

Data manager

On this page:

- [Running the Data Manager](#)
- [Data Manager commands](#)
 - [Deleting selected resource](#)
 - [Deleting selected user](#)
 - [Deleting selected branch](#)
 - [Deleting revisions](#)

Teamwork Cloud is equipped with a Data Manager for administrative processes. You need to download the data manager .zip file and extract it to the machine where you installed Cassandra. Before using Data Manager, we recommend that you [backup your database](#) in the same manner as you do before using the migration tool. You can download the Data Manager tool from the same location you download the migration tools for the Cassandra database and UML meta-model.

Running the Data Manager

To run Data Manager

1. Stop the Teamwork Cloud server.

 If the command-line migration tool is running, it needs to be stopped as well.

2. Make sure Cassandra is up and running.

 The ports through which Data Manager is connected to Cassandra are **9160** and **9042**.

3. Download **DataManager.zip**.

 Be sure to only use Data Manager distributed with version 2021x Refresh2 Hot Fix 5.

4. Extract the zip file to the machine where you installed Cassandra.
5. On the command line, change the directory to the DataManager folder.
6. Execute the following command on Linux to display all available commands:

```
./datamanager -help
```

7. Run the desired command on Linux, for example, to show the user list:

```
./datamanager -lu
```

 The Data Manager will connect to Cassandra on localhost using anonymous authentication. If your Cassandra node is configured to bind to a specific IP address (by default ports 9160 and 9042 bind to all interfaces), or if you have configured Cassandra to use an authenticator other than

 The log file of Data Manager appears in the following directory:

```
<user folder>/twcloud/2021x/datamanager.log
```

Data Manager commands

The following table outlines the commands you can use in Data Manager to manage your data.

Server tool command	Function
-help	Displays Help information.

-dk, --drop-keyspace <keyspace=value> --silent-drop	Drops the 19.0 keyspace to free up the storage space after the database migration from 19.0 to 2021x.
	<p> • You can add --silent-drop to run the command silently, i.e., without requesting user permission.</p>
-dr,--delete-resource <resource ID>	<p>Deletes the given resource(s) and all of its related information. Specify multiple resource IDs using space as a separator.</p> <p> If you delete a synchronized resource from the target server, the resource will be removed together with its remote resource configuration allowing you to create new configurations with the same resource from the source server.</p>
-drb,--delete-branch <resourceID=value branchID=<ID1>,<ID2> includeChildBranch=<true/false>>	<p>Deletes the given branch(es) and all of its related information. Specify multiple branch ID using a comma as a separator.</p> <p>includeChildBranch has two values: true and false.</p> <ul style="list-style-type: none"> • When the includeChildBranch value is set to false, and the branch has child branches, this branch will not be deleted. • When the includeChildBranch value is set to true, the branch including its child branches will be deleted. <p> This command does not work for the branches of a synchronized resource. You can delete only the whole synchronized resource, but not its branches.</p>
-drd,--delete-marked-deleted-resources	<p>Deletes all resources that were marked as deleted.</p> <p> If you delete a synchronized resource from the target server, the resource will be removed together with its remote resource configuration allowing you to create new configurations with the same resource from the source server.</p>
-drr,--delete-revision <resourceID=value revision=<revision1,revision2> includeChildRevision=<true/false>>	<p>Deletes the given revision(s) and all of its related information. Specify multiple revisions using a comma as a separator.</p> <p>includeChildRevision has two values: true and false.</p> <ul style="list-style-type: none"> • When the includeChildRevision value is set to false, and the revision is not the latest, it will not be deleted. • When the includeChildRevision value is set to true, the upward revisions and branches within it from the selected version will be deleted. <p> This command does not work for the revisions of a synchronized resource. You can delete only the whole synchronized resource, but not its revisions.</p>
-du,--delete-user <username>	<p>Deletes the given username(s) and all of its related information. Specify multiple usernames using comma as a separator.</p> <p> If the username consists of two words (for example, dan smith) you must add double quotation marks (" ") before and after the username, as shown in the following example:</p>
-rrof, --remove-read-only-flag <resourceID=<value1>,<value2> --categoryID=<value1>,<value2>>	<pre>Re Sp --delete-user "dan smith" -du "dan smith" c 3d5a6e675714 --categoryID=a1d569ea-fe6a-402f-8a25-aef6b932de48</pre> <p></p>
-lb,--list-branch <resourceID=value>	Lists all branches of the given resource.

-lc,--list-cluster-ids	<small>Removes the read-only flag from periodically synchronized resources or categories. This will cause version inconsistencies after subsequent synchronizations.</small>
-li,--list-index	Lists all available data that have indexes.
-lr,--list-resources	Lists all available resources.
-lrd,--list-marked-deleted-resources	<small>Lists all resources that are marked as deleted to allow removal at the synchronization operation and you want to make revisions in the given branch.</small>
-lrv,--list-revision <resourceID=value branchID=value>	Lists all revisions in the given branch.
-lu,--list-users <--no-info>	Lists all usernames and their information, sorted by name.  You can add --no-info to show only the usernames.
-ram,--reset-administrator	Resets admin password, status, and permissions.  • If there is no Administrator account, it will be created. • If the Administrator account exists in the database:
-rc,--recreate-cluster-id	Recreates a cluster ID.
-ri,--rebuild-index <data=value1,value2>	<small>Rebuilds all indexes for the given data and uses the data=* option to rebuild all indexes of all available data.</small> If the account is for an external user, it will be reset to an internal user with a new password. This user will be shown in the Teamwork Cloud Admin console.
-urd,--unmark-deleted-resources <resource ID>	<small>Unmarks the given resource(s) previously marked as deleted. Specify the resource ID using space as a separator.</small> the user will be shown in the Teamwork Cloud Admin console.
-vb,--validate-blob <savePath=value onlyInconsistent=value --readAllWithCQL --createCompareOfThriftAndCQL>	Validates blob information. • The default roles of the Teamwork Cloud Admin will be restored to make sure that this account can fully operate as the default administrator.
-ve,--validate-eobject <resourceID=value1,value2 saveDir=value onlyInconsistent=value>	Validates the eObject information of a specific resource ID and uses the resourceID=* option to validate all resources.
-version	Displays the version of Data Manager.
-vi,--verify-index <data=value1,value2>	Verifies all indexes for the given data and uses the data=* option to verify all indexes of all available data.
-vr,--validate-rbac <savePath=value onlyInconsistent=value>	Validates a user, user group, role, and role assignment information.
-vrc,--validate-resource-commit <savePath=value onlyInconsistent=value>	Validates a metamodel, branch, and committed information.
-col,--cleanup-object-locks	Removes released object locks (that are no longer used) from the database.  How to safely execute this command: 1. Before performing object lock cleanup make sure that all Cassandra nodes are running and accessible. 2. After Data Manager finishes cleaning locks, run the command in a separate terminal nodetool repair which you can find in the <i>Cassandra/bin</i> directory. Once nodetool repair is done, return to Data Manager and confirm operation to proceed. 3. DataManager will ask to run command in a separate terminal nodetool compact . After running the command in nodetool compact terminal, return to Data Manager and confirm operation to proceed.

The **delete** command to delete resources and users works as follows.

Deleting selected resource
<pre>-dr,--delete-resource <resource ID> <resource ID> ...</pre>

These commands delete a selected resource and all of the following resource-related information:
and confirm operation to proceed.

- Tags information of the resource.
- Category information (the resource will be removed from all categories).
- Branch information.
- Commit information.
- Role assignment information (role assignment information related to the resource will be removed).
- Locking information.

Deleting selected user

```
-du, --delete-user <username>,<username> ...
```

This command deletes a selected user and all of the following user-related information:

- Role assignment information (all role assignment information belonging to the user will be removed).
- Session (the existing user session will be removed).
- Lock information (model elements in projects locked by the user will be unlocked).
- User group information (the user will be removed from all user groups).



In the modeling tool's history panels, the word *deleted* will appear in brackets next to the deleted users.

Deleting selected branch

```
-drb, --delete-branch <resourceID=value branchID=<ID1>,<ID2> includeChildBranch=<true/false>>
```

The branch-related information which will be removed together with the branch includes:

- Tags information of each revision in the branch
- Object information created/modified at each revision in the branch
- Commit information of each revision in the branch
- Locking information of each object locked in the branch
- Branch information of the branch
- If the branch has child branches, depending on the **includeChildBranch** value, the branch will be deleted or not:
 - When the **includeChildBranch** value is set to *False* and the branch has child branches, then this branch will not be deleted.
 - When the **includeChildBranch** value is set to *True*, then even if the branch has child branches it will be deleted.

Deleting revisions

```
-drr, --delete-revision <resourceID=value revision=<revision1,revision2> includeChildRevision=<true/false>>
```

The revision-related information which will be removed together with the revision includes:

- Tags information of the revision
- Object information created/modified at the revision
- Commit information of the revision
- Locking information of objects whose id's do not exist anymore in the branch that the revision belongs to
- Branch information of the branch if the revision was the first revision of the branch
- If a selected revision is not the latest, depending on the **includeChildRevision** value, the revision will be deleted or not:
 - When the **includeChildRevision** value is set to *False*, and the revision is not the latest, then it will not be deleted.
 - When the **includeChildRevision** value is set to *True*, then revisions upward and branches within it from the selected version will be deleted.

Related pages

- [Backup and restore data procedures](#)
- [Migrating data and upgrading Teamwork Cloud](#)