Data Lakes task, all of the mapping tables should be populated by data from one Replicate task.
  - You must regenerate the task instructions and then run a Storage Zone task whenever the mappings are modified. Populating the Storage Zone can either be done manually as described in Controlling Data Storage Tasks or automatically as described in Scheduling Tasks.
    If you have already run the data mart ETL tasks, then you also need to regenerate the data mart ETLs and run the tasks again as described in Managing Task Definitions.

Adding and Duplicating Tasks

As the default task definitions are generated automatically (by discovering the Landing Zone tables), there is usually no reason to manually create or duplicate an task. One possible reason to duplicate an task is if your Metadata contains different types of tables and you want to manage them in separate tasks.

The following task types are available:

- **Full Load** - Loads the selected tables from the Landing Zone into the Storage Zone.
- **Change Processing** - Updates the Storage Zone tables with the Landing Zone changes.
- **Full Load and Change Processing** - Loads the selected tables into the Storage Zone and then updates them with the Landing Zone changes.

To	Do This
Add a new Task	1. Click the **Manage** button at the bottom left of the **Storage Zone** panel. The **Manage Tasks** window opens. 2. Click the **New Task** toolbar button. The **Add New Task** window opens. 3. Specify a name for the task. 4. Optionally, specify a description. 5. Choose **Full Load** and/or **Change Processing** as the task type. 6. Click **OK**. 7. Select the task name in the left pane and continue from Editing Column Mappings.
Duplicate a Task	1. Click the **Manage** button at the bottom left of the **Storage Zone** panel. The **Manage Tasks** window opens. 2. Select the task you want to duplicate and then click the **Duplicate** toolbar button. The **Duplicate** window opens. 3. Specify a **Name** for the new Task. 4. Select a Landing Zone. 5. Optionally change the default **Schema**. 6. Select the **Task type** according to your Replicate task type. 7. Click **OK**. 8. Select the task name in the left pane and continue from Editing Column Mappings.

Editing Column Mappings

For improved metadata performance during discovery and mapping, Compose caches the metadata of the Landing Zones after reading them. However, synchronization issues may arise if the Landing Zone is modified outside of Compose for Data Lakes. In such cases, you should click **Clear Landing Cache** in the **Mappings** tab of the **Storage Zone** panel in order to refresh the cache on the next reading of the metadata.

For details on recreating the Storage Zone cache, see Clearing or Recreating the Metadata Cache.

The mappings show the current mapping between the Landing Zone tables and the Storage Zone tables. By default, the columns names and data in the Landing Zone tables and the Storage Zone tables will be identical. However, you can manually change the mappings according to your needs, either by simply mapping a Landing Zone column to a different Storage Zone column and/or by using an expression.

To edit column mappings:

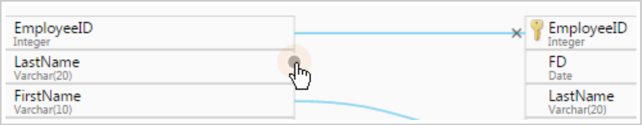
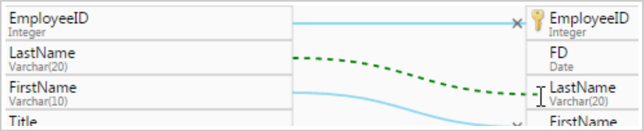
1. Click the **Manage** button in the **Storage Zone** panel.

   The **Mappings** tab is displayed. Each of the Storage Zone tables has a corresponding mapping name.

2. In the **Mappings** column, click the mapping that you want to edit.

   The **Edit Mapping: Name** window opens.

3. Edit the mapping as described in the table below.

To	Do This
Map a column in a Landing Zone table to a column in a Storage Zonetable	**Note** The mapping procedure differs depending on whether you are in Standard View or Compact View. For information on changing the view, see Change the view. **In Standard View:** 1. Hover the mouse cursor over the Landing Zone column name as shown in the image below. A gray dot appears to the right of the column name.  EmployeeID Integer LastName Varchar(20) FirstName Varchar(10) EmployeeID Integer FD Date LastName Varchar(20) 2. Drag the mouse cursor from the gray dot to the desired column in the Storage Zone table.  EmployeeID Integer LastName Varchar(20) FirstName Varchar(10) Title EmployeeID Integer FD Date LastName Varchar(20) FirstName 3. When the dotted line turns green (as shown below), release your mouse button. The mapping operation is completed. Note that if the dotted line turns red (instead of a green), you will not be able to map the Landing Zone column with the desired Storage Zone column. A red dotted line indicates that the Landing Zone and Storage Zone column data types are incompatible with each other. **In Compact View:** 1. Switch to Compact View as described in Change the view. 2. Drag the Landing Zone column to the cell located to the left of the target Storage Zone column.
Auto-generate mapping	Click the **Auto-Map** toolbar button.
Remove all mappings	Click the **Reset** toolbar button.

To	Do This
Change the view	Changing to a more compact view is recommended for Landing Zone tables that have numerous columns. In compact view, the table columns are organized in rows (instead of a single list), making it easier to locate Landing Zone columns and map them to the desired Storage Zone columns. **To change the view:** Click the **Change View** toolbar button. For information on creating mappings in Compact view, see Map a column in a Landing Zone table to a column in a Data Lake table.
Select a different source database	Select a database from the **Landing Area Database** drop-down list on the left of the window.
Select a different source schema	Select a schema from the **Schema** drop-down list on the left of the window.
Select a different table	Select a table from the **Table** drop down list on the left of the window.
Create a column-level transformation	1. Hover the mouse cursor over the Storage Zone Column for which you want to create a transformation and then click the **fx** button that appears to its right. The Expression Builder opens. 2. Continue from Opening the Expression Builder.

Adding and Deleting Mappings

You can add and delete mappings as required. For example, if you want one of the Storage Zone tables to contain columns from several tables in the Landing Zone, then you need to add a new mapping for each of the Landing Zone tables.

To add, delete, and rename mappings:

1. Click the **Manage** button in the **Storage Zone** panel.

   The **Manage Tasks** window opens.

2. In the left pane, select the desired Task.
3. Add or delete mappings as described in the following table.

To	Do This
Add a new mapping	1. In the **Data Lake Tables** column, select the table that you want to map. 2. Click the **New Mapping** button above the **Delivery Tables** column. The **New Mapping** window opens. 3. Optionally change the default mapping name. 4. From the **Entity** drop-down list, select the entity in the Storage Zone to which you want to map. 5. Click **OK** to save the mapping. 6. Enable the mapping.
Delete a mapping	1. In the **Mappings** column, hover the mouse cursor over the mapping you want to delete. 2. Click the **Delete(x)** button that appears to its right. 3. Click **OK** when prompted to confirm the deletion.
Rename a mapping	1. In the **Mappings** column, hover the mouse cursor over the mapping you want to rename. 2. Click the **Rename (A)** button that appears to its right. The **Rename** window opens. 3. Specify a new name for the mapping and then click **OK.**

Using Lookup Tables

Lookup tables are useful for replacing source data with the actual data that you want to appear in the Storage Zone. For example, a lookup table could be used to replace a zip code with a full address or, conversely, to replace a full address with a zip code.

Lookup on a column which is mapped to the Compose for Data Lakes "From__Date" column is not supported.

To link a lookup table column to a Storage Zone table column:

1. Click the link to the desired task in the **Storage Zone** panel.

   The **Manage Tasks** window opens.

2. In the **Mappings** column, click the mapping for the Storage Zone table containing the result column (with the data that you want to replace).

   The **Edit Mapping - *Name*** window opens.

3. Hover the mouse cursor over the relevant Storage Zone column and then click the **Lookup** button that appears to the right of the column name.

   The **Select Lookup Table** window opens.

   a. From the **Database** drop-down list, select the database containing the lookup table.

      The database must reside in your Landing Zone.

   b. From the **Schema** drop-down list, select the schema containing your source lookup tables.

   c. Select either **Table** or **View** according to the lookup table type.

   d. From the **Table** drop-down list, select the lookup table.

      The right side of the **Select Lookup Table** window displays the lookup table columns and their data types. To view the data in the lookup table, click the **Show Lookup Data** button.

   e. After you have selected the lookup table, click **OK**.

   The **Lookup Transformations** - ***Table Name*.*Column Name*** window opens.

   The window is divided into the following panes:

   Upper pane: The upper part of the right pane (**Condition**) displays the condition expression, which stipulates the condition(s) for performing the lookup.

   Lower pane: The lower part of the right pane (**Result Column**) displays the column result expression, which stipulates what data to replace in the target column.

4. To change the lookup table, click the **Change Lookup Table** button above the lookup table columns and then perform steps a. to d. above.

5. To view the lookup table or landing table data, click the **Show Lookup Data** or **Show Landing Data** buttons respectively.

6. To specify the condition(s) for performing the lookup, click the **Create Expression** button (which changes to **Edit Expression** after an expression has been created) above the **Condition** expression.

   The **Condition Expression - Column Name** window opens.

7. Create an expression using the landing and lookup table columns on the left.

   For an example, see Lookup Example.

For information on creating expressions, see Creating Expressions.

8. To specify what data to replace or add if the lookup conditions are met, click the **Create Expression** button (which changes to **Edit Expression** after an expression has been created) above the **Result Column** expression.

   The **Result Expression - Column Name** window opens.

9. Create an expression using the landing and lookup table columns on the left.

   For an example, see Lookup Example.

   For information on creating expressions, see Creating Expressions.

10. To preview the results, click the **Preview Results** button.
11. Click **OK** to save your settings and close the **Lookup Transformations** - **Table Name**.**Column Name** window.

Lookup Example

The following example shows how a lookup table is used to concatenate a Dutch translation of the category name (located in the lookup table) to the original category name located in the landing table.

The lookup could be defined using the following expressions:

1. Condition expression:

   ```
   ${Lookup.CategoryID}=${Landing.CategoryID}
   ```

   Meaning: Perform the lookup only if the Category ID in the landing table and the lookup table are the same.

2. Result column expression:

   ```
   ${Lookup.CategoryName} + ' is ' + ${Landing.CategoryName}
   ```

   Meaning: Add the data in the **CategoryName** column in the lookup table to the data in the **CategoryName** column in the landing table (separated by the word "is").

Assuming the result column name is "Split Name", clicking the **Preview Results** button would display the following table:

Split Name	Category Name (Lookup)	Category Name (Landing)	Category ID (Lookup)	Category ID (Landing)
dranken is Beverages	dranken	Beverages	1	1

Split Name	Category Name (Lookup)	Category Name (Landing)	Category ID (Lookup)	Category ID (Landing)
Specerijen is Condiments	Specerijen	Condiments	2	2
Gebak is Confectionary	Gebak	Confectionary	3	3
Zuivelproducten is Dairy Products	Zuivelproducten	Dairy Products	4	4
Grains/Granen is Grains/Cereal	Grains/Granen	Grains/Cereal	5	5
Vlees/Gevolgete is Meat/Poultry	Vlees/Gevolgete	Meat/Poultry	6	6

Dropping and Recreating Tables and Deleting Files

Compose for Data Lakes enable you to drop and recreate tables in your Storage Zone as required. The procedure for doing so differs depending on the project type (Apache Spark, Apache Hive, or Databricks). For Apache Spark projects you can also delete files.

When changing certain project settings (e.g. table prefixes) drop and create is required. If you change the Metadata after the Storage Zone tables and/or files were already created and loaded with data, you should adjust the Storage Zone to reflect the modified Metadata (as described in Validating the Metadata and Storage). Some changes however cannot be resolved by adjusting the Storage Zone. In such cases, you can either revert the Metadata to its pre-modified state or drop and (optionally) recreate the Storage Zone tables.

Note that dropping and recreating entities will delete *all* of the data in the tables and should only be performed in lieu of a better option.

- In some scenarios, you need to edit the CREATE table statements before they are run. This can be done using the Generate DDL scripts but do not run them in Editing the Project Settings. For example, if you want to override the default sorting of your Storage Zone tables or add specific formatting annotations, you will need to edit the script to accomplish this.
- The Change Processing context (i.e. the point in time when changes were last captured) is deleted when dropping all tables but preserved when dropping selected tables. Therefore after deleting selected tables, in order for Compose for Data

Lakes to continue processing changes from when the tables were dropped, you need to perform the following additional steps:

1. Duplicate the Full Load and Change Processing tasks.
2. Delete the old tasks (i.e. the tasks that were duplicated) making sure to select the **Delete mappings not used by other tasks** option.
3. Run the Full Load and Change Processing tasks again. This may result in duplicates in the Storage Zone tables, but the duplicates are excluded from the Storage Zone Views and will not appear in any provisioned data.

Dropping and Recreating Tables and/or Files in an Apache Spark Project

To drop and recreate entities in an Apache Spark project:

1. In the Storage Zone panel, select the **File and Table Actions|Drop and Recreate** item from the menu in the top right corner.

   The **Drop and Recreate** window opens.

2. Select which entities to drop and recreate.
3. From the options below the **Selected Entities** list, select one of the following:
   - **Files and External Tables** - to drop the selected entities and any related files
   - **Files** - to drop only files related to the selected entities
   - **External Tables** - to drop only the selected entities' tables
4. Click **OK** to perform the drop and/or recreate operation.

Dropping and Recreating Tables in an Apache Hive Project

To drop and recreate all tables:

1. In the Storage Zone panel, select the **Drop and Recreate|All Tables** item from the menu in the top right corner.
2. The **Drop and Recreate Tables** dialog box opens.
3. Select whether to drop and recreate the tables or to drop them only.
4. Click **OK** to perform the drop and/or recreate operation.

To drop and recreate specific tables:

1. In the Storage Zone panel, select the **Drop and Recreate|Specific Tables** item from the menu in the top right corner.
2. The **Drop and Recreate Tables** dialog box opens.

3. Select which tables to drop and recreate.
4. Click **OK** to perform the drop and recreate operation.

Viewing and Exporting Task Commands

You can view the Task Commands that were run during the Storage Zone task. You can also export the Task commands to a TSV file for reviewing and sharing.

To view the Task Commands:

Click the **Task Commands** toolbar button.

The **Task Commands - <*Source_Landing_Zone_Name*>** window opens in **List View**. The **Description** column provides a description of each operation that was performed on the target tables. Each operation had a different process number (displayed in the **Process Number** column). For additional details about an operation, double-click the operation.

-OR-

Click the **Item View** button and navigate through the commands using the navigation buttons at the bottom of the **Task Commands - <*Source_Landing_Zone_Name*>** window.

To jump to a specific command, type the command number in the **Go To** field at the bottom of the window and then press [Enter].

To export the Task Commands to a TSV file:

In **List View**, click the **Export to TSV File** button located to the left of the search field.

A file named "<name>_Task_Instructions.tsv" will be saved to your default "Downloads" location or you will be prompted to save it (according to your browser settings).

To hide non-SQL steps from the display:

Select the **Filter non-SQL steps** check box

Modifying Task Settings

For each task, you can modify the settings according to your needs.

To modify task settings:

1. In the **Manage Tasks** window, select a task in the left pane and then click **Settings**.

   The **Setting - <*Task_Name*>** window opens.

2. In the **General** tab, edit any of the following settings:

   Options marked with an asterisk are relevant for Compose for Data Lakes for Hive projects only.

   - ***Sqoop incremental import (load):** 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 Data Lake table(s).

     This option supports INSERT operations (i.e. new records) only.

     Currently, to perform **Sqoop incremental import (load)** on several Landing Zones in the same project, a separate task needs to be run for each of the Landing Zones.

     Mapping of the FROM__DATE/Last Modified column can either be created manually or automatically. If you only want "Sqoop incremental import (load)" to be performed on a few tables or if the source column names are different in each of the Landing Zone tables, you should create the mapping(s) manually. However, if you want "Sqoop incremental import (load)" to be performed on many tables and the source column name is the same in each table, best practice is to specify the column name in the **From__Date source column** field. This will save time, as the mappings will be created automatically when the task is generated.

     For information on creating mappings, see Adding and Deleting Mappings.

     This option is not relevant for "Operational Data Store" projects or for tables (in "Historical Data Store with ACID" and "Historical Data Store" projects) that were set *not* to save history.

     For information on enabling/disabling "Save History" for tables, see Managing Entities.

   - ***If no source column is mapped to the FROM__DATE column:** When working with tables that store history, you can insert the current date and time or the "Lowest Date" value in the corresponding target table's FROM_DATE column.

Use the current date and time as the FROM__DATE: Select this option to populate the FROM__DATE column with the date and time that the Full Load operation started.

Use the "Lowest Date" (in the project settings) as the FROM__DATE: Select this option to populate the FROM__DATE column with the "Lowest Date" value in the project settings.

These options are not relevant for "Operational Data Store" projects or for tables (in "Historical Data Store with ACID" and "Historical Data Store" projects) that were set *not* to save history.

For information on enabling/disabling "Save History" for tables, see Managing Entities.

- ***Default History Resolution:** Specifies at what resolution level the history timestamp is saved. If no mapping exists for the **From Date** (FD) column to the source, Qlik Compose for Data Lakes uses **Minutes**. At this level, the timestamp includes the hours and minutes of when the change occurred. If you select a value of **Days**, the FD timestamp includes only the date, not the hours and minutes. If a mapping exists, then the FD level comes from the source.
- **Component Logging Level:** Select the log level granularity, which can be any of the following:
    - **INFO** (default) - Logs informational messages that highlight the progress of the task at a coarse-grained level.
    - **DEBUG** - Logs fine-grained informational events that can be used to debug the task.
    - **TRACE** - Logs finer-grained informational events than the **DEBUG** level.

    Note that the log levels **DEBUG** and **TRACE** impact performance. You should only select them for troubleshooting if advised by Qlik Support.

3. In the **Advanced** tab, edit any of the following settings:
    - **Sequential Processing:** Select this option if you want all the Storage Zone processes to run sequentially, even if they can be run in parallel. This may be useful for debugging or profiling, but it may also affect performance.
    - **Maximum number of Compose Agent threads:** Enter the maximum number of threads that the Compose Agent is allowed to run for the task. The default number is 10.

- **JVM memory settings:** Edit the memory for the java virtual machine (JVM) if you experience performance issues. **Xms** is the minimum memory; **Xmx** is the maximum memory. The JVM starts running with the Xms value and can use up to the Xmx value.
- **Position in default workflow:** Select where you want the Storage Zone tasks to appear in the default workflow. For more information on workflows, see Workflows.
- **Limit number of change data partitions read**: In order to limit the number of partitions that the Compose Agent should process per task, select this check box and enter the maximum number of partitions that can be read. The default number is 10.

4. In the **Spark** tab (relevant for Compose for Data Lakes for Spark projects only), enter any Spark session parameters that you wish to use.

   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`

   Note For supported Amazon EMR versions, see Supported Hive Distributions.

   In addition to standard Spark parameters, you can also set the `attunity.compose.coalesce` parameter to determine Spark's coalesce value. When this parameter is omitted, Spark's default value will be used. Best practice is to set this parameter to a lowish value for small tables (e.g. up to 100MB) although it is also recommended to follow the Spark guidelines.

5. To save your changes, click **OK**.

8 Creating and Managing Provisioning Tasks

In a Compose for Data Lakes for Spark project, the final step in the project setup is to define a provisioning task. The provisioning task will move the data from the Storage Zone to the selected provisioned data zone.

In this chapter:

Limitations and Considerations

When running provisioning tasks, the following limitations apply:

- 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.
- Mapping from multiple sources is not supported, in both Spark and Hive projects.

Defining and Running Provisioning Tasks

A provisioning directory may be used by only one Compose for Data Lakes project; however, several provisioning tasks may write to it if they are writing to different tables.

To define and run a provisioning task

1. Click the **Provisioning Tasks** button in the bottom left of the **Provisioning Zone** panel.

   The **Manage Provisioning Tasks** window opens.
2. Click **New Task**.

   The **New Provisioning Task** wizard opens, displaying the **General** screen.

3. In the **Name** field, provide a name for your task.
4. Optionally, in the **Description** field, provide a name for your task.
    a. Select one of the following task types:
        - With task types that incrementally update the target, temporary tables named **tmp_<table name>** are created during the task and deleted when the task completes. Note that any tables with the same name that already exist on the target will be replaced with the Compose for Data Lakes temporary tables during the task.
        - For all task types except **Snapshot**, both tables and views will be created in the provisioning target. The current view represents the most up-to-date data.
        - **Snapshot** - Provisions records matching the current or specified date and time.

            When **Now** is selected, only the most recently updated records in the Storage Zone will be provisioned. The frequency with which records are updated in the Storage Zone depends on how often (if at all) the Replicate task updates the Change Tables as well as how often the Compose for Data Lakes Data Storage task is run.

            To add the Compose for Data Lakes "From Date" column to the provisioned tables, select the **Add "header__from_date" column** check box.

            To create a snapshot identical to the source tables without any additional columns, deselect the **Add "header_modified_batch" column** check box. By default, this check box is selected.
        - **HDS** - Provisions both historical records and recently updated records.

            To create an HDS task type identical to the source tables without any additional columns, deselect the **Add "header_modified_batch" column** check box.
        - **Incrementally Updated ODS** - Creates an operational data store that contains the current data and applies incremental updates.

            With this task type:
            - The **Target type** and **File format** fields in the **Target** screen will be set to "Hive" and "ORC" respectively (and cannot be changed).
        - **Incrementally Updated HDS** - Creates a subset of the storage which contains the selected entities and their history and applies incremental updates.

5. Click **Next**.

   The **Tables** screen is displayed.

6. Using the arrows, select which tables you want to provision. Optionally, use the **Search** field(s) to find tables.

   Selected tables will appear in the **Selected Tables** list on the right.

   By clicking **Rename Options** you can specify a prefix and/or suffix that will be used for all the tables in the provisioning task. This will enable the same database to be reused for other provisioning tasks.
   To prevent provisioning task name conflicts, make sure that the prefix and/or suffix is different for each provisioning task.

7. Click **Next**.

   The **Target** screen is displayed.

8. Set the target connection options as described in the following table:

Field	Description
Target type	Choose one of the following locations according to where you want the provisioned data files to be transferred: » Hive » For optimal performance, it is recommended to allocate a dedicated queue to Compose tasks only. » To prevent table name conflicts, the Landing Zone, Storage Zone, and Provisioning Zone databases should be different. » Google BigQuery - Only available for "Google" or HDFS projects. When the task is run, Compose for Data Lakes creates BigQuery managed tables and populates them. When Google BigQuery is selected as your target type, enter the following parameters: » PROJECT_ID: the ID of your project » You must make sure that your project has the required permissions to allow Compose for Data Lakes to access it. » If the PROJECT_ID parameter has no value, the storage project ID will be used. » DATASET: a dataset in your project » Dataset Location: a string which holds values of locations for the dataset. The current default is "US". **Google BigQuery Limitations** » **Reset Project** and **Delete Task** do not delete the tables if data deletion was marked by the user. » **Test Connection** is disabled. » HDFS 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. » Amazon S3 - Only available when Amazon S3 is also set as the Storage Zone data store. When Amazon S3 is your selected target type, specify the the **Bucket name** of the target bucket. » Azure Data Lake Storage Gen1 - When selected as your target type, specify the **ADLS** (Data Lake Store) **URL** to which you want

Field	Description
	the files to be provisioned. Example: `adl://<data_lake_store_name>.azuredatalakestore.net/dir/file` » Azure Data Lake Storage Gen2 - When selected as your target type, specify the **Storage account**, **File system** and **Target folder** of the ADLS storage. » Google Cloud Storage - When Google Cloud Storage is your selected target type, specify the **Bucket name** of the target bucket.
Target folder	This field is not available when Hive is the provisioning target. Specify the target folder under which the Full Load and Change Processing subfolders will be created. The subfolders will be named according to the task name. If the target folder does not exist, Compose for Data Lakes will create it for you. Incremental Provisioning creates a new file on every write. This may cause many files to amass and degrade performance. Therefore, it is recommended to monitor the provisioning target directory and periodically consolidate small files into larger ones and move/delete files that are no longer required.
Database name	Specify the database name. This must be the same as the Hadoop target database defined in the Qlik Replicate task but *different* from the database specified in the Storage Zone connection settings. For more information, see Defining a Qlik Replicate Task. To prevent table name conflicts, the Landing Zone, Storage Zone, and Provisioning Zone databases should all be different.
Create Hive external tables	This field is not available when Hive is the provisioning target. Optionally, select this check box to create Hive tables in addition to the data files.

Field	Description
File format	Choose one of the following file formats for the provisioned files: » ORC » Avro » Parquet When using Cloudera and Spark provisioning with AVRO file format, the use of external Hive tables is not supported. If you are not using external Hive tables, the AVRO file format is supported with Cloudera.

9. Optionally (recommended), click **Test Connection** to verify that the information you entered is correct and then click **Finish**.
10. Click **Run** to run the task.

See also: Supported Characters.

Modifying Task Settings

For each task, you can modify the settings according to your needs.

To modify task settings:

1. In the **Manage Provisioning Tasks** window, select a task and then click **Settings**.

   The **Setting - <*Task_Name*>** window opens.
2. In the **General** tab, you can change the **Component Logging Level**.
   - Select the log level granularity, which can be any of the following:
     - **INFO** (default) - Logs informational messages that highlight the progress of the task at a coarse-grained level.
     - **DEBUG** - Logs fine-grained informational events that can be used to debug the task.
     - **TRACE** - Logs finer-grained informational events than the **DEBUG** level.

       Note that the log levels **DEBUG** and **TRACE** impact performance. You should only select them for troubleshooting if advised by Qlik Support.
3. In the **Spark** tab (relevant for Compose for Data Lakes for Spark projects only), enter any Spark session parameters that you wish to use.

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`

Note For supported Amazon EMR versions, see Supported Hive Distributions.

In addition to standard Spark parameters, you can also set the `attunity.compose.coalesce` parameter to determine Spark's coalesce value. When this parameter is omitted, Spark's default value will be used. Best practice is to set this parameter to a lowish value for small tables (e.g. up to 100MB) although it is also recommended to follow the Spark guidelines.

4. To save your changes, click **OK**.

Managing Provisioning Tasks

All management operations are performed in the **Manage Provisioning Tasks** window, which can be opened by clicking the **Provisioning Tasks** button at the bottom of the **Provisioning Zone** panel.

To	Do This
Delete a task	Select the desired task and then click **Delete|Delete Task**. When prompted for confirmation, click **Yes**.

To	Do This
Delete the provisioned data and metadata	This operation is only relevant for incremental task types and can be performed using the UI or using the CLI. Note that this action cannot be undone. Periodically deleting the data and metadata ensures optimal performance by deleting the small files (which may be numerous) that Compose for Data Lakes creates on the target when applying changes. Note that the actual number of files will depend on how often the provisioning task runs. **Using the UI:** Select the desired task and then click **Delete|Delete Data and Metadata**. When prompted for confirmation, type "confirm" and then click **Yes**. **Using the CLI:** Change the working directory to **<Product_Dir>\bin** and then run the following command: `ComposeCli.exe drop_provision -p` *`project_name`* `-n` *`task_name`* Running a CLI command can be useful if you wish to delete the task files and tables using external schedulers such as HP OpenView or Control-M. Before you can perform this operation, you must first run the `Connect` command as described in Connecting to the Qlik Compose for Data Lakes Server.
Edit a task	Select the desired task and then click **Edit**. Continue from Defining and Running Provisioning Tasks.
Search for a task	Enter a search string in the **Search** box. Only tasks with properties matching the search string will be shown.

9 Creating and Managing Command Tasks

Command tasks enable you to incorporate custom processes into your Compose for Data Lakes workflow. This is especially useful if you need to leverage external tools to transfer files, validate data, and so on. A Command task can run any script or executable supported by the operating system including batch files, Python scripts, PowerShell scripts, executables and so on.

In this chapter:

Defining Command Tasks

This section explains how to define a command task. You can define as many command tasks as you need and execute them at different stages of a Compose for Data Lakes workflow.

For security reasons, before you define a command task, make sure that the executable or script file that you want to run resides in the following directory on the Compose for Data Lakes server machine:

PRODUCT_DIR\data\projects\YOUR_PROJECT\scripts

To define a command task:

1. From the project drop-down menu, select **Manage Command Tasks**.

   The **Manage Command Tasks** window opens.
2. Provide a name for the task.
3. Optionally, enter a description.
4. In the **Script/Executable File** field, specify the name of the files that you want to run.
5. In the **Parameters** field, specify any parameters required by the command. Parameters should be separated by a space.

6. The user context is the user account under which the Task will run. To change the current user context, provide the **User**, **Password** and **Domain** of the account under which you want the Task to run.
7. Click **Save** to save your changes or **Discard** to discard any unsaved changes.

   The task will be added to the list of tasks in the left of the window.

Managing Command Tasks

The table below describes the task management options.

To	Do This
Edit a task	Select the task in the tasks list in the left of the **Manage Command Tasks** window and edit it as described in Defining Command Tasks.
Delete a task	Select the task in the tasks list in the left of the **Manage Command Tasks** window and then click the **Delete** toolbar button. When prompted to confirm the deletion, click **OK**.
Search for a task	Enter part of the task name in the search box above the task list. The list of tasks will be filtered to show only tasks that include the search term in their name.

Controlling and Monitoring Command Tasks

Command Tasks can be run from the **Manage Command Tasks** window or from the main Compose for Data Lakes Monitor view. Although they can be run individually, command tasks are usually run as part of a workflow.

For information on defining workflows, controlling and monitoring tasks, and controlling and monitoring workflows, see Controlling and Monitoring Tasks and Workflows .

To run a command task from the Manage Command Tasks window:

1. Open the **Manage Command Tasks** window and select the task you want to run.
2. Click the **Run** toolbar button.
3. The **Manage Command Tasks** window switches to **Monitor** view.

   In Monitor view the following information is available:

   - The task ID
   - The current status

- When the task started and ended
- The overall task progress

10 Controlling and Monitoring Tasks and Workflows

The Compose for Data Lakes monitor shows the current status of all your tasks and enables you to drill-down for additional information about each task. Task instances can be run immediately or scheduled to run in the future (either once or at set intervals).

In this chapter:

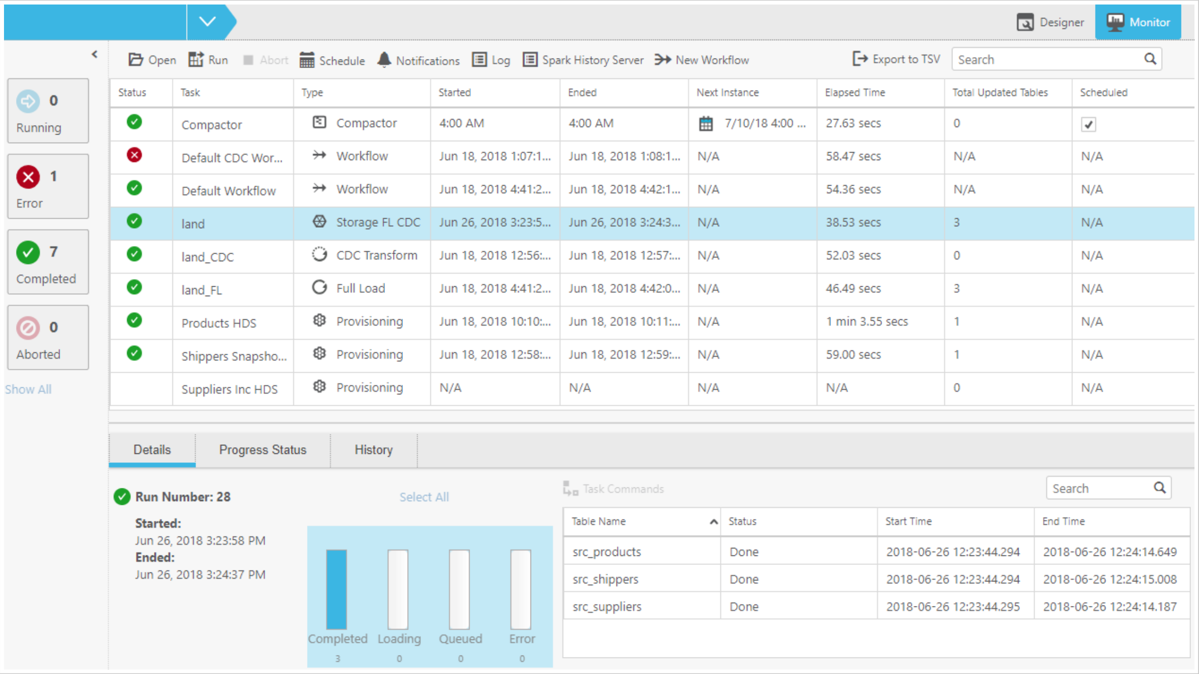
Viewing Information in the Monitor

As well as providing a high-level summary of all your tasks, the monitor also lets you view more detailed information about specific tasks.

To switch to monitor view:

1. Open a Compose for Data Lakes project and click the **Monitor** icon in the top right of the console.

   A list of tasks is displayed for the current project. The left pane of the monitor allows you to filter the task list by status as well as indicating the current number of running, failed and completed tasks.



Status	Task	Type	Started	Ended	Next Instance	Elapsed Time	Total Updated Tables	Scheduled
✓	Compactor	Compactor	4:00 AM	4:00 AM	7/10/18 4:00 ...	27.63 secs	0	☑
✗	Default CDC Wor...	Workflow	Jun 18, 2018 1:07:1...	Jun 18, 2018 1:08:1...	N/A	58.47 secs	N/A	N/A
✓	Default Workflow	Workflow	Jun 18, 2018 4:41:2...	Jun 18, 2018 4:42:1...	N/A	54.36 secs	N/A	N/A
✓	land	Storage FL CDC	Jun 26, 2018 3:23:5...	Jun 26, 2018 3:24:3...	N/A	38.53 secs	3	N/A
✓	land_CDC	CDC Transform	Jun 18, 2018 12:56:...	Jun 18, 2018 12:57:...	N/A	52.03 secs	0	N/A
✓	land_FL	Full Load	Jun 18, 2018 4:41:2...	Jun 18, 2018 4:42:0...	N/A	46.49 secs	3	N/A
✓	Products HDS	Provisioning	Jun 18, 2018 10:10:...	Jun 18, 2018 10:11:...	N/A	1 min 3.55 secs	1	N/A
✓	Shippers Snapsho...	Provisioning	Jun 18, 2018 12:58:...	Jun 18, 2018 12:59:...	N/A	59.00 secs	1	N/A
	Suppliers Inc HDS	Provisioning	N/A	N/A	N/A	N/A	0	N/A

Run Number: 28

Started:
Jun 26, 2018 3:23:58 PM
Ended:
Jun 26, 2018 3:24:37 PM

Table Name	Status	Start Time	End Time
src_products	Done	2018-06-26 12:23:44.294	2018-06-26 12:24:14.649
src_shippers	Done	2018-06-26 12:23:44.294	2018-06-26 12:24:15.008
src_suppliers	Done	2018-06-26 12:23:44.295	2018-06-26 12:24:14.187

For each task, the monitor displays the following information:

- **Status** - Running, Completed, Failed or Aborted.
- **Task** - The task name.
- **Type: The following task types are available:**

  Note that types marked with an asterisk are relevant for Spark projects only.

  - Full Load - Moves the data in its entirety from the Landing Zone to the Storage Zone.
  - *Provisioning - Moves the data from the Storage Zone to the Provisioning Zone.

  - *Compactor - Merges the changes (i.e. history) from the **delta_hds** folder with the **hds** folder and then deletes the **delta_hds** folder. This is a system task that should be run manually or scheduled to run as required.
  - CDC Transform - Moves changes to the data from the Landing Zone to the Storage Zone.
  - Workflow - Executes several tasks in succession. See also Creating Workflows.
  - Command - For information about Command Tasks, see Creating and Managing Command Tasks
  - Replicate - The Qlik Replicate task that moves the data from the source database to the Landing Zone.
- **Started and Ended** - The date and time the task started and completed (according to the server time). If the task is running, the **Ended** column will display the current progress. In the case of a Replicate task performing Change Processing, **Running CDC** will be displayed.
- **Next Instance** - The next time the task is due to run (if the task is scheduled).
- **Elapsed Time** - The time it took for the task to complete or - if the task is still running - how long the task has been running.
- **Total Updated Tables** - The number of tables updated in the Storage Zone or in the Provisioning Zone.
- **Scheduled** - Whether the task has been scheduled. "N/A" indicates that the task has never been scheduled whereas a check box indicates that the task has been scheduled. Clear the check box to disable the scheduling.

2. To view additional information about a task, select the task. The information is displayed in the following tabs in the lower pane of the monitor:

   - **Details** - The **Details** tab shows the following status bars:
     - **Completed** - Shows the tables that have already been loaded into Hive.
     - **Loading** - Shows the tables currently being loaded into Hive.
     - **Queued** - Shows the tables waiting to be loaded into Hive.
     - **Error** - Shows the tables that could not be loaded into Hive due to error. Click the **Show Details** link below the bar to see more information about the statement(s) that resulted in the error.

   To see more information about tables in a particular status, click the status bar. A list of tables in the selected status will be shown.

   You can also click the Task Commands button for more information about the operations performed during the task.

- **Progress Status** - The **Progress Status** tab shows the task's current progress as well as the sub-status (Waiting/Standby, Running, Failed, etc.) of operations within the task. To see details about a specific operations, click the number to the right of the operation status.

  For example, to view more information about an operation with an error status, click the number to the right of the **Failed** bar.

  With Apache Spark projects, you can click the **Spark History Server** button for information about completed Spark applications.

  > **Note** If you encounter an access error when clicking the **Spark History Server** button, try one or both of the following:
  >
  > - Add an entry to the client host file that maps the Spark History Server host name to its externally accessible IP address.
  > - Open the necessary firewall ports to allow Compose for Data Lakes to access the Spark History Server.

- **History** - The **History** tab provides a list of previous task instances.

  To view a task instance's log file, select the task and click the **View Log** button.

  To view more details about a task instance, either double-click the instance or select the instance and then click the **View Instance Details** button. The Details tab is shown.

3. To run a job immediately, select the task and then click the **Run** toolbar button.
4. To view and manage a task's settings, select the task and then click the **Open** toolbar button. For more information about the settings, see the relevant chapter in this guide.

Controlling Tasks

You can run and stop tasks/workflow manually or you can schedule them using the scheduling options described in Scheduling Tasks.

Running and Aborting Tasks Manually

You can run tasks manually and abort them if required.

To run a task manually:

- Select the task and click the **Run** toolbar button.

To abort a task:

- Select the task and click the **Abort** toolbar button.

  The task process is aborted. Note that aborting a task may leave the Storage Zone tables in an inconsistent state. Consistency will be restored the next time the task is run.

Scheduling Tasks

Scheduling tasks is a convenient way of continually updating the Storage Zone.

Note that as Compose currently does not provide a task-chaining option (i.e. run another task as soon as the current task completes), it may be better to schedule tasks using an external tool that supports this capability.

You can also use the command line interface (CLI) to run a task. For details, see Running Tasks using the CLI.

To schedule a task:

1. Click the **Schedule** toolbar button.
2. In the **<Name> Scheduler** window, choose one of the following options from the **Run Job** drop-down list.
   - **Once** - to run the job once on a specific date and time.
   - **Every** - to run the job at set intervals.
   - **Daily** - to run the job every day at a specific time.
   - **Weekly** - to run the job on selected days at a specific time
   - **Monthly** - to run the job on the *n*th of every month at a specific time
   - **Advanced** - to use a Cron expression. For a description of allowed cron formats together with usage examples, see Cron Format and Examples .
3. Set the scheduling parameters according to the selected scheduling option.
4. Click **OK** to save your settings.

   The date and time the next instance is scheduled to run will appear in the **Next Instance** column.
5. To disable a scheduled job, select the task and click the **Edit Scheduling** toolbar button. Then, select the **Disable** check box in the **<Name> Scheduler** window.
6. To cancel a scheduled job for a task, select the task and click the **Edit Scheduling** toolbar button. Then, in the **<Name> Scheduler** window, click **Delete**.

Running Tasks using the CLI

You can also run tasks using the CLI. This is especially useful if you wish to run Compose tasks from external schedulers such as HP OpenView or Control-M. Before you can run a task, you must first run the `Connect` command as described in Connecting to the Qlik Compose for Data Lakes Server.

As Compose for Data Lakes CLI requires Administrator permission, make sure to select "Run as administrator" when opening the command prompt.

The `RunTask` command populates the Storage Zone with data. The task can also be run using the **Run** toolbar button located in **Monitor** view as well as in the **Manage Task** window.

When this command succeeds, it returns `0`.

Syntax:

```
ComposeCli.exe run_task --project [project-name] --type
[storage|provisioning|workflow|compactor] --task [task-name] --wait
[timeout-in-sec]
```

where:

- **`project`** is the name of your Compose for Data Lakes project
- **`type`** is the type of task you want to run. These can be any of the following (tasks marked with an asterisk are relevant to Spark projects only):
  - `storage` - Data Storage task
  - `workflow` - Workflow task
  - `provisioning` - Provisioning task
  - `compactor` - Compactor task
- **`task`** is the name of the task you want to run.
- **`wait`** is the wait time specified in seconds.

The command line can run in sync or async mode. A value of `0` (seconds) indicates sync mode. This means that as soon as the task finishes, the command line returns to prompt. The default mode is async, with a value of -1. This is also applied if you leave this parameter empty. Other negative values are not permitted.

Note that if **`wait`** is excluded from the command, the task may appear to complete successfully even if it encountered an error.

Example:

```
ComposeCli.exe run_task --project nw --task hive_analytics --type DW
```

```
Compose Control Program started...

Instance Number: 9

Compose Control Program completed successfully.
```

Defining Notifications Rules

You can select events, on the occurrence of which, a notification will be sent to the specified recipients.

Notifications will not be sent unless the mail server settings are correctly defined.

To set a notification rule:

1. Switch to **Monitor** view.
2. Click the **Notifications** toolbar button.

   The **Notification Rules** window opens.
3. Click the **New** toolbar button.

   The **New Notification** wizard opens.
4. In the **Events** screen:
   - Specify a name for the notification
   - Choose for which type of events you want the notification to be sent, both at the task level and at the workflow level.
5. Click **Next**. In the **Recipients** screen:
   - Select **Windows Event** to send the notification to Windows Event Viewer and/or **Recipients** to send the notification to a list of email recipients.

     See also: Event IDs in Windows Event Log.
   - If you selected **Recipients**, enter the recipient email addresses in the **To**, **Cc** (optional) and **Bcc** (optional) boxes. Multiple addresses must be separated by a semi-colon.
6. Click **Next**. In the **Message** screen, optionally, edit the default notification message. You can add variables to the message by selecting the variable on the right and then clicking the arrow to the left of the variables list.

   The following variables are available:

Variable	Description
${PROJECT}	The name of the Compose project in which the event occurred.
${TASK_NAME}	The name of the task in which the event occurred.
${INSERTED}	The number of rows inserted in the Storage Zone.
${UPDATED}	The number of rows updated in the Storage Zone.
${DELETED}	The number of rows deleted from the Storage Zone.
${ERROR_CODE}	The error code if an error was encountered during the task.
${ERROR_DETAILS}	The error message if an error was encountered during the task.
${EVENT_TYPE}	The event type (Started, Error or Completed).
${EVENT_TYPE_ DESCRIPTION}	
${EVENT_TIME}	The date and time the event occurred.
${LINK}	A link to the relevant Compose for Data Lakes project.

7. Click **Next**. In the **Apply to** screen, select whether to apply the rule to all tasks of to selected tasks. If you chose **Selected Tasks**, select which tasks to apply the rule to.
8. Click **Next** to see a summary of the notification settings or **Finish** to save your settings and exit the wizard.
9. If you clicked **Next**, review your settings and then click **Finish** to save the notification rule and exit the wizard or **Prev** to edit your settings. You can also click the headings on the right of the wizard to go directly to a specific window.

   The notification will be added to the list of notifications in the Notification Rules window.

Managing Notification Rules

In the **Notification Rules** window, you can edit, delete and enable/disable notification rules as described in the table below.

To	Do This
Delete a Rule	Select the rule and then click the **Delete** toolbar button. When prompted to confirm the deletion, click **Yes**.
Edit a Rule	Either double-click the rule you want to edit or select the rule and click the **Edit** toolbar button. Continue from Defining Notifications Rules.
Disable a Rule	Select the rule you want to disable and then either click the **Disable** toolbar button or clear the check box in the **Enabled** column.
Enable a Rule	Select the rule you want to enable and then click the **Enable** toolbar button or select the check box in the **Enabled** column.

Event IDs in Windows Event Log

The table below lists the Event IDs for Compose for Data Lakes events in Windows Event Log.

If a notification is set for several events, the event ID will be 0 for each of the events.

Event ID	Description
OTHER	
261	Any error.
TASK	
400	Task has started.
406	Task has stopped due to a non-recoverable error.

Workflows

Workflows enable you to run tasks both sequentially and in parallel. You can either schedule workflows as described in Scheduling Tasks or run them manually using the **Run** toolbar button or Compose for Data Lakes CLI.

You can create your own workflow and/or use the built-in workflow. The built-in workflow enables you to run all of your tasks as a single, end-to-end process. The built-in workflow appears in the **Type** column as "Default Workflow".

When you create your *own* workflow, you decide which tasks to include in the workflow and the order in which they will be run.

Creating Workflows

This section provides instructions for creating workflows.

To create a workflow:

1. Switch to Monitor view by clicking the **Monitor** button in the top right of the Compose console.
2. Click the **New Workflow** toolbar button.

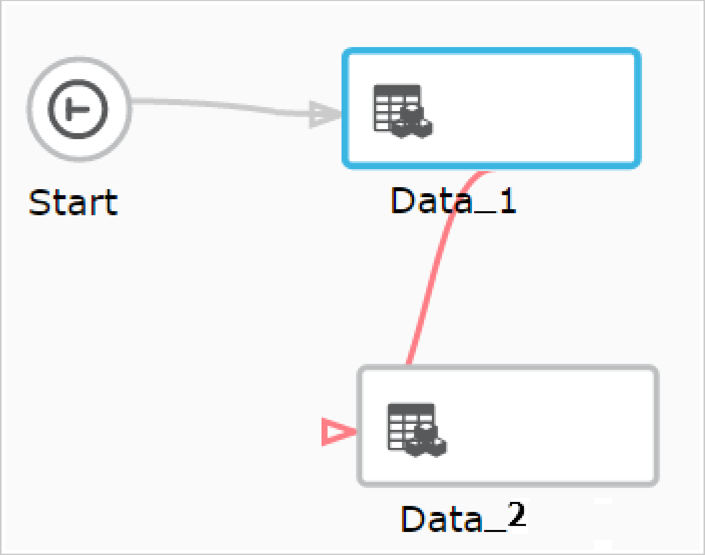
   The **New Workflow** window opens.
3. Specify a name for your workflow.
4. Optionally, select the **Duplicate from** check box and then select an existing workflow from the drop-down list.
5. Click **OK** to save your settings.

   The **<workflow_name>** window opens.
6. The window is divided into two panes. The Design pane on the left is where you design your workflow and contains two default elements: **Start** and **End**.

   The right pane contains elements and tasks that you can use in your workflow. The following elements and tasks are available:
   - **Tasks** - All existing tasks defined in Compose for Data Lakes.
   - **Workflow Elements** - There are two types of element: **Parallel Split** and **Synchronize**. Use the Parallel Split element to create parallel paths. This is useful, for example, if you want two or more tasks to run in parallel.

     Use the Synchronize element to merge parallel paths. The workflow waits for all the Tasks that precede the element to complete before continuing the flow.
7. To design your workflow:
   a. Drag the desired workflow elements from the pane on the right to the pane on the left.
   b. Arrange the elements in the order that you want them to run.
   c. Connect the elements to each other by dragging the connector from the gray dot (that appears on the right of an element when you hover the mouse cursor over it) to the target element. When a blue outline appear around the target element, release the mouse button.
   d. Optionally add error paths to the workflow. The workflow will follow the error path if a task encounters an error. For example, if an error occurs with one task, you may want to run another task in its place.

To add an error path, hover your mouse cursor over the task element. A red dot will appears below the element. Drag the connector from the red dot to the target element, as shown below.



Connecting two error paths to the same task should be avoided as the workflow will fail if the task tries to run twice.

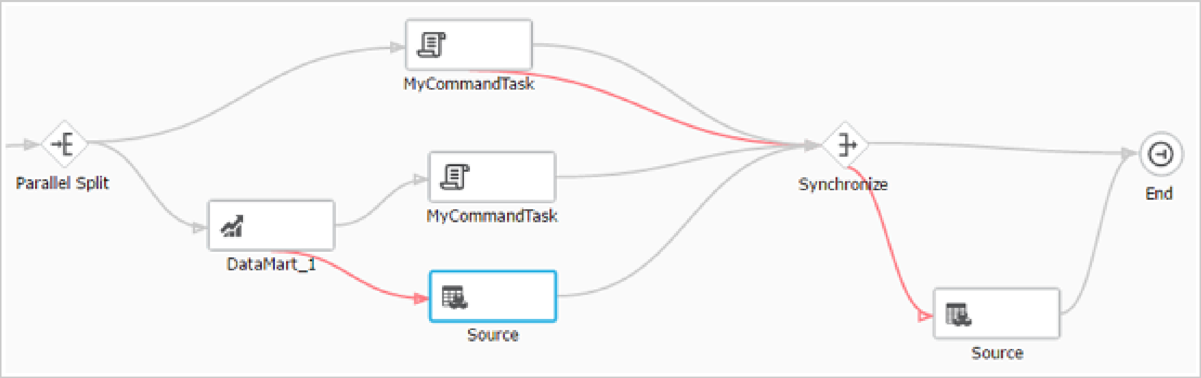
Continuing a Workflow in the Event of Parallel Task Failure

In a workflow, all task elements have an error port. This allows you to change the course of the workflow in the event of a task failure, as described in Step Creating Workflows above. Similar to Task elements, the **Synchronize** gateway also has an error port which can be used to reroute the workflow if any of the tasks between the **Parallel Split** and **Synchronize** gateways should fail.

By default, a workflow will end with an error if one or more parallel tasks do not complete successfully. However, in certain cases you may want the workflow to continue, even if one or more of the parallel tasks failed.

To do this, you need to connect the error port of the relevant task(s) directly to the **Synchronize** gateway. You can also design the workflow so that it follows the path leading from the **Synchronize** error port, instead of continuing its normal flow.

In the example below, the error port of the **MyCommandTask** is connected to the **Synchronize** gateway, meaning that even if **MyCommandTask** task fails, the workflow will continue. However, if the **MyCommandTask** task fails, the workflow will not proceed directly to the **End** element. Instead, it will follow the **Synchronize** gateway's error path to the **Source** task.



Validating Workflows

It is strongly recommended to validate your workflow before running it. This will prevent errors from occurring during runtime due to an invalid workflow.

Workflow rules include:

- All elements must be connected to each other
- A workflow must contain Start and End elements and at least one task.
- A workflow cannot contain a Parallel Split gateway without a Synchronize gateway and vice versa.
- Storage Zone tasks that update the same tables cannot run in parallel.
- A workflow cannot contain a Parallel Split gateway without a Synchronize gateway and vice versa.
- The execution order of elements must be sequential and not cyclic. For example an element cannot loop back to an element that precedes it the execution order.

To validate your workflow:

- Click the **Validate Flow** toolbar button.

If the workflow is valid, a "`<workflow_name> is valid`." message will be appear at the top of the window. If the workflow is not valid, a message describing the problems will appear instead.

Managing Workflows

The table below describes the options available for managing workflows.

To	Do This
Delete a Workflow	In Monitor view, select the workflow in the **Task** column and then click the **Delete Workflow** toolbar button.
Edit a Workflow	In Monitor view, either double-click the workflow you want to edit or select the workflow and click the **Open** toolbar button. Continue from Creating Workflows.
Delete an element in workflow	Either right-click the element and select **Delete** or select the element and then click the **Delete** toolbar button.
Reset the workflow view	Click the reset button to the right of the slider at the top of the window.
Zoom in to/zoom out of the workflow	Move the slider at the top of the window to the left or right as required.

Running and Monitoring Workflows

You can either schedule workflows as described in Scheduling Tasks or run them manually using the **Run** toolbar button. The **Run** toolbar button appears both in the main Monitor view and in the workflow design window. Note that when you run a workflow from the workflow design window, a new **Monitor** tab is added to the window and the view automatically switches to the **Monitor** tab.

You can monitor the workflow either in the **Monitor** tab or in the **Progress Status** tab. During runtime, the workflow elements fill with blue providing a graphic indication of progress. If a task encounters an error, the task element will appear with red fill instead of blue.

Monitoring and Controlling Replicate Tasks

Before you can create a Compose for Data Lakes project, you need to define a Replicate task that replicates the relevant source tables from the source database to the Landing Zone. You can define a different task for each project or the same task can serve several projects. You can also define multiple tasks for a single project. The tasks can either reside on the same Replicate server or on several Replicate servers distributed throughout your organization.

Monitoring and controlling Replicate tasks from within Compose for Data Lakes involves the following steps:

- **Step 1:** Configure Qlik Compose for Data Lakes to connect to the Qlik Replicate machine(s) as described in Managing Replicate Servers.
- **Step 2:** Add the Replicate task name to the source Landing Zone settings as described in Defining Landing Zones Connections.
- **Step 3:** Monitor and control the Replicate task as described below.

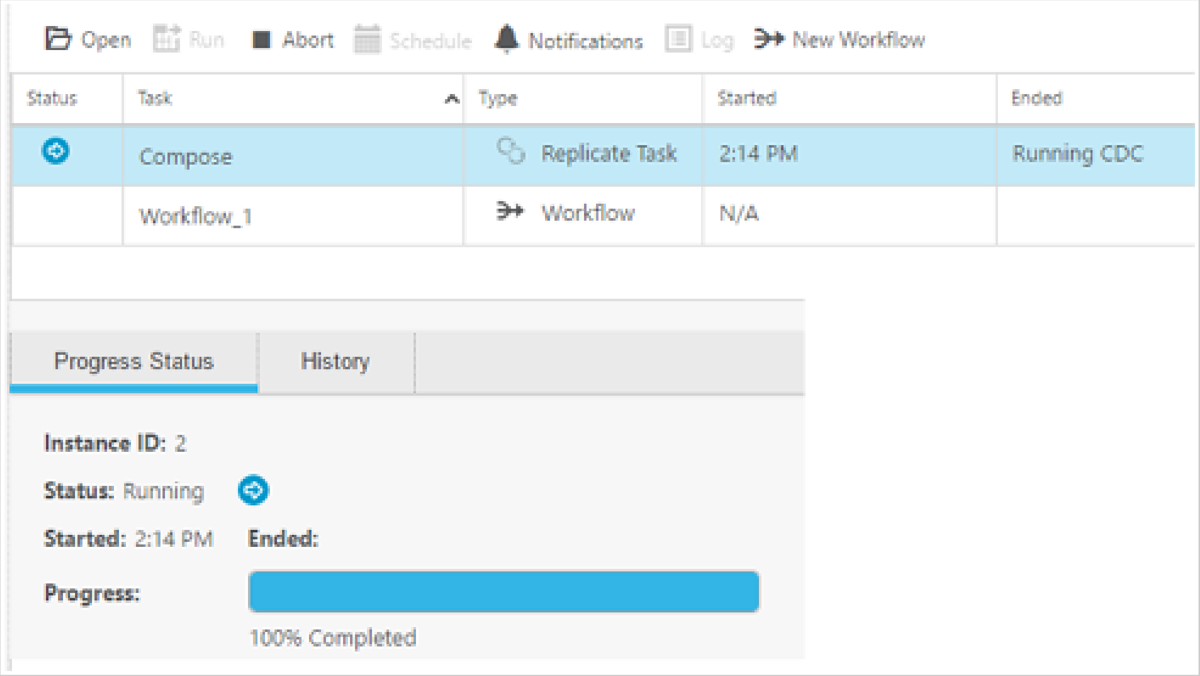
{Color}Figure Figure 11.1, "Replicate Task in the Compose for Data Lakes Monitor " {Default ¹ Font} shows how the Replicate task appears in the Compose for Data Lakes Monitor. You can stop and start the Replicate task using the **Abort** and **Run** toolbar buttons.

If a task is stopped from within Replicate, the task status in Compose for Data Lakes will be "Completed" instead of "Aborted".

You can also define notifications for the task and add the task to a workflow. For more information, see Defining Notifications Rules and Workflows respectively.

The monitor provides various information about the task. For details, see Viewing Information in the Monitor.

Figure 11.1 | Replicate Task in the Compose for Data Lakes Monitor



Status	Task	Type	Started	Ended
●	Compose	Replicate Task	2:14 PM	Running CDC
	Workflow_1	Workflow	N/A	

Instance ID: 2

Status: Running

Started: 2:14 PM **Ended:**

Progress:

100% Completed

11 Managing Compose for Data Lakes

This chapter describes the available Compose for Data Lakes management options. Qlik Compose for Data Lakes management options can be accessed from the **Management** menu located at the top of the Compose for Data Lakes main page.

In this chapter:

License Settings

You need to register a valid license in order to use Qlik Compose for Data Lakes. The license file contains details such as the product expiration date, the date the license was issued, which source databases can be used, and so on.

The following sections describe how to register and view your Compose for Data Lakes license:

- License Settings
- Viewing a License

Registering a License

This section describes how to register your Compose for Data Lakes license. You can register the license either using the console or using a command line.

To register a license using the console:

1. Copy the license file to the computer on which Compose for Data Lakes is installed or to any computer in your network that can be accessed from the Compose for Data Lakes computer.

2. From the **Management** menu, select **License**|**Register License**.

   The **Register License** window opens.

3. Either copy the license text into the text box or click **Load File** and select the license file.

   The license text is displayed in the window as shown above. Check to be sure that the details are correct.

4. Click **Register License** to register the license. A message indicating the license was registered successfully is displayed.

To register a license using the command line:

1. Open a command prompt and change the working directory to:

   `PRODUCT_DIR\bin`

   The default `PRODUCT_DIR` is: **C:\Program Files\Attunity\Compose**

2. Run the following command to connect to the server:

   ```
   ComposeCli.exe connect
   ```

3. Issue the following command:

   ```
   ComposeCli.exe register_license --req @license_file|license_text
   ```

   where:

   - *`license_file`* is the full path to the Qlik Compose for Data Lakes license file. Note that the path should always be preceded by the "@" symbol.

     Example:

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

   - *`license_text`* is a string in JSON format. When specifying a JSON string any quote symbols should be escaped using a backslash (\).

     Example:

     ```
     ComposeCli.exe register_license --req "{ \"$type\":
     \"ComposeLicense\", \"product\": \"AttunityComposeForDataLakes\",
     \"issued_to\": \"qa\", \"issued_by\": \"Attunity Israel\", 07-
     17\", \"hosts\": \"\", \"product_version\": \"2.8\", \"notes\":
     \"\", \"host_role\": \"\", \"source_db_types\": \"\", \"dwh_
     type\": \"\", \"number_of_dms\": \"0\", \"managed_dwh_size\":
     \"0\",
     ```

```
LcVLPfXvD4wY5ZyUYlasdjtOvQd1Hwk5UzT7xe5+pqhZtB1dfUUyl50+7zKju7vm1
kkPnz3I+L5LbLm3FpvqxIxOFrj2LQBk1LoUxMN+v06vI+w5aMSGQw6fttUgbYohFC
IOduk8=\"}"
```

If the license is registered successfully, the following message will be displayed:

```
Compose for Data Lakes control program completed susccessfully.
```

Otherwise, an appropriate error message will be displayed.

Viewing a License

You can view the license information in the Qlik Compose for Data Lakes Console at any time.

To view the license information:

- From the **Management** menu, select **License|View License**.

The **License** window opens, displaying all of the license information for the currently registered project type: Apache Hive or Apache Spark.

Logging Settings

This section describes the server logging settings

In this section:

- Setting Logging Levels
- Setting Automatic Roll Over and Cleanup
- Viewing and Downloading Compose for Data Lakes Log Files

Setting Logging Levels

The logging level determines what type of information is written to the log files. The log files provides information about the Compose for Data Lakes Server process and - in a Spark project - the Compose Agent processes as well.

The following logging levels are available (ordered from the lowest level to the highest level):

1. Errors Only
2. Warnings
3. Info

4. Debug
5. Detailed Debug

The higher levels always include the messages from the lower levels. Therefore, if you select **Error Only**, only error messages are written to the log files. However, if you select **Info**, informational messages, warnings, and error messages will be included. Selecting **Detailed Debug** writes all possible messages to the log.

You can set a global logging level for all log components or you can set a separate logging level for each component.

To set the logging level:

1. From the **Management** menu, select **Logs|Log Management**.

   The **Log Management** window opens displaying the **Server Log** tab and - in a Spark project - the **Compose for Data Lakes Agent** tab as well.
2. Select the desired logging tab.
3. To set a global logging level for Compose for Data Lakes Server, move the top slider to the level you want. All of the sliders for the individual modules move to the same level that you set in the main slider.
4. To set a logging level for individual Compose for Data Lakes Server components, select a module and then move its slider to the desired logging level.

   Changes to the logging level take place immediately. There is no need to restart the Attunity Compose for Data Lakes service.
5. Click **OK** to save your changes and close the **Log Management** window.

Setting Automatic Roll Over and Cleanup

You can define when log files should be automatically rolled over as well as how many days to keep log files.

To set the log file roll over and cleanup options:

1. From the **Management** menu, select **Logs|Log Management**.

   The **Log Management** window opens.
2. Select the **Log Settings** tab.
3. The following options are available:
   - **Enable automatic roll over**: Select this check box to determine the maximum size a log file can reach before it is rolled over. The current log file

is called `Attunity_Compose_FDL.log` and saved (older) log files are called `Compose_xxxxxxxxxxxx.log` where `xxxxxxxxxxxx` represents a 12-digit timestamp.

- **Roll over the log if the log file is larger than (MB)**: Use the counter or type in the maximum amount of megabytes for a specific log file. When the log file reaches the specified size, the old log is saved with a timestamp appended to its name and a new log file is started.

  The default value is 10 megabytes.

> **Note** The scheduled job that checks the log size runs every five minutes. Consequently, the actual size of the log when rolled over might be larger than specified.

- **Enable automatic cleanup:** Select this check box to define the maximum number of days a log file can exist before it is deleted.
  - **Delete log files that are older than (days)**: Use the counter or type in the maximum number of days to keep a log file. Log files that are older than the specified number of days are automatically deleted from the system. For example, if **4** is specified, then all log files will be deleted on the fifth day.

    The default value is 10 days.

4. Click **OK** to save your settings and close the **Log Management** window.

Viewing and Downloading Compose for Data Lakes Log Files

You can view and download Compose for Data Lakes Server and Compose Agent log files in the Log Viewer.

To open the Log Viewer:

- From the **Management** menu, select **Logs|View Logs**.

  The **Log File Viewer** opens.

To view log files:

1. Select one of the following from the **Log Files** drop-down list in the top left of the Log Viewer.

   To view server logs, select **Server**.

   To view Compose Agent logs, select **Compose Agent**.

2. Select the log file you want to view in the left pane.

   The contents of the log file will be displayed in the right pane. When you select a row in the log file, a tooltip will display the full message of the selected row.

3. Browse through the log file using the scroll bar on the right and the navigation buttons at the top of the window.
4. To search for a specific string in the log file, enter the search string in the search box at the top of the window.

   Any terms that match the specified string will be highlighted blue.

To download log files:

1. In the left pane, select the desired log file.
2. Click the **Download Log File** button in the top right of the Log Viewer.

   The log file will be downloaded.

Mail Server Setting

The Mail parameters define the mail server used to send notifications.

To set the log file roll over and cleanup options:

1. From the Management menu, select **Mail Settings**.

   The **Mail Server Settings** window opens.

2. Configure the settings as follows:
   - **Mail Server**: Specify the outgoing mail server that will be used to send Qlik Compose for Data Lakes notifications, for example, `smtp.example.com`.
   - **Port**: Enter the mail server port number. The default value is **25**.
   - **Use SSL**: Select this check box to connect to the mail server using SSL.
   - **Anonymous Login**: Enable this to allow Qlik Compose for Data Lakes to access the mail server without having to provide any user credentials.
   - **User Name**: Specify the user name for the account that will be used to send notifications.
   - **Password**: Specify the password for the account that will be used to send notifications.
   - **Sender Email Address**: Enter the email address that sends the email notifications. This is the address that appears in the **From** field of the email notification.

- **Send Test Mail**: You this option to validate your mail server settings.

  Click **Send Test Mail** to open the **Send Test Email** window.

  In the **Email address for test email**, enter the email address to which you want the test email to be sent and then click **Send**.

3. Click **OK** to save your settings and close the **Mail Server Settings** window.

Compose Agent Settings

These settings are relevant for Apache Spark projects only.

In an Apache Spark project, first install the Qlik Compose for Data Lakes Agent as described in Installing Compose Agent in a Non-Ephemeral Environment or Launching an Amazon EMR Cluster with Compose Agent and then configure the Compose Agent connection settings as described below.

To specify the connection settings:

1. From the **Management** menu in the projects view, select **Compose Agent Settings**.

   The **Compose Agent Settings** window opens.

2. Select **Remote Server** and provide the required connection details. Note that the password is the password you provided when you installed the Compose Agent.
3. Optionally (but recommended), click **Test Connection** to verify the settings.
4. Click **OK** to save your settings.

Managing Replicate Servers

Before you can create a Compose for Data Lakes project, you need to define a Replicate task that replicates the relevant source tables from the source database to the Landing Zone. If you want your ingested data from multiple sources, then you need to define a separate task for each source. Tasks can either reside on the same Replicate Server or on different Replicate Servers. A single Replicate task can serve several projects.

You can also monitor the Replicate task(s) from within the Compose for Data Lakes monitor. This requires you to provide the information that will allow Compose for Data Lakes to establish a connection with the Replicate Server on which the tasks are running. After providing this information, you will then be able to associate a source Landing Zone with a specific Replicate Server and task.

To add a Replicate Server:

1. From the **Management** drop-down menu in the main toolbar, select **Manage Replicate Servers**.

   The **Manage Replicate Servers** window opens.

2. Click **Add Replicate Server**.

   The **Add Server** window opens.

3. Enter the following information:

   - **Name:** A display name for the server.
   - **Description:** (Optional) A description for the server.
   - **Host:** The IP address or host name of the Qlik Replicate machine.

     When Replicate Server is installed on Linux, enter the IP address of the Windows machine on which the Replicate UI Server is running.

   - **Port:** Optionally, change the default port (443). You should only change the default port if you are certain that a different SSL port is being used.
   - **User Name** and **Password:** Your login information for the Qlik Replicate machine.

     When Replicate Server is installed on Linux, enter the user name and password for the Windows machine on which the Replicate UI Server is running.

4. Click **Test Connection** and then click **OK** if the connection is successfully verified.

To remove a Replicate Server:

1. From the **Management** drop-down menu in the main toolbar, select **Manage Replicate Servers**.

   The **Manage Replicate Servers** window opens.

2. Select the server you want to remove and then click **Delete**.

To edit a Replicate Server:

1. From the **Management** drop-down menu in the main toolbar, select **Manage Replicate Servers**.

   The **Manage Replicate Servers** window opens.

2. Select the desired server and then click **Edit**.
3. Edit the server settings as described above.

Setting up User Permissions

You can grant Qlik Compose for Data Lakes users different permissions according to the tasks you want them to perform. Four predefined "roles" are available: Admin, Designer, Operator and Viewer. Each role has its own set of permissions, which are described in the following table.

Role	Active Directory Group
Administrator	AttunityComposeForDataLakesAdmins
Designer	AttunityComposeForDataLakesDesigners
Operator	AttunityComposeForDataLakesOperators
Viewer	AttunityComposeForDataLakesViewers

Managing User Permissions

This section explains how to edit user permissions, and how to add and remove users or groups.

To edit the user permissions:

1. From the **Management** menu in the projects view, select **User Permissions**.

   The **User Permissions** window opens.
2. Adjust the permission sliders as desired.
3. Click **OK** to save your settings.

To add new users or groups:

1. From the **Management** menu in the projects view, select **User Permissions**.

   The **User Permissions** window opens.
2. Click the **Add** toolbar button.

   The **Add User/Group** window opens.
3. Select **User** or **Group** as appropriate.
4. Enter the user or group name in the following format:

   For domain users/groups: `domain\group_name` or `domain\user_name`

- Only NetBIOS domain names are supported. The NetBIOS domain name is the leftmost label in the DNS domain name. For example, entering `qa.int\mike` would result in a connection error whereas entering `qa\mike` would not.
- Active Directory distribution groups are not supported.

For local users/groups: `computer_name\group_name` or `computer_name\user_name`

5. Click **OK** to add the user/group to the **User/Group** list.

To remove a user or group:

1. From the **Management** menu in the projects view, select **User Permissions**.

   The **User Permissions** window opens.

2. Select the user/group you want to remove and then click the **Remove** toolbar button.

   The user/group is removed.

3. Click **OK** to save your settings.

Audit Trails

The information provided in an Audit Trail can be leveraged for user accountability, reconstruction of events, intrusion detection, and other operational issues. As such, Audit Trails are an indispensable tool for regulatory compliance (e.g. SOX).

For operations performed by users with Operator privileges or above, the Compose for Data Lakes Audit Trail shows which user performed the operation, when it was performed, and on which objects.

Compose for Data Lakes retains audit files for two weeks or until they reach a total size of 500 MB (50 files). You can change these settings through the command line interface (CLI) as described in Exporting Audit Trail files below.

Audit Trail files are located in the following folder:

<Installation_Directory>\data\AuditTrail\audit_service

You can also export an audit trail file for a specific time range, as described in Exporting Audit Trail files.

Audit Trail Information

Audit Trail files provide all or some of the following information:

- **Timestamp** - The time when the row was inserted into the audit trail.
- **User** - The user that performed the operation.
- **Node** - The IP of the server on which the operation was performed.
- **Requested Action** - The API method/function that was called.
- **Required Permission** - The minimum role of the user that can perform the operation.
- **Effective Permission** - The actual role of the user that performed the operation.
- **Security Result** - Whether the user is allowed to perform the operation.
- **Action Result** - The completion status of the operation (success of failure).
- **Error Message** - The error message if the operation failed.
- **Task** - The name of the task where relevant.
- **Notification** - The notification defined for the operation (if defined).
- **Payload** - A URL. To view payload information, simply copy the link from the **Payload** column and paste it into your browser's address bar.

  Payloads for some operations (e.g. RegisterLicense) contain sensitive information and need to be decoded. For information on decoding payloads, see Decoding an Encoded Payload.
- **Project Name** - The name of the Compose for Data Lakes project.

Audit Trail files are compressed and tamper-protected.

Exporting Audit Trail files

Note Exporting an audit trail using the CLI is supported from Compose for Data Lakes 6.6 SP11 only.

You can export an audit trail file with a record of activity for a specific time range. In Compose, there are two way of doing this:.

- Using the management console
- Using the CLI

Note You can also export audit trails using the ExportAuditTrail API method. For further information, see the *Qlik Enterprise Manager Help and API Guide*.

Exporting an Audit Trail file via the management console

You can use the Compose for Data Lakes management console to export the audit trail as a CSV file.

To do this:

1. From the **Management** drop-down menu, select **Audit Trail**. The Audit Trail window opens.
2. From the **Time Range** drop-down list, select the desired time range. If you select **Custom**, set **From** and **To** values as well.
3. Click **Generate**.

Depending on your browser settings, you will either be prompted for a download location or the file will be downloaded automatically to your preferred location.

Exporting an Audit Trail file via the CLI

You can use the Compose for Data Lakes CLI to export the audit trail as a JSON file.

To do this:

1. Open a command prompt as Admin and switch the path to <COMPOSE_INSTALL_DIR>\bin.
2. Establish a connection to the Compose server by running the following command:

   ```
   ComposeCli.exe connect
   ```

3. To export the audit trail, run the following command:

   ```
   ComposeCli.exe generate_audit_trail --start_timestamp timestamp [--
   end_timestamp timestamp] --outfile full_path
   ```

   Example:

   ```
   ComposeCli.exe generate_audit_trail --start_timestamp 2020-06-
   30T16:15:00Z --end_timestamp 2020-07-14T16:15:00Z --outfile
   "C:\compose audit trails\audit.csv"
   ```

   Where:

   - `--start_timestamp` is the date and time from which you want the audit trail to start, in UTC format.
   - `--end_timestamp` is the date and time on which you want the audit trail to end, in UTC format. This is optional. When not specified, the file will end at the latest audit trail record.

- `--outfile` is the full path to the output file. If the path contains spaces, it should be enclosed in quotation marks.

Configuring Audit Trail Size and Retention

To do this:

1. Open a command prompt and change the working directory to:

   `<COMPOSE_INSTALL_DIR>\bin>`

   Default:

   `C:\Program Files\Attunity\Compose for Data Lakes\bin>`

2. Run the following command:

   `ComposeCli.exe connect`

   The following message should be displayed:

   `ComposeForDataLakes Control Program completed successfully.`

3. Run the following command:

   `ComposeCli.exe audit_trail control --age` *`weeks`* `--size` *`megabytes`*

   Where:

   `weeks` is the number of weeks to retain the audit trail file (default 2 weeks)

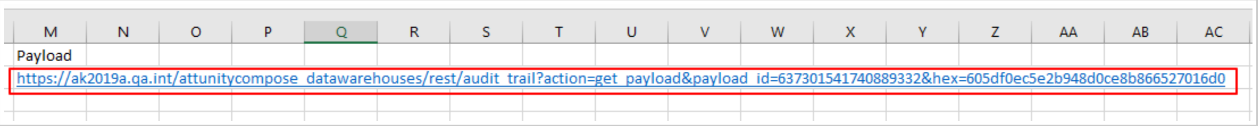
   `megabytes` is the maximum size of the audit file to retain (default 500 MB)

Decoding an Encoded Payload

Some audit records (e.g. RegisterLicenses) may contain an encoded payload. Encoded payloads are displayed as byte arrays and need to be decoded using Base64.

To decode an encoded stream payload:

1. Locate the payload URL in the audit record.

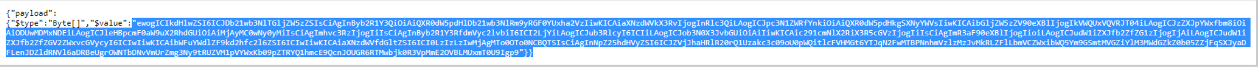


Payload

https://ak2019a.qa.int/attunitycompose_datawarehouses/rest/audit_trail?action=get_payload&payload_id=637301541740889332&hex=605df0ec5e2b948d0ce8b866527016d0

2. Copy the URL into your browser's address bar and press [Enter].

   A byte array will be displayed.



```
{"payload":
{"$type":"Byte[]","$value":"ewogICIkdHlwZSI6ICJDb21wb3NlTGljZW5zZSIsCiAgInByb2R1Y3QiOiAiQXR0dW5pdHlDb21wb3NlRm9yRGF0YUxha2VzIiwKICAiaXNzdWVkX3RvIjogInRlc3QiLAogICJpc3N1ZWRfYnkiOiAiQXR0dW5pdHkgSXNyYWVsIiwKICAibGljZW5zZV90eXBlIjogIkVWQUxVQVRJT04iLAogICJzZXJpYWxfbm8iOi
AiODUwMDMxNDEiLAogICJleHBpcmF0aW9uX2RhdGUiOiAiMjAyMC0wNy0yMiIsCiAgImhvc3RzIjogIiIsCiAgInByb2R1Y3RfdmVyc2lvbiI6ICI2LjYiLAogICJub3RlcyI6ICIiLAogICJob3N0X3JvbGUiOiAiIiwKICAic291cmNlX2RiX3R5cGVzIjogIiIsCiAgImR3aF90eXBlIjogIioiLAogICJudW1iZXJfb2ZfZG1zIjogIjAiLAogICJudW1i
ZXJfb2ZfZGV2ZWxvcGVycyI6ICIwIiwKICAibWFuYWdlZF9kd2hfc2l6ZSI6ICIwIiwKICAiaXNzdWVfdGltZSI6ICI0LzIzLzIwMjAgMTo0OTo0NCBQTSIsCiAgInNpZ25hdHVyZSI6ICJZVjJhaHRlR20rQ1Uzakc3c09oU0pWQitlcFVHMGt6YTJqN2FwMTBPNnhmVzlzMzJvMkRLZFlLbmVCZWxibWQ5Ym9GSmtMVGZiYlM3MWdGZkZ0b05ZZjFqSXJyaD
FLenJDZldRNVl6aDRBeUgrOWNTbDNvVmUrZmg3Ny9tRUZVM1pVYWxKb09pZTRYQ1hmcE9QcnJOUGR6RTMwbjk0R3VpMmE2OVBLMUxmT0U9Igp9"}}
```

3. Copy the byte array into a Base64 decoder and decode it.

A Setting up Qlik Compose for Data Lakes on a Windows HA Cluster

This appendix describes how to set up Compose for Data Lakes in a Windows Server High Availability Cluster environment. For instructions on how to set up a Windows clustering environment, refer to the Microsoft Help.

The steps should be performed in order.

In this appendix:

Step 1: Installing Qlik Compose for Data Lakes in the Cluster

This topic describes how to install Qlik Compose for Data Lakes in a high availability cluster environment.

Preparation

1. Allocate two shared folders: one for the Compose for Data Lakes Server and one for the Compose Agent.

   The setup instructions below assume that the Compose for Data Lakes Server data folder is **F:\server-data** and the Compose Agent folder is **F:\agent-data**.

2. Generate a 32 character random master key by running the following command from <PRODUCT_DIR>\bin:

   ```
   ComposeCtl.exe utils genpassword
   ```

The setup instructions below assume that your key is

```
WdAHWEwXSvwxDFetcl7TVVFfSXPbMrFx
```

Primary Node Setup

1. Install Qlik Compose for Data Lakes.
2. Stop the Attunity Compose for Data Lakes service.
3. Edit the service executable path as follows:

   ```
   SC CONFIG AttunityComposeForDataLakes binPath="<PRODUCT_
   DIR>\bin\ComposeCtl.exe -d F:\server-data service run”
   ```

4. Run the following command from <PRODUCT_DIR>\bin:

   ```
   ComposeCtl.exe -d "F:\server-data" setup install
   ```

5. Run the following command from <PRODUCT_DIR>\bin:

   ```
   ComposeCtl.exe -d "F:\server-data" masterukey set -p
   WdAHWEwXSvwxDFetcl7TVVFfSXPbMrFx
   ```

6. Edit <PRODUCT_DIR>\java\bin\acjs.bat and immediately below the line with `SET JAVA_LIB_PATH`, add the following:

   ```
   set AT_DATA=-d F:\agent-data
   ```

7. Start the Attunity Compose for Data Lakes service and then stop it. This will create the java repository.

   This step should be performed on the primary node only.

8. Run the following command from <PRODUCT_DIR>\java\bin:

   ```
   acjs.bat masterukey set WdAHWEwXSvwxDFetcl7TVVFfSXPbMrFx
   ```

9. In the Cluster Manager, move to the secondary node.

Secondary Node Setup

Perform 1-8 of the the Primary Node Setup steps on the secondary node (excluding step 7) and then start the Attunity Compose for Data Lakes service.

Step 2: Adding the Qlik Compose for Data Lakes Service

After installing Qlik Compose for Data Lakes in the cluster, you need to add the Attunity Compose for Data Lakes service as a resource to the service (Windows Server 2008

Cluster) or role (Windows Server 2012\2016 Cluster).

To add the Attunity Compose for Data Lakes service:

1. Do one of the following (according to your Windows Server version):
   - **Windows Server 2008 Cluster:** In the left pane of the Failover Cluster Manager, right-click the service and point to **Add a resource**. Then select **Generic Service**.
   - **Windows Server 2012\2016 Cluster:** In the left pane of the Failover Cluster Manager, select **Roles**. The available roles will be listed in the right pane of the console. Right-click the role you are working with and point to **Add a resource**. Then select **Generic Service**.
2. In the **Select Service** screen of the New Resource wizard, select **Attunity Compose for Data Lakes** from the list.
3. Click **Next** and follow the directions in the wizard to create the resource. For information on how to use this wizard, see the Microsoft online help.

Note Qlik Compose for Data Lakes must be installed on the computer where you defined the service in order for the service to be available in the list.

Step 3: Defining the Service Dependencies

You need to define dependencies for the Attunity Compose for Data Lakes service that will enable the Storage and Network names to start up before the service. If these resources do not start before the service, Qlik Compose for Data Lakes will not be able to start as it will be searching for the data location.

To define the service dependencies:

1. Do one of the following (according to your Windows Server version):
   - **Windows Server 2008 Cluster:** In the left pane of the Failover Cluster Manager, select the Attunity Compose for Data Lakes service.

     The properties for this service are displayed in the center pane.
   - **Windows Server 2012\2016 Cluster:** In the left pane of the Failover Cluster Manager console, select **Roles**. From the list of available roles in the right pane of the console, select the role you are working with and then, in the bottom right pane, select the **Resource** tab. From the list of the available roles, select **Qlik Compose for Data Lakes**.

2. Do one of the following (according to your Windows Server version):
   - **Windows Server 2008 Cluster:** In the **Other Resources** section, double-click the **Attunity Compose for Data Lakes** service.
   - **Windows Server 2012\2016 Cluster:** Right-click the **Qlik Compose for Data Lakes** role and select **Properties**.
3. In the **Qlik Compose for Data Lakes Properties** dialog box that opens, select the **Dependencies** tab.
4. Click **Insert**. A new line is added to the Resource list.
5. In the **Resource** column, click the arrow and select the Qlik Compose for Data Lakes **Data storage** resource from the list.
6. Click **Insert** and add the Network Name resource (it should have the same name as the cluster).
7. Start the Service using the Failover Cluster Manager and access the console using the Network name.
8. Register the license. The license should contain all host names of the cluster.

> **Note** To open the Qlik Compose for Data Lakes Console, it is recommended to use an address that includes the name or IP address of the cluster machine (as opposed to the specific node name).
>
> **Example:**
>
> `https://cluster_name_ip/attunitycompose_datalakes/6.3.118/#`

Step 4: Defining the URL for the Cluster

By default, the Attunity Compose for Data Lakes service generates the UI URL when it starts, according to the hostname of the machine on which Qlik Compose for Data Lakes is installed.

In a cluster environment, this is not good practice because the URL will change each time the cluster is rolled over. To resolve this issue, you need to set the cluster name as the Qlik Compose for Data Lakes URL.

To set the cluster name as the Qlik Compose for Data Lakes URL:

1. In the left pane of the Failover Cluster Manager, select **Nodes**. The right pane of the Console displays a list of cluster nodes.
2. Select a node to see the cluster name. This is the name you want to set (for example: `Cluster_Network_1`).

The cluster name must be registered in DNS, before you can set it.

3. Run the following command from the primary node: `<PRODUCT_DIR>\bin>ComposeCtl.exe -d <COMPOSE_DATA_FOLDER> configuration set --address <CLUSTER_NAME>`

   The host configuration will be updated.

4. Restart the Attunity Compose for Data Lakes service for the changes to take effect.
5. To make sure Qlik Compose for Data Lakes is now using the correct URL, use the `<COMPOSE_DATA_FOLDER>\service.url` shortcut to check the cluster name in the service Properties.
6. Try to open Qlik Compose for Data Lakes from a remote browser using `<COMPOSE_DATA_FOLDER>\service.url`.

Upgrading Qlik Compose for Data Lakes on the Cluster

To upgrade Qlik Compose for Data Lakes on a Windows Server High Availability Cluster:

1. Make sure you are on the Primary node.
2. In the left pane of the Failover Cluster Manager, select **Roles**. From the list of available roles in the right pane, right-click the Qlik Compose for Data Lakes role you are working with and change it to offline.
3. Run the standard Upgrade procedure.
4. Bring back online the Qlik Compose for Data Lakes role and make sure there are no connection errors.
5. Repeat steps 2-4 on the Secondary node.

B Impact of DST Change on Qlik Compose for Data Lakes

This appendix describes how Qlik Compose for Data Lakes is affected by Daylight Saving Time (DST) and provides guidelines for handling changes brought about by DST.

There are two types of DST changes:

- **DST On** - Occurs approximately when Summer starts (actual date is country specific). Its impact on local time is that local time is moved one hour forward (so, for example, 01:00AM becomes 02:00AM). This DST change does not impact Qlik Compose for Data Lakes as it does not result in time overlap.
- **DST Off** - Occurs approximately when Winter starts (actual date is country specific). Its impact on local time is that local time is moved back one hour (so, for example, 02:00AM becomes 01:00AM). This DST change results in time overlap where local time travels over the same hour twice in a row.

The comments below assume that the customer has not changed the time but rather the timezone or the DST setting. Changing the actual time (not for minor time adjustments) is a sensitive operation and is best done when Qlik Compose for Data Lakes is stopped.

There are two places where DST may have an effect:

1. Timestamps in logs and audit messages are in local time. As a result, when Winter time starts, the logs will show the time going back an hour; conversely, when Summer time starts, the logs may appear to be missing one hour.
2. Statistics shown on the console are also sensitive to local time and thus may also show confusing/inaccurate data in the overlap period (going in to Winter time) or for the skipped period (going into Summer time).

In general, it is recommended to avoid non-critical task design changes during the first overlap period (going in to Winter time) so as to prevent confusion about when the changes took place.

In addition to Qlik Compose for Data Lakes, other components are also involved including:

- The source endpoint system
- The target endpoint system
- The local operating system
- The task design (specifically using timestamp based variables)

Given the complexity of the topic and the involvement of many independent components and settings, Qlik generally recommends that customers first verify the impact of DST changes in their test environment.

C Cron Format and Examples

Cron expressions can be used to schedule a Compose for Data Lakes task. This appendix describes the Cron format used in Compose for Data Lakes (Quartz), provides a description of the special characters that can be used in an expression and ends with some examples of Cron usage.

In this appendix:

Cron Format

A cron expression is a string comprised of five fields separated by a white space. Fields can contain any of the allowed values, along with various combinations of the allowed special characters for that field. The fields are described in the table below.

Field Name	Mandatory	Allowed Values	Allowed Special Characters
Seconds	✓	0-59	, - * /
Minutes	✓	0-59	, - * /
Hours	✓	0-23	, - * /
Day of month	✓	1-31	, - * ? / L W
Month	✓	1-12 or JAN-DEC	, - * /
Days of week	✓	1-7 or SUN-SAT	, - * ? / L #

Special Characters

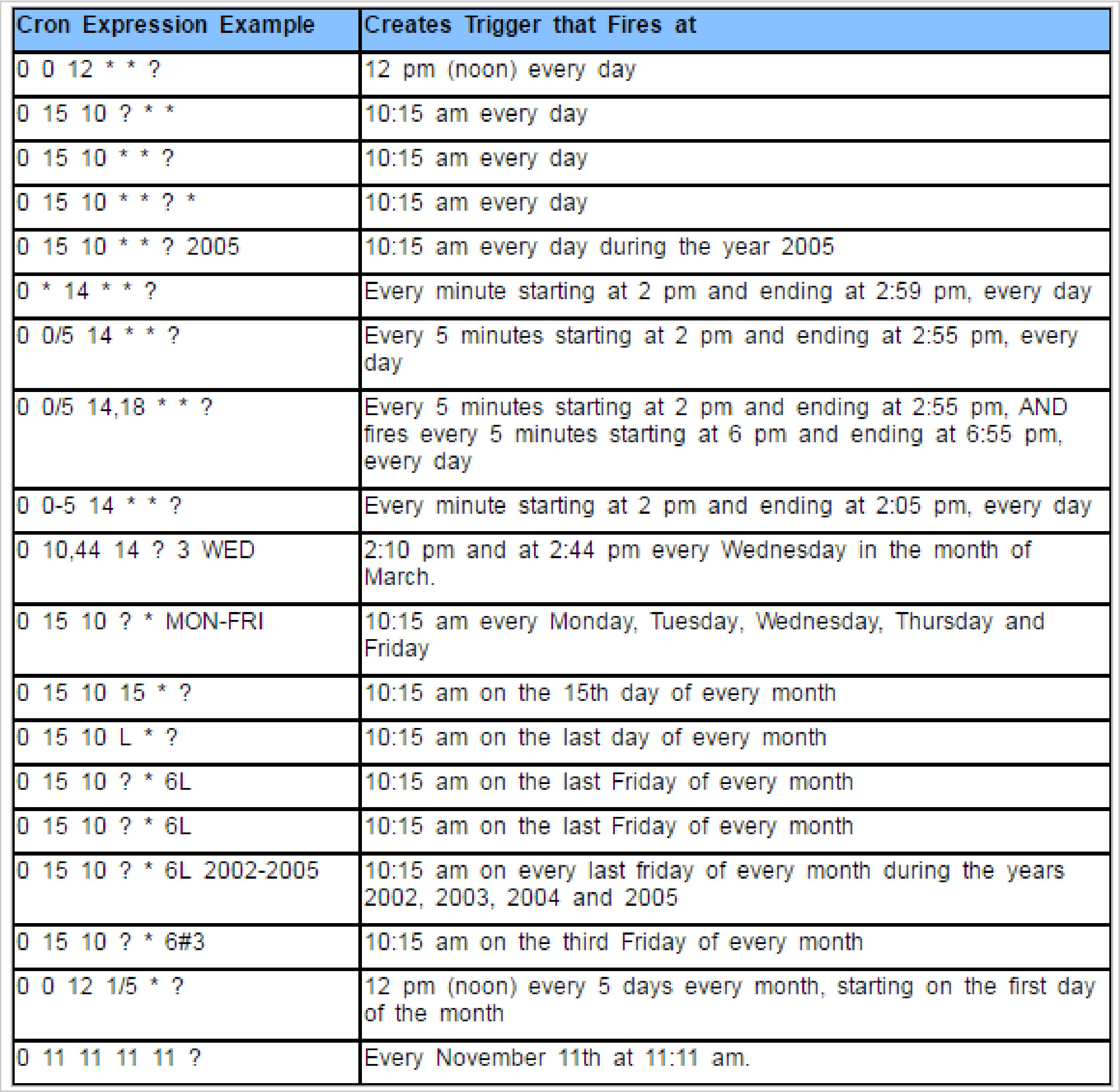
The following special characters are supported:

- ***** ("all values") Used to select all values within a field. For example, ***** in the minute field means "every minute".
- **?** ("no specific value") Useful when you need to specify something in one of the two fields in which the character is allowed, but not the other. For example, if I want my

task to run on a particular day of the month (say, the 10th), but don't care what day of the week that happens to be, I would put "10" in the day-of-month field, and "?" in the day-of-week field. See the examples below for clarification.

- **-** Used to specify ranges. For example, "10-12" in the hour field means "the hours 10, 11 and 12".
- **,** Used to specify additional values. For example, "MON,WED,FRI" in the day-of-week field means "the days Monday, Wednesday, and Friday".
- **/** Used to specify increments. For example, "0/15" in the seconds field means "the seconds 0, 15, 30, and 45". And "5/15" in the seconds field means "the seconds 5, 20, 35, and 50". You can also specify '/' after the '' character - in this case '' is equivalent to having '0' before the '/'. '1/3' in the day-of-month field means "run every 3 days starting on the first day of the month".
- **L** ("last") Has a different meaning in each of the two fields in which it is allowed. For example, the value "L" in the day-of-month field means "the last day of the month" - day 31 for January, day 28 for February on non-leap years. If used in the day-of-week field by itself, it simply means "7" or "SAT". But if used in the day-of-week field after another value, it means "the last xxx day of the month" - for example "6L" means "the last friday of the month". You can also specify an offset from the last day of the month, such as "L-3" which would mean the third-to-last day of the calendar month. When using the 'L' option, it is important not to specify lists, or ranges of values, as you'll get confusing/unexpected results.
- **W** ("weekday") Used to specify the weekday (Monday-Friday) nearest the given day. As an example, if you were to specify "15W" as the value for the day-of-month field, the meaning is: "the nearest weekday to the 15th of the month". So if the 15th is a Saturday, the trigger will run on Friday the 14th. If the 15th is a Sunday, the trigger will run on Monday the 16th. If the 15th is a Tuesday, then it will run on Tuesday the 15th. However if you specify "1W" as the value for day-of-month, and the 1st is a Saturday, the trigger will run on Monday the 3rd, as it will not 'jump' over the boundary of a month's days. The 'W' character can only be specified when the day-of-month is a single day, not a range or list of days. ** The 'L' and 'W' characters can also be combined in the day-of-month field to yield 'LW', which translates to "last weekday of the month".
- **#** Used to specify "the nth" XXX day of the month. For example, the value of "6#3" in the day-of-week field means "the third Friday of the month" (day 6 = Friday and "#3" = the 3rd one in the month). Other examples: "2#1" = the first Monday of the month and "4#5" = the fifth Wednesday of the month. Note that if you specify "#5" and there is not 5 of the given day-of-week in the month, then no firing will occur that month. ** The legal characters and the names of months and days of the week are not case sensitive. MON is the same as mon.

Usage Examples



Cron Expression Example	Creates Trigger that Fires at
0 0 12 * * ?	12 pm (noon) every day
0 15 10 ? * *	10:15 am every day
0 15 10 * * ?	10:15 am every day
0 15 10 * * ? *	10:15 am every day
0 15 10 * * ? 2005	10:15 am every day during the year 2005
0 * 14 * * ?	Every minute starting at 2 pm and ending at 2:59 pm, every day
0 0/5 14 * * ?	Every 5 minutes starting at 2 pm and ending at 2:55 pm, every day
0 0/5 14,18 * * ?	Every 5 minutes starting at 2 pm and ending at 2:55 pm, AND fires every 5 minutes starting at 6 pm and ending at 6:55 pm, every day
0 0-5 14 * * ?	Every minute starting at 2 pm and ending at 2:05 pm, every day
0 10,44 14 ? 3 WED	2:10 pm and at 2:44 pm every Wednesday in the month of March.
0 15 10 ? * MON-FRI	10:15 am every Monday, Tuesday, Wednesday, Thursday and Friday
0 15 10 15 * ?	10:15 am on the 15th day of every month
0 15 10 L * ?	10:15 am on the last day of every month
0 15 10 ? * 6L	10:15 am on the last Friday of every month
0 15 10 ? * 6L	10:15 am on the last Friday of every month
0 15 10 ? * 6L 2002-2005	10:15 am on every last friday of every month during the years 2002, 2003, 2004 and 2005
0 15 10 ? * 6#3	10:15 am on the third Friday of every month
0 0 12 1/5 * ?	12 pm (noon) every 5 days every month, starting on the first day of the month
0 11 11 11 11 ?	Every November 11th at 11:11 am.

D Control Tables

Compose for Data Lakes creates various "Control Tables" in the Storage Zone during runtime. These tables provide information that can be used for post-processing or querying the Storage Zone data.

Only relevant for Apache Hive project types.

In this appendix:

Apply Batches

The **attrep_apply_batches** table contains records of batches (groups of partitions) which were read and processed by Compose for Data Lakes.

Column	Type	Description
APPLIER_TASK_ NAME	VARCHAR (255)	The Compose for Data Lakes project type.
BATCH_NAME	VARCHAR (32)	The name of the Compose for Data Lakes batch (consists of the batch start time and end time).
BATCH_START_TIME	VARCHAR (32)	When the Compose for Data Lakes batch containing the CDP changes started.
BATCH_END_TIME	VARCHAR (32)	When the Compose for Data Lakes batch containing the CDP changes ended.
LOADER_SERVER_ NAME	VARCHAR (255)	The host name of the Replicate Server machine.
LOADER_TASK_ NAME	VARCHAR (255)	The Replicate task name.

Column	Type	Description
CDC_PARTITION_ NAME	VARCHAR (32)	The partition name consists of the partition start and end time. **Example:** 20170313T123000_20170313T170000
CDC_PARTITION_ START_TIME	VARCHAR (32)	When the partition was opened: **Example:** 2017-03-13 12:30:00.000
CDC_PARTITION_ END_TIME	VARCHAR (32)	When the partition was closed: **Example:** 2017-03-13 17:00:00.000
TABLE_OWNER	VARCHAR (255)	The table schema or owner.
TABLE_NAME	VARCHAR (255)	The table name.
COMPOSE_TASK_ NAME	VARCHAR (32)	The Compose for Data Lakes task name.
COMPOSE_SOURCE_ LANDING_ZONE	VARCHAR (32)	The Compose for Data Lakes Landing Zone database name.

Apply Query Batches

The **attrep_apply_query_batches** table contains a history of batches created when the Sqoop incremental import (load) task setting is enabled.

Column	Type	Description
APPLIER-TASK_NAME	VARCHAR (255)	The Compose for Data Lakes project type.
BATCH_NAME	VARCHAR (32)	The name of the Compose for Data Lakes batch (consists of the batch start time and end time).
BATCH_ START_TIME	VARCHAR (32)	When the Compose for Data Lakes batch containing the changes started.

Column	Type	Description
BATCH_END_TIME	VARCHAR (32)	When the Compose for Data Lakes batch containing the changes ended.
CONTEXT_START_TIME	VARCHAR (255)	When changes started: **Example:** 2017-03-13 12:30:00.000
CONTEXT_END_TIME	VARCHAR (255)	When the changes ended: **Example:** 2017-03-13 12:40:00.000
TABLE_OWNER	VARCHAR (255)	The table schema or owner.
TABLE_NAME	VARCHAR (255)	The table name.

DDL History

The **attrep_ddl_history** table contains a history of DDL changes that occurred in the Storage Zone.

The default prefix `attrep_` might have been changed to allow reuse of the same database for Landing, Storage, and Provisioning Zones.

A new record is inserted into the table whenever a supported DDL change occurs in the source. Multiple ALTER TABLE statements that occur during a task may be represented as a single row in the Control Table.

Column	Type	Description
CDC_PARTITION_NAME	VARCHAR (32)	The name of the Change Data Partition created by Replicate.
REPLICATE_LAST_CHANGES_END_TIME	TIMESTAMP	When the last change occurred in the Replicate Change Data Partition.
DDL_BATCH_START_TIME	TIMESTAMP	When the DDL batch started in Compose for Data Lakes.
DDL_BATCH_END_TIME	TIMESTAMP	When the DDL batch ended in Compose for Data Lakes.

Column	Type	Description
LANDING_DATABASE	VARCHAR (128)	The Landing Zone database name.
REPLICATE_CURRENT_CLOSED_ PARTITION_END_TIME	TIMESTAMP	When the current Replicate Change Data Partition was closed.

E Supported Platforms, Databases and Replicate Versions

In addition to listing the platforms on which Qlik Compose for Data Lakes can be installed, this appendix also specifies which source and target database versions can be used in a Qlik Compose for Data Lakes task.

Supported Platforms

This topic lists the platforms on which Qlik Compose for Data Lakes and Compose Agent can be installed.

Supported Windows Platforms for Compose for Data Lakes

Qlik Compose for Data Lakes can be installed on any of the following Windows platforms:

- Windows Server 2012 (64-bit)
- Windows Server 2012 R2 (64-bit)
- Windows Server 2016 (64-bit)
- Windows Server 2019 (64-bit)

Supported Linux Platforms for Compose Agent

Compose Agent can be installed on any of the following Linux platforms:

- Centos 6.9
- Amazon Linux
- Debian 16.04

Supported Browsers

The Qlik Compose for Data Lakes Web UI supports the following browsers:

- Internet Explorer 11 and above
- Chrome (always updates itself to the latest version)
- Firefox (always updates itself to the latest version)

Supported Replicate Versions

Compose for Data Lakes supports working with Replicate versions November 2020 (from Service Pack 09), 6.6 and 6.5.

Supported Hive Distributions

The table below lists the Hive distributions supported by Qlik Compose for Data Lakes.

Hive Distribution	Version
Hortonworks	**Hive projects:** 2.x (where x>=5) and 3.1.x Tables will be created as transactional tables (ACID). **Tip:** It is recommended that you set the `metastore.catalog.default` to Hive in Apache Ambari for the entire cluster. 2.5.x is only supported when using a Historical Data Store (no ACID). **Spark projects:** 2.5.x and 2.6.x, and 3.1.x **Tip:** For each Spark task, it is recommended to add the following Spark setting that overrides the default task settings: `spark.hadoop.metastore.catalog.default=hive`
Amazon EMR	**Hive projects:** 5.x (where x>=15) 6.x (where x>=1) - Supported from Compose for Data Lakes SP09 only **Spark projects:** 5.15 - 5.27 when AWS Glue is not used 5.28 and 5.29 - Supported with or without AWS Glue

Hive Distribution	Version
Cloudera	**Hive projects:** 5.x (where x>=11), 6.x, and 7.x. Note that Cloudera 7.x is supported from Compose for Data Lakes SP11 only. **Spark projects:** 6.1
Microsoft Azure HDInsight	3.6
Google Dataproc	1.2, 1.3 and 1.4 on Debian only.
Google BigQuery	Supported only for provisioning tasks.
Databricks on AWS	6.x Fully binary compatible versions are also supported.
Databricks on Azure	6.x Fully binary compatible versions are also supported.

F Supported Characters

To prevent character validation errors, Compose for Data Lakes best practice is to only use alphanumeric characters, underscores and hyphens in table and column names. This is because object naming rules are always determined by the database type, of which there may be several in a single Compose for Data Lakes project. For example, if the Storage Zone is configured to use HDFS and the Provisioning Zone is configured to use Google Cloud Storage, column names which are supported in HDFS may not be supported in Google Cloud Storage (and vice versa).

Glossary

A

Attribute
In the table metadata, an attribute is a logical representation of a physical column in a Landing Zone table.

Attributes Domain
A list of all the attributes available in the metadata. You can add, edit and delete attributes according to your data warehousing needs. The Attributes Domain also shows you which entities each attribute is used in, as a single attribute may be used in several entities.

C

Change Tables
Change Tables are created in the landing area when the Replicate task is defined as Full Load and Store Changes or Store Changes only. When the Store Changes replication option is enabled in the Replicate task, any changes to the source tables will be replicated to the Change Tables in the landing area. The Change Table name format comprises the original table name appended with a "__ct".

E

Entity
In the table metadata, an entity is a logical representation of a physical Landing Zone table or view.

F

Full Load
A Full Load replication task is a Replicate task that replicates all of the selected source tables to the landing zone and populates them with data from the source database. When you duplicate an existing delivery zone task, you can set the task type to Full Load and Change Tables (i.e. initially extract all the data from the landing zone tables and then only the changes), Full Load Only (i.e. extract all the data from the landing zone tables) or Change Tables Only (i.e. extract only the changes to the landing zone tables).

L

Landing Zone
The zone (database) in Hive to which the source tables are replicated. This is also the target endpoint in a Replicate task.

M

Metadata
The metadata should contain all of the information needed to create the tables in the Delivery Zone. Metadata can be generated automatically by discovering (otherwise known as reverse engineering) the Landing Zone.