

# Permissions

## User Permissions

Qube! contains a permissions mechanism to allow administrators to control user access. The permissions are broken down into classes:

- Administrative
- Submission
- Job Control

The default permission set is configured by `supervisor_default_security` in the Supervisor's `qb.conf` file to prevent remote administrators from modifying it.

Each permission must be specified per user. Users not listed in the table will use the default permission set.

Individual and group user permissions are specified using the user access list which is modified from the [QubeGUI User Permission View](#) or by using the `qbusers` command.

In the WranglerView, the user groups are separated from the user by a ":".

Refer to [qbusers](#) for commandline examples.

## Group Permissions

Qube! user groups are designed to collect a set of users into a kind of partition where each user can only control other jobs owned by users in the same group. Each user in the group then can be granted permissions that only affect the jobs of other group members.

Qube! groups might be put to use at a facility that has a team of people responsible for managing the day-to-day workload on the compute farm, but only need to have the ability to control the jobs of other team members. Instead of granting full administrator privileges, the team would be placed in a group.

A group can be managed by either a actual Qube! administrator, or by the group's administrator. However, the group administrator is not allowed to add users to the group, they are only allowed to modify the permissions of already existing users.

User permissions take precedence over group permissions. For example, if a user doesn't have permission to kill a job, even if their group permissions allow it, the user still won't be able to kill a job. A user with "administrator" or "impersonate" privileges is not subject to this restriction.

## A common group permission scenario: set of users with a "group" administrator

You can use the Group permissions to create a set of users, and then make only certain users have the admin rights to this restricted set of users.

The trick is to add the user to the new group with **no** permissions. You'll end up with a user of the same name (but not in the group) with the same set of permissions as the `[default]` permission set. Then, grant the privileged user in that group the `admin` privilege.

The cmdline-equivalent is:

```
jburk-15-mbPro:~ root# qbusers --add --group showA foo jburk  ## neither
user was explicitly granted privileges before this
modified group permissions
```

```
jburk-15-mbPro:~ root# qbusers --list
total 8
---l jc- krmpbuicseyqg-vftn [default]
asil jcg krmpbuicseyqgpvftn administrator
---- --- ----- bar
---l jc- krmpbuicseyqg-vftn foo [default]
---l jc- krmpbuicseyqg-vftn jburk [default]
asil jcg krmpbuicseyqgpvftn qube
asil jcg krmpbuicseyqgpvftn qubesupe
asil jcg krmpbuicseyqgpvftn root
asil jcg krmpbuicseyqgpvftn system
```

```
jburk-15-mbPro:~ root# qbusers --member
group total 8
[default]
-- ----- default

foo
-- ----- showa

jburk
-- ----- showa
```

Notice that both users "jburk" and "foo" in the "showA" group have no "group" privilege. Both users still still have exactly the same privileges as if they weren't explicitly granted privileges; they can't modify each other's jobs.

Then grant the 'admin' privilege to 'jburk', but only in the "showA" group, that user's privileges remain unchanged outside of that group.

```
jburk-15-mbPro:python jburk$ qbusers --group showa --add --admin jburk
```

```
jburk-15-mbPro:~ root# qbusers --list
total 8
---1 jc- krmpbuicseyqg-vftn [default]
asil jcg krmpbuicseyqgpvftn administrator
----- bar
---1 jc- krmpbuicseyqg-vftn foo [default]
---1 jc- krmpbuicseyqg-vftn jburk [default]
asil jcg krmpbuicseyqgpvftn qube
asil jcg krmpbuicseyqgpvftn qubesupe
asil jcg krmpbuicseyqgpvftn root
asil jcg krmpbuicseyqgpvftn system
```

```
jburk-15-mbPro:~ root# qbusers --member
group total 8
[default]
-- ----- default
```

```
foo
-- ----- showa
```

```
jburk
a- ----- showa
```

```
jburk-15-mbPro:~ root# qbusers --group showA --drop foo
modified group permissions
```

```
jburk-15-mbPro:~ root# qbusers --list
total 7
---1 jc- krmpbuicseyqg-vftn [default]
asil jcg krmpbuicseyqgpvftn administrator
----- bar
---1 jc- krmpbuicseyqg-vftn jburk [default]
asil jcg krmpbuicseyqgpvftn qube
asil jcg krmpbuicseyqgpvftn qubesupe
asil jcg krmpbuicseyqgpvftn root
asil jcg krmpbuicseyqgpvftn system
```

## Common Permission Actions

### Users

#### List users and their permissions:

WranglerView: See the Users Layout for a hierarchical list

Command-line: `qbusers --list`

#### Add one or more users and assign permissions:

WranglerView: In User Layout, right-click and select "Add New User"

Command-line: (Use format for `--add` under changing a user's permissions)

**Change a user's permissions:**

WranglerView: In User Layout, select user, right-click "Change Permissions"

Command-line (Add Permission): `qbusers --add permission [permission,...] user [user,...]`

Note: the "--all" option will add all non-admin user permissions for the user

Command-line (Remove Permission): `qbusers --delete permission [permission,...] user [user,...]`

**Delete a user:**

WranglerView: In User Layout, select user, right-click "Remove User/Group"

Command-line: `qbusers --drop user [user,...]`

## Groups

**List group membership:**

WranglerView: See the Users Layout for a hierarchical list

Command-line: `qbusers --member`

**Add User to a Group:**

WranglerView: In User Layout, select user, right-click "Add User Group"

Command-line: `qbusers --add --group group user [user,...]`

**Remove User from a Group:**

WranglerView: In User Layout, select user group, right-click "Remove User/Group"

Command-line: `qbusers --drop --group group user [user,...]`

**Change permissions to the user group:**

WranglerView: In User Layout, select user group, right-click "Change Permissions"

Command-line (Add Permission): `qbusers --group group --add permission [permission,...] user [user,...]`

Command-line (Remove Permission): `qbusers --group group --delete permission [permission,...] user [user,...]`

## See Also

[WV User Permissions](#)

[qbusers](#)