

NIMBY

"NIMBY" -- "Not In My Back Yard" -- controlling access to user-managed desktop blades.

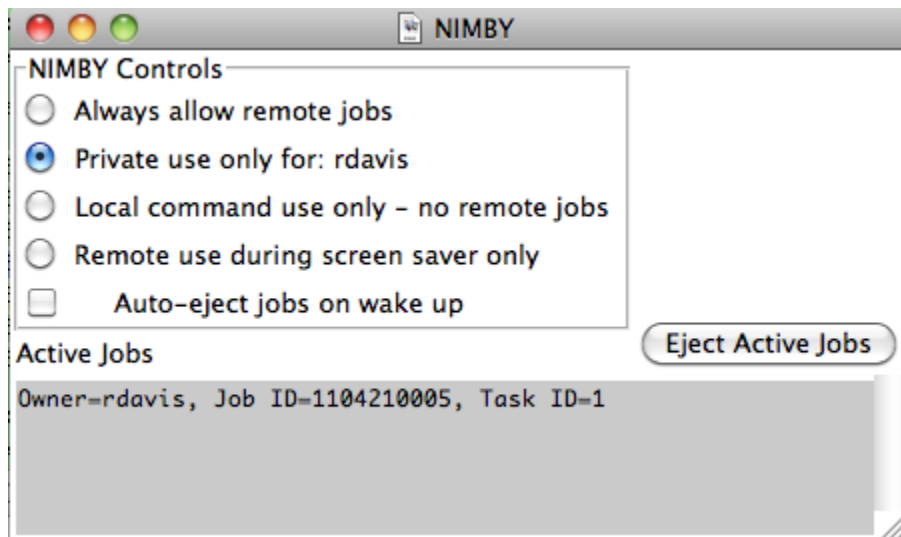
Tractor allows user desktop machines to be used as part of the site-wide renderfarm simply by running the tractor-blade server program on them, just like on more typical server hosts. The blade process can be started automatically as a system service or manually by users themselves, then the blade registers itself automatically with tractor-engine.

However, it is not always desirable to allow all jobs from the site job queue to run on every machine, especially desktop machines. Tractor provides several levels of control over which jobs are allowed to run on a given blade.

In some cases the host is just not a good match to run certain **types** of jobs. Some jobs will require special software to be installed on target blades, or they may require a certain minimum amount of install RAM or disk. These are the cases where the "service key" job scripting system is used to match job requirements to server capabilities. For more details on this type of restriction, see the job scripting discussion of "RemoteCmd -service" keys, and the blade profile discussion of "Provides" lists.

Blade profile definitions, in blade.config, can also define which **job owners** have access to a given set of blades. The "Access" parameters in a blade profile define these user restrictions for *all* blades that are members of that profile. This generally the right level of control for typical farm machines.

In contrast, a given blade's NIMBY settings are often used to further restrict the set of jobs that will run on that specific host. Also, unlike blade.config profile settings, the NIMBY settings can be changed dynamically by users or wranglers through the main dashboard or the desktop nimby application. The most common form of NIMBY restriction for desktop blades is limit them to accepting work only from jobs that were *spooled* from that same host, that is from the user sitting at that desktop.



Running the NIMBY application

- **macOS**

The *nimby* application resides in /Applications/Pixar/tractor-blade-\$version directory. Start the application by running /Applications/Pixar/tractor-blade-\$version/nimby`

- **Linux**

The *nimby* application resides in /opt/tractor-blade-\$version directory. Start the application by running /opt/pixar/tractor-blade-\$version/nimby`

- **Windows**

The *nimby.bat* batch file resides in C:\Program Files\Pixar\tractor-blade-\$version directory. Start the application by running C:\Program Files\Pixar\tractor-blade-\$version\nimby.bat

Program Arguments

The NIMBY application responds to the following command line arguments:

General Options

Option	Description
-h, --help	Show the help message and exit
--monitor=MONITOR	Tractor monitor hostname and port (host:port). Default (tractor-engine:80)

-P MPORT	Port on which the tractor daemon(s) are listening Default: 80
-L BPORT	Port on which blade listens for admin queries Default: 9005
--nimby=NIMBY	Restrict jobs that the blade will execute; '1' to only accept jobs spooled from this blade's host, '0' to always allow remote jobs, 'private' to only accept remote or local jobs for current user, and 'screensaver' to only accept remote jobs when the screen saver is active.
--querytime=JOBQUERYTIME	The query interval (in seconds) that nimby will use when querying the blade for running jobs. The default is 5 seconds.
--user=USER	Tractor user to login. Default: current computer logged in user.
--pass=PASSWD	Password for tractor user to login. Default: None.
--configfile=CONFIGFILE	File containing login and password information. Default: None.

Logging Options

Option	Description
-v, --verbose	Set logging level to INFO and above.
--debug	Set logging level to DEBUG and above.
--warning	Set logging level to WARNING and above. This is the default logging level.
-q, --quiet	Set logging level to CRITICAL only.
--logfile=LOGFILE	Define a local file for logging. Default logging is to stdout. If a logfile is specified, the Python logger will use a rotating file handler, which allows the log file to be rotated at various intervals.

Operational Modes

The NIMBY UI supports 4 modes of operation:

- **Always allow remote commands**

When checked, the local blade will accept remote or local commands from any job which matches normal service key and crew based requirements.

- **Private use for: *username***

When checked, the local blade will accept remote or local commands for the named user.

- **Local command use only - no remote jobs**

When checked, the local blade will only accept local commands. Remote commands, even from the local user will be rejected.

- **Remote Use during screen saver only**

Similar to local command use only, above, this mode will change to accept all remote commands when the screen saver is operational. This allows the blade to be available for work when the machine is considered to be idle.

On Linux the last checkbox will read **Remote use after 5 minutes idle**. It is difficult on Linux to determine if a screen saver is running, so instead the X11 server is queried for idle time. By default, if the mouse has been idle for 5 minutes, the UI will change the NIMBY mode to accept all remote or local commands.

Ejecting Active Jobs

If there are one or more active commands running on the local blade, they will be detailed in the *Active Jobs* box in the UI. If the operational mode is changed above to a mode which will not allow access, the current jobs will be allowed to finish, but no new tasks will be picked up by the blade.

Clicking on the *Eject Active Jobs* button will cause the blade to message the engine to kill the currently running task, and to re-queue it.

NIMBY Profile Settings

By default, all Tractor blades will accept work from any job owner, assuming that service keys match and other dispatching criteria are met. The optional "NIMBY" entry in a profile definition can be used to limit the jobs that will reach blades using that profile:

"NIMBY": 1

If you add "NIMBY": 1, to a blade's profile it will only accept *local* jobs spooled from that blade's host. Setting it to 0 restores the default behavior, described above.

"NIMBY": "username"

The "NIMBY" setting can also accept a user name as a string. In so doing, all blades matching that profile will only accept jobs from the named user.

Note: This parameter is at the "top" level of a profile definition, is *not* placed in the "Access" dictionary. Also note that this value can be changed for an individual blade via the blade list context menus in the Tractor Dashboard.

Setting NIMBY

In addition to using blade profiles, there are two other ways to set the NIMBY state:

- --nimby=foo can be added to the tractor-blade command line options at startup. This method will always override any setting in a blade profile, even after a profile reload.
- A new NIMBY change request can be sent to the engine [via URL](#), just as the Tractor Dashboard does with the blade context menu. The engine contacts the blade to implement the change. The URL values can be:
 - **1** - will only accept jobs spooled from that blade's host.
 - **username** - will only accept jobs from the named user.
 - **0** - will accept jobs from anyone
 - **default** - reverts the blade to profile settings or its state at startup via command line options

Values sent via URL are sticky and persist until they are changed by another URL or until the blade is completely restarted. It is important to note that URL settings *will* override both command line options *and* profile settings.

Working with Authentication

Beginning with the 1.5 version of Tractor, scripts now have the ability to authenticate via the engine. The NIMBY application supports three different ways of handling the password required for authentication.

- **Direct Prompt**

If authentication is enabled in the tractor system, and no command-line option is used to provide the necessary login password, the nimby application will prompt to the terminal for that password one time.

- **Password in the Clear**

If security of passwords is not important, then the password can be specified on the command line using the --pass option. Note, that this is insecure and process searches or history searches will reveal this password.

- **Password Contained in a Configuration File**

A more secure method of sending the password is to use the --configfile command line option. The file specified by this parameter is a JSON file which at a minimum should contain the user and passwd information within a tractorconfig dictionary.

It is expected that this file is properly protected by system level file permissions.

The format of the file should look like this:

```
{
  "tractorconfig": {
    "monitor": "tractor-engine",
    "port": 8080,
    "user": "buildrobot",
    "passwd": "fixem"
  }
}
```

Note that the configuration file may contain other information which might be read by other tractor scripts which also require authentication. The NIMBY application will read and use any information defined in the Tractor config dictionary as shown above.