Service Mode vs. Desktop User Mode

The Qube! worker can run either as a daemon (a "system service" on Windows), or a user process. When it runs as a daemon, it's said to be running in **service mode**. When it runs as a user process, it's said to be running in **Desktop User mode**.

Service mode:

- The worker process is usually started at system boot time and runs as long as the system is up
- The worker process runs as either a Windows service or a daemon owned by the root user on OS X and linux.
- The worker process will run jobs under a user other than root or the system service. This user is determined by the proxy_execution_mode value:
    - `proxy_execution_mode = proxy`
      means it will always authenticate as the user defined in proxy_account.
    - `proxy_execution_mode = user`
      means it will always authenticate as the user who submitted the job.

> ⚠ The worker process will be unable to access screenspace on Windows and OS X; no processes will be able to render to a hardware buffer, applications that can only run by displaying their full GUI will usually not be able to start, etc.

Desktop User mode:

- The worker process is started by a logged in user, usually when that user logs in, and is killed when the user logs out
- The worker process is owned by the logged in user
- The worker process does not re-authenticate, and all jobs are run by the user who owns the worker process
- The worker process has full access to the screenspace

Worker Authentication in Service mode

When the Worker launches a remote job as dispatched by the Supervisor, it can potentially create several processes, all controlled by the Worker. Since Qube is designed to emulate a user executing jobs on a remote host, the Worker will have to run these processes as some user.

Unix & OS X

Under Unix-based operating systems, the Worker does a setuid in order to switch user identities before starting the process. One implication of this is that the user's shell environment is *not* set up (tcshrc or equivalent is not run), so the job's environment may not be identical to an actual login by that user.

Windows

On Windows, this is handled quite differently. Under Windows, any process attempting to impersonate another user will have to provide certain security information, including the user's encrypted password.

To enable the Worker to provide this information every time it creates a new process, the information is stored in the worker's configuration, either in the worker's local qb.conf or the central worker configuration file stored on the supervisor.

> ⚠ **Considerations when running "User" mode on Windows**
> The Worker creates an authentication token when it creates a new process, and the Windows Operating System then permits the process access to the user's environment and files. For this reason, if running with **`proxy_execution_mode = user`** on Windows, every Windows user will need to register (and keep current) their password with Qube! Most facilities choose **`proxy_execution_mode = proxy`** when running in service mode on Windows.

Authenticating on Windows using an Active Directory Domain account

It's possible to use an AD account as the proxy account; this allows the proxy account to be added to the ACL's on networked filesystem, be centrally managed, etc.

You set the proxy_account to the user name, but without the Active Directory portion, and then specify the Active Directory domain name with worker_host_domain.

For example, if you have a domain account that I would login as MyCorp\someuser, where the AD domain name is "MyCorp", and the user's name is "someuser", you would configure it as follows:

- `proxy_account = someuser`
- `worker_host_domain = MyCorp`

These values can be set either in the worker's local qb.conf, but a best practice is to set in the supervisor's central worker configuration file.

See Also

How to switch a worker from Service mode to Desktop User mode

proxy_account

proxy_password

proxy_execution_mode

worker_host_domain