

Upgrade Guide

- [Upgrading From an Older Release](#)
 - [Upgrading to version 6.3](#)
 - [Parquet is the new default storage format](#)
 - [External authentication changes](#)
 - [Upgrading to versions 7.0+](#)
 - [Upgrading to version 7.2](#)
 - [Upgrading with Kerberos](#)
 - [Migrating workbooks and other artifacts when upgrading](#)
- [Upgrade Planning](#)
 - [Disabling Housekeeping/Compaction services](#)
- [Workbook Validation](#)
- [Backup Keyfiles and Keystore](#)
- [Upgrade the Application](#)
- [Database Upgrade](#)
 - [Upgrade the MySQL database](#)
 - [MySQL thread stack overrun during database upgrade](#)
 - [Migrating from HSQL to MySQL](#)
 - [Upgrade the HSQL database](#)
- [Check or Update YARN Path Settings](#)
- [Retrigger the Search Index](#)
- [Recompile Plug-ins](#)
- [Validate the Upgrade](#)
- [Removing Extraneous Properties After Upgrade Adjustments](#)
- [How to Roll Back to a Previous Database Version](#)

Upgrading From an Older Release

Datameer customers can download updated software versions from <http://my.datameer.com>

To upgrade from an older release, you should first back up your data. Then you will need to upgrade the application and upgrade the database.

Upgrading to version 6.3

Parquet is the new default storage format

Datameer now uses [Parquet](#) as its default storage format as opposed to the previous sequence storage format. Sequence Files are still readable in Datameer but all new artifacts are written in Parquet. Depending on the configuration of your jobs, it might be difficult to roll back after upgrading to Datameer 6.3.

There is a property that forces Datameer to use Sequence Files instead of Parquet if a rollback is required. Under the `JAVA_OPTIONS` in `etc/das-env.sh`, add "`-Duse-sequence-storage=true`" before starting Datameer for the first time following the upgrade. This property has been removed as of Datameer 7.0.

If upgrading from a Datameer version before 6.1, we highly recommend first upgrading to Datameer 6.1 and [verifying that all job functions work without the MapReduce framework before upgrading to Datameer 6.3](#).

In 6.3, Datameer uses [Parquet](#) as its default storage format. While Datameer can still read data written in the sequence file format, all artifacts are now stored using the Parquet file format. Datameer still uses the sequence file format for writing intermediate files and preview files for performance reasons, but the final artifacts are saved using the Parquet format. Due to this change, it might not be possible to roll back after upgrading to Datameer 6.3, depending on the configuration of your artifacts. For example, if import jobs and workbooks are configured to purge historical data and only keep the last one result, the final artifacts are stored as Parquet files, and prior versions of Datameer cannot read them.

External authentication changes

As of Datameer 6.3, its' required to use ports 389/636 to connect to LDAP or the ports 3268/3269 to connect to Active Directory.

Upgrading to versions 7.0+

Ensure that Datameer application server, as well all data nodes, have Java 1.8 (Oracle recommended)

Upgrading to version 7.2

If you are upgrading to Datameer 7.2 from a 6.1.x or earlier version, must first upgrade to a 7.1.12 or later 7.1.x release, in order to trigger an important Database schema migration processes.

Upgrading with Kerberos

Keep the following in mind if you use Kerberos:

- If you used Kerberos prior to 5.11, you need to install the plug-in during upgrade.
- If you are running a Kerberos-secured cluster with impersonation enabled, you need to run the [secure_hdfs_tool.sh command line tool](#) when upgrading to versions 6 and above. For versions 6.1 and above, run the command after upgrading, starting, and stopping Datameer. Once you've run the command, restart Datameer.

Migrating workbooks and other artifacts when upgrading

Workbooks and other JSON files downloaded as a backup in older versions of Datameer are not supported in newer versions of Datameer.

When upgrading to a newer version of Datameer, ensure that all needed workbooks and files are part of the migration process so that they can be used in your new version of Datameer.

Upgrade Planning

- Before upgrading Datameer, a plan should be made for application downtime scheduling, notifications, and creating a maintenance window for Datameer. Please take note of the [system requirements](#) and consider upgrading the MySQL service for Datameer's application database from 5.1 to 5.5 or 5.6, if applicable.
- Backed up Datameer files need to be included in the migration process. Previous files that aren't part of the migration process aren't supported in newer versions.
- Jobs will need to be set to stop executing prior to the upgrade process. This could be done e.g. by [Pause Job Scheduler](#).
- Request possible assistance from the database administrator in case credentials for database are not available to the application owner.
- Custom plug-ins which were created by [Using the Plug-in SDK](#) of a former Datameer version need to be re-compiled!
- Since it is recommended to not copy old configuration files or scripts to the new location, note the changes you have made in the previous setup. This information you will need to make necessary changes in the new configuration files.
 - **As of Datameer 7.4**, a property file exists so Job Scheduler and Event Bus settings adjusted in the UI are read during Datameer start up.
 - A properties file called "*overrides.properties*" isn't written on Datameer but your systems HOME folder. (Path = *<home>/datameer/overrides.properties*)
 - This property file is auto-created when adjusting the Event Bus or Job Scheduler settings under the Admin tab. This file can also be manually added and edited.
 - This is the last properties file read on start up. (E.g., it overrides other property files like *default.properties* and *deployMode.properties*)
 - This file can be modified to allow for storing and the restoring of custom properties.

Disabling Housekeeping/Compaction services

In the event of a problem after a Datameer upgrade, [it is possible to roll back to the previous release](#). To allow for the possibility of a roll back and avoid any potential data loss, Datameer strongly recommends disabling the [Housekeeping](#) and [Compaction](#) services before beginning the upgrade process. Once upgrade validation is complete and the environment is considered stable, the Housekeeping and Compaction services can be re-enabled.

1. Open the *<Datameer new version installation directory>/conf/default.properties* file.
2. Disable Housekeeping by setting the property *housekeeping.enabled* to **false**.
3. Disable Compaction by setting the property *auto-compaction.enabled* to **false**.
4. Start Datameer and [perform upgrade validation testing](#).
5. Once the upgrade validated and is considered successful, turn Housekeeping and Compaction back on by setting the above mentioned properties back to **true**.

Restarting Datameer is required to apply these changes

Workbook Validation

A Workbook is a complex structure that Datameer is constantly striving to improve. Workbooks can be built in a near unlimited number of different combinations of Column Renames, Joins, Unions, Filters, and Functions - including user built functions. While we do attempt to cover all the possibilities while implementing our upgrade code, some customer development decisions are unpredictable and as a result our Workbook upgrade code can not always successfully transform all workbooks.

To validate whether any Workbook has been broken during an upgrade, Datameer created the Workbook Health Check feature. The tool reviews

a Workbook's structure and reports any logical issues. This tool is available in versions **6.4.14, 7.1.13, 7.2.13, 7.4.11+, 7.5.4+, and 10.0.1+**. Datameer's best practice recommendations for upgrades are as follows:

- Upgrade within the current branch to the version where the Workbook Health Check tool is available. For Example, if the current version is **6.4.13**, upgrade to **6.4.14**.
- Navigate `http://<Datameer host>:<port>/dev` (admin rights are required), then click on *Workbook Health Check* and press *Start button*.
- As soon as validation is complete (execution time varies based on the number and complexity of the existing Workbooks), you will receive a status summary and the broken Workbooks configuration IDs. The same information will also be written into Datameer *conductor.log* together with the error for every artifact. You might want to review broken Workbooks and fix the errors before the upgrade. Note, a session timeout will interrupt the workbook validation check.
- Upgrade to the desired Datameer version.
- Rerun the Workbook Health Check and ensure that there are no new broken items. In case there are Workbooks that have been broken by the upgrade, check the error messages contained in *conductor.log* and decide whether you could fix them manually or need to roll back and get Datameer support assistance on the issue.

Backup Keyfiles and Keystore

If you have [set up password encryption](#) and/or [enabled TLS with custom certificates](#), backup your Keystore and Keyczar keyfiles.

Upgrade the Application

If you update Datameer using user/group *root* it is recommended for security to go back and change permissions back to the user/group *datameer*.

Pausing jobs being submitted to the cluster

Before performing a [graceful shutdown of Datameer](#), [view the current running or queued jobs](#) and double-check that there is nothing running or scheduled.

1. Stop the application.

```
<Datameer Application Folder>/bin/conductor.sh stop
```

2. Unzip the upgrade file.

Move the MySQL JDBC driver located in `<Datameer path>/das-data/jdbc-jars`, as described into the [Install the JDBC Database Drivers](#) section of the [Installation Guide](#).

In case you are using Datameer with MySQL as an application database, please check that database mode is configured accordingly in your `etc/das-env.sh` file

```
export DAS_DEPLOY_MODE=live
```

3. If upgrading in Workgroup or Enterprise versions, users may want to consider allocating additional memory in `etc/das-env.sh`.

```
# Adjust max available memory (-Xmx) according to your needs
# WARNING: Path variables that may contain blanks should be added to jetty.sh start_das() method (see DAP-6342)
export JAVA_OPTIONS="-Xmx2048m -XX:MaxPermSize=384m -Xms256m -XX:MaxNewSize=448m
-XX:SurvivorRatio=6 -XX:+UseConcMarkSweepGC -XX:CMSInitiatingOccupancyFraction=80
-XX:+HeapDumpOnOutOfMemoryError -XX:+CMSClassUnloadingEnabled
-XX:+CMSPermGenSweepingEnabled"
export JAVA_OPTIONS="$JAVA_OPTIONS -Dfile.encoding=utf-8"
```

4. Consider changing the container sizing of the Map, Reduce, and AM containers, which respectively correspond to MapReduce and Tez jobs. To do so, change the settings of the following properties:

```
das.job.map-task.memory=2048
das.job.reduce-task.memory=2048
das.job.application-manager.memory=2048
```

5. If existing, copy the files from the `das-data` folder of the old distribution to the new location. Keep the original `das-data` information

as a backup.

```
cp -r /<old-location>/das-data /<new-location>/
```

6. Users need to check, and update if necessary, the folder where plug-in configurations are stored.

Open the file system of the previous Datameer version you are upgrading from and navigate to the `default.properties` file.

Search for the property file that defines the folder where plug-in configurations are stored.

```
# Defines the folder where to store plugin configurations
system.property.plugin.configs.dir=<folder name>
```

Open the file system of the upgrade version of Datameer and find the same property in the `default.properties` file.

Check or update the new property value to be the same folder name in the previous version.

Copy the contents of the previous plug-in configurations folder into the file system of the new upgraded version of Datameer.

7. If you made changes in your `conf/` directory previously (like changing `live.properties` and `log4j-<xyz>.properties`), you need to apply** the same changes to the new `conf/` directory of your upgraded instance.
8. Copy over the native libraries that you have added to Datameer (if this applies). You don't have to copy over the native libraries that are already bundled with Datameer.

```
cp -r /<old-location>/lib/native/* /<new-location>/lib/native/
```

9. If you have made changes to the `conductor.sh` script (for example, to enable SSH), apply** these changes to `/<new-location>/bin/conductor.sh` again. **Notice JAVA_OPTIONS has been moved to `etc/das-env.sh`**
10. Migrate the SSL values from the previous `start.ini` to the new one.

```
# Set up a demonstration keystore and truststore
jetty.keystore=etc/keystore
jetty.truststore=etc/keystore

# Sets the demonstration passwords
# OBF passwords aren't secure. They are only protected from casual observation.
# See http://www.eclipse.org/jetty/documentation/current/configuring-security-secure-passwords.html
jetty.keystore.password=OBF:1vny1z1o1x8e1vnw1vn61x8g1z1ulvn4 # storepwd
jetty.keymanager.password=OBF:1u2ulwml1z7slz7a1wn1lu2g # keypwd
jetty.truststore.password=OBF:1vny1z1o1x8e1vnw1vn61x8g1z1ulvn4 # storepwd
```

11. Copy over files from `etc/custom-jars` (these files could be database drivers or 3rd party libraries).
12. If you are using custom Datameer plugins (jar archives stored under `etc/custom-plugins`), update those to be in sync (API compatible) with the new version of Datameer and be sure to remove jar files related to older versions from `etc/custom-plugins`.

Do not copy the entire old `conf/` directory or `conductor.sh` script to the new location. You need only to make changes in the new file for the changes you made previously.

Database Upgrade

Follow the steps for upgrading your MySQL or HSQL database. **As of Datameer 7.4:** MariaDB is supported as an alternative to MySQL.

Upgrade the MySQL database

- As of Datameer 6.0, only versions MySQL versions 5.5 and above are supported.
- It is highly recommend to [backup the application database](#) before proceeding further with the upgrade process.
- If you are upgrading from an older release, you may need to migrate your database schema.

1. To upgrade the database, enter the following command:

```
bin/upgrade_db.sh
```

If either the structure of the MySQL database or any of the tables in the database will be modified by the Datameer upgrade, running this command automatically creates a dump of the database.

- Ensure that the script is not executed under root, but by the same user account that starts and stops Datameer.
- Run the upgrade script directly from the <INSTALLDIR>. Preface the script with the *bin* directory.

2. If your database name is different than the default `dap` or the database runs on a different server or customized port, you will need to provide the additional arguments:

```
bin/upgrade_db.sh -h [db hostname] -o [db port] -n [db name] -u [db  
username] -p [db password]
```

3. Check to ensure your database character encoding is set to UTF-8.
In the Database, enter:

```
SELECT DEFAULT_CHARACTER_SET_NAME FROM INFORMATION_SCHEMA.SCHEMATA  
WHERE SCHEMA_NAME = '<The name of your Datameer database>';
```

If UTF8 was returned, open the default properties and comment in the setting `system.property.hibernate.connection.characterEncoding=utf8`.

If something other than UTF8 was returned, change the database character encoding setting to UTF8 or [special characters may show as corrupt](#).

4. To finish the upgrade, restart the application.

```
<Datameer Application Folder>/bin/conductor.sh start
```

MySQL thread stack overrun during database upgrade

If you receive the above error message during an update, raise the MySQL stack size. For example,

```
mysqld -O thread_stack=192k
```

or higher if necessary.

Migrating from HSQL to MySQL

Datameer provides a tool to migrate data from an HSQL file database to a MySQL database:

1. Set-up MySQL as per the [Installation Guide](#)
2. Migrate HSQL database to MySQL

```
bin/migrate-db-tool.sh hsql-file:<Datameer user  
path>/Datameer/<version>/das-data/database mysql
```

* Ensure this script is being run from root Datameer installation folder.

Note

Ensure your users clear their browser caches after implementing a Datameer upgrade.

Upgrade the HSQL database

You have to use the script `bin/upgrade_hsql_db.sh` to upgrade your HSQL database to a higher version. Ensure this script is run from the root Datameer installation folder.

```
usage: upgrade_hsql_db.sh
  -n<new>, --new-das-data <folder>          new das-data folder
  -o<old>, --old-das-data <folder>         old das-data folder
```

With the `-pi` option you define the previous datameer installation. We need this folder to find the HSQL database we want to upgrade.

We backup your previous database into your current installation folder. Furthermore we use all upgrade scripts with name pattern `upgrade-<version>.hsql` in your current installation folder.

As an option you can use the parameters `-n<new>` and `-o<old>`. These parameters define the new/old location of your `das-data` folder. When you running datameer in LOCAL mode it could be possible that you want to copy your `das-data` folder to a new location. A common use case is when the `das-data` folder is saved within the installation directory.

Here is an example of what it looks like when upgrading a database.

```
bin/upgrade_hsql_db.sh --old-das-data <file path>/<file version>/das-data
--new-das-data <file path>/<file version>/das-data
```

```
#Detect databases
detect old database
[/Users/marko/Desktop/Datameer-2.0.1-0.20.2/das-data/database/hsql-db]
detect new database
[/Users/marko/Development/Java/datameer/dap/build/dist/Datameer-2.0.2-0.20.2/das-data/database/hsql-db]

Upgrade datameer database version [2.0.0.6] to version [2.0.2.0]

#Assert datameer is running
No

#Copy Das Data
copy das-data from [/Users/marko/Desktop/Datameer-2.0.1-0.20.2/das-data] to
[/Users/marko/Development/Java/datameer/dap/build/dist/Datameer-2.0.2-0.20.2/das-data]

#Upgrade starts...
Process upgrade scripts on database
jdbc:hsql:file:das-data/database/hsql-db
upgrade-2.0.1.hsql

#Update schema-version
set system.schemaVersion on database
```

```
[/Users/marko/Development/Java/datameer/dap/build/dist/Datameer-2.0.2-0.20.2/das-data/database/hsqldb] to [2.0.1]
```

```
#Update data uri's [/Users/marko/Desktop/Datameer-2.0.1-0.20.2/das-data] -> [/Users/marko/Development/Java/datameer/dap/build/dist/Datameer-2.0.2-0.20.2/das-data]
```

```
log4j:WARN No appenders could be found for logger  
(org.hibernate.type.BasicTypeRegistry).  
log4j:WARN Please initialize the log4j system properly.  
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for  
more info.
```

```
Converting 12 data entries
```

```
file:/Users/marko/Desktop/Datameer-2.0.1-0.20.2/das-data/fileuploads/1/1 ->  
file:/Users/marko/Development/Java/datameer/dap/build/dist/Datameer-2.0.2-  
0.20.2/das-data/fileuploads/1/1  
file:/Users/marko/Desktop/Datameer-2.0.1-0.20.2/das-data/fileuploads/2/2 ->  
file:/Users/marko/Development/Java/datameer/dap/build/dist/Datameer-2.0.2-  
0.20.2/das-data/fileuploads/2/2  
file:/Users/marko/Desktop/Datameer-2.0.1-0.20.2/das-data/fileuploads/3/3 ->  
file:/Users/marko/Development/Java/datameer/dap/build/dist/Datameer-2.0.2-  
0.20.2/das-data/fileuploads/3/3  
file:/Users/marko/Desktop/Datameer-2.0.1-0.20.2/das-data/fileuploads/4/4 ->  
file:/Users/marko/Development/Java/datameer/dap/build/dist/Datameer-2.0.2-  
0.20.2/das-data/fileuploads/4/4  
file:/Users/marko/Desktop/Datameer-2.0.1-0.20.2/das-data/fileuploads/5/5 ->  
file:/Users/marko/Development/Java/datameer/dap/build/dist/Datameer-2.0.2-  
0.20.2/das-data/fileuploads/5/5  
file:/Users/marko/Desktop/Datameer-2.0.1-0.20.2/das-data/workbooks/6/6 ->  
file:/Users/marko/Development/Java/datameer/dap/build/dist/Datameer-2.0.2-  
0.20.2/das-data/workbooks/6/6  
file:/Users/marko/Desktop/Datameer-2.0.1-0.20.2/das-data/workbooks/7/7 ->  
file:/Users/marko/Development/Java/datameer/dap/build/dist/Datameer-2.0.2-  
0.20.2/das-data/workbooks/7/7  
file:/Users/marko/Desktop/Datameer-2.0.1-0.20.2/das-data/workbooks/8/8 ->  
file:/Users/marko/Development/Java/datameer/dap/build/dist/Datameer-2.0.2-  
0.20.2/das-data/workbooks/8/8  
file:/Users/marko/Desktop/Datameer-2.0.1-0.20.2/das-data/workbooks/9/9 ->  
file:/Users/marko/Development/Java/datameer/dap/build/dist/Datameer-2.0.2-  
0.20.2/das-data/workbooks/9/9  
file:/Users/marko/Desktop/Datameer-2.0.1-0.20.2/das-data/fileuploads/10/10  
->  
file:/Users/marko/Development/Java/datameer/dap/build/dist/Datameer-2.0.2-  
0.20.2/das-data/fileuploads/10/10  
file:/Users/marko/Desktop/Datameer-2.0.1-0.20.2/das-data/workbooks/11/11 ->  
file:/Users/marko/Development/Java/datameer/dap/build/dist/Datameer-2.0.2-  
0.20.2/das-data/workbooks/11/11  
file:/Users/marko/Desktop/Datameer-2.0.1-0.20.2/das-data/fileuploads/10/12
```

->

```
file:/Users/marko/Development/Java/datameer/dap/build/dist/Datameer-2.0.2-0.20.2/das-data/fileuploads/10/12
```

Check or Update YARN Path Settings

Check or update YARN application classpath and LD_LIBRARY path. The path settings are used to lookup and load libraries , therefore incorrect path settings may lead into unexpected behavior .

1. Start Datameer
2. Open the Administration tab
3. Open the Hadoop Cluster options
4. Check or update YARN application classpath and LD_LIBRARY path

Cluster Settings

YARN Resource Manager Address: *

master:8032 host:port (Default port is 8032)

YARN Resource Manager Webapp Address: *

http://master:8088 http://host:port (Default port is 8088)

YARN Resource Manager Scheduler Address: *

master:8030 host:port (Default port is 8030)

YARN MR Job History Address: *

master:10020 host:port (Default port is 10020)

YARN Application Classpath:

`$SHADOOP_CONF_DIR, $SHADOOP_COMMON_HOME/*, $SHADOOP_COMMON_HOME/lib/*, $SHADOOP_HDFS_HOME/*, $SHADOOP_HDFS_HOME/lib/*, $SHADOOP_MAPRED_HOME/*, $SHADOOP_MAPRED_HOME/lib/*, $YARN_HOME/*, $YARN_HOME/lib/*`

Use the 'yam.application.classpath' property out of your yarn-site.xml or execute 'yam classpath' on the command line.

Max Concurrent Jobs: *

25 Number of simultaneous jobs that may be run including imports/exports/workbooks etc.

Amazon EC2 Cluster? Only applies if your hadoop cluster is hosted on EC2, but Datameer is outside of EC2.

Hadoop Properties

Default Hadoop Properties:

```
# properties for YARN distributions
## max = 5GB (s3n), min = 16MB
mapreduce.input.fileinputformat.split.maxsize=5368709120
mapreduce.input.fileinputformat.split.minsize=16777216

## set compression without specifying codec
mapreduce.output.fileoutputformat.compress=true
mapreduce.output.fileoutputformat.compress.type=BLOCK
```

Default Hadoop properties.
[Reset to default value](#)

Hadoop Distribution Specific Properties:

```
# for HDP Ambari installations
# please set "hdp.version" to the installed version
#
# hdp.version=2.4.0.0-169
# mapreduce.application.framework.path=/hdp/apps/${hdp.version}/mapr
# mapreduce.application.classpath=/usr/hdp/${hdp.version}/hadoop/conf:
```

Properties specific to hdp-2.4.0
[Reset to default value](#)

Custom Properties:

```
tez.am.launch.env=LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/hdp/
tez.task.launch.env=LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/hdp/

mapreduce.application.classpath=
mapreduce.application.framework.path=

#das.fileupload.originalfile.codec=org.apache.hadoop.io.compress.Gzip
#das.execution-framework=sparkcluster
```

Customized Properties. One key/value pair per line. Multiple values are comma separated. E.g. key=value1,value2
[See About custom properties.](#)

Example:

When upgrading from CDH 4.x to CDH 5.x the classpath was renamed.

```
$HADOOP_CONF_DIR, $HADOOP_COMMON_HOME/*, $HADOOP_COMMON_HOME/lib/*,
$HADOOP_HDFS_HOME/*, $HADOOP_HDFS_HOME/lib/*, $HADOOP_MAPRED_HOME/*,
$HADOOP_MAPRED_HOME/lib/*, $YARN_HOME/*, $YARN_HOME/lib/*
```

to

```
$HADOOP_CONF_DIR, $HADOOP_COMMON_HOME/*, $HADOOP_COMMON_HOME/lib/*,
$HADOOP_HDFS_HOME/*, $HADOOP_HDFS_HOME/lib/*, $HADOOP_MAPRED_HOME/*,
$HADOOP_MAPRED_HOME/lib/*, $HADOOP_YARN_HOME/*, $HADOOP_YARN_HOME/lib/*
```

The correct YARN classpath can be found for the Datameer client via command:

```
yarn classpath | tr ':' ','
```

Retrigger the Search Index

INFO

The search settings to enable file search within the 'File Browser' must be **retriggered manually** after the upgrade.

Recompile Plug-ins

Custom plug-ins which were created by [using the plug-in SDK](#) need to be recompiled. Adjust the [versioning](#) and [create the plug-in again](#) under the new plug-in SDK.

All plug-ins used that aren't a part of standard distribution, e.g., Advanced Governance package, require an upgrade.

1. Check if any plug-ins are located in *<OLD Datameer version installation folder>/etc/custom-plugins/*.
2. Request a Datameer representative to provide these plug-ins recompiled for the Datameer version you are upgrading to.
3. Copy the new versions of the plugins to *<NEW Datameer version installation folder>/etc/custom-plugins/*.

Validate the Upgrade

After the upgrade has completed, perform a test to validate that the upgrade was successful.

1. Start Datameer.
2. If applicable, upload recompiled plug-ins.
3. Validate the upgrade by performing tests with baseline jobs that could previously be run without errors.
4. If applicable, test custom plug-ins.
5. Create a connection where the data is stored, import the data, add and modify the data in a workbook, and run the data.

If no unexpected errors occur the upgrade has been validated.

Attention

- Fields that were made unnecessary on the AD/LDAP configuration page have been removed.
- There is no longer a radio button needed to switch between AD and LDAP.
- The difference between AD and LDAP can be configured in field input.
 - Username Attribute: uid (LDAP), sAMAccountName (AD)
 - Group Definition: objectClass=groupOfNames (LDAP), objectClass=group (AD)
- There is no longer a drop down box to switch between simple and custom group definition. (LDAP only)

Removing Extraneous Properties After Upgrade Adjustments

The support for MapReduce has been deprecated and was removed in [Datameer 6.3](#). After it is verified that all jobs work without the MapReduce framework, remove MapReduce related custom properties from the Hadoop cluster page under the Administration tab and in the advanced settings of data links, import jobs, and/or workbooks.

How to Roll Back to a Previous Database Version

When upgrading, a database dump is created by Datameer, the file name is similar to `mysql_db_dump_<oldVer>.sql`. If an upgrade failed (failing upgrade scripts / failing upgrade injectors during startup), the database paths might be conflicting, it is therefore necessary to restore the Datameer MySQL database dump. Follow the steps to complete this task.

1. Drop the new database, for example on the database server or the MySQL CLI.

```
DROP DATABASE dap;
```

2. Using the `mysql-init.sql` script, create a new database.

```
mysql [-h <dbhost>] -u root -p < bin/mysql-init.sql
```

3. Import and restore the previous database using the following shell command.

```
mysql [-h <dbhost>] -u root -p dap < /path/to/dump_file_name
```