

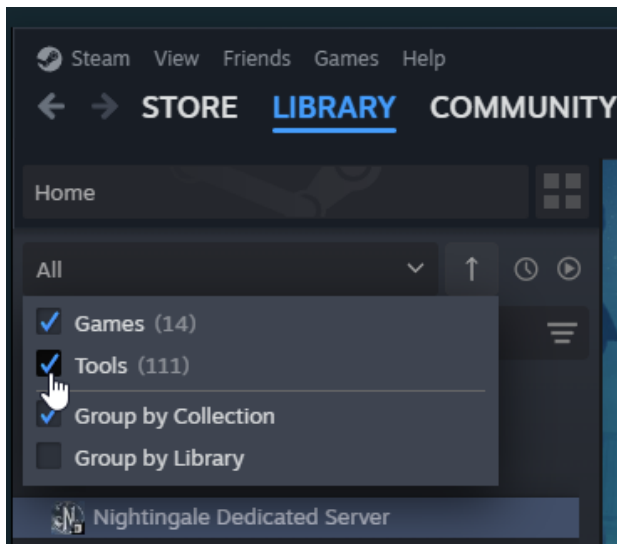
# Nightingale Dedicated Server

## Installation

The server package can be downloaded through the regular Steam client, or through the command-line steamcmd tool.

### Steam Client

First ensure that Tools are enabled in the Steam Library:



Then find “Nightingale Dedicated Server” and click install.

### steamcmd

The server can also be downloaded through the steamcmd command-line tool. First, please see Valve’s install and setup instructions to get steamcmd:

<https://developer.valvesoftware.com/wiki/SteamCMD>

Then install the server to a chosen path:

```
Shell
steamcmd +login anonymous +force_install_dir C:\Path\To\NightingaleServer +app_update
3796810 +quit
```

This works similarly on both Windows and Linux, the Nightingale Dedicated Server tool (app ID 3796810) contains versions for both operating systems.

# Importing Player and Realm Data

When connecting to a listen or dedicated server, Nightingale will bring a player's character to that server, including inventory, appearance, and everything else attached to the player.

When connecting to a new dedicated server that has not had a player connect previously, Nightingale will additionally send and import the realms associated with the first player to connect. This allows you to take a player and set of realms from the online environment and fully migrate to a dedicated server.

If you do not want this behaviour, create a new character to join the server with first, to set up an empty Abeyance realm. Then reconnecting with an existing character will not import realms.

# Running the Dedicated Server

## Steam Client

From the Steam client, add any desired commandline arguments through right-click, Properties > Launch Options. Nightingale Dedicated Server can be run at the same time as Nightingale. Add `-log` to get a console window when launching the dedicated server. All command line arguments below can be entered as a launch option.

## Command-line

Run `./NWXServer.sh` or `.\NWXServer.exe` depending on your OS. Add any applicable command line arguments from above. Add `-log` on Windows to get a console window.

Hit Control-C once (or twice if needed) to start a graceful shutdown.

# Configuration

There are two ways to configure most options for the server:

- Pass arguments on the command-line.
  - If using the Steam Client, these are added to the Launch Options under the Properties entry on the context menu.
  - If running from the command-line directly, add the arguments directly.
- Enter options in the .ini configuration file.
  - First, copy the NWX/Config/ExampleServerSettings.ini file to NWX/Config/ServerSettings.ini, and then edit these settings in that file.
  - Options are in the [/Script/NWX.NWXServerSettings] section unless otherwise noted.

Some options are only available on the command-line, and will not work if added to the config file. These are noted below.

## Connection Password

By default the server will start with no password required for players to connect. It is recommended to set one to ensure only players you intend can connect.

Command-line	-ini:ServerSettings:[/Script/NWX.NWXServerSettings]:Password=<password>
Config File	[/Script/NWX.NWXServerSettings] Password=<password>

where <password> is replaced with the intended password.

## Admin Password

If an admin password is set, players with the correct password can authenticate in-game in order to kick and ban other players.

Command-line	-ini:ServerSettings:[/Script/NWX.NWXServerSettings]:AdminPassword=<password>
Config File	[/Script/NWX.NWXServerSettings] AdminPassword=<password>

where <password> is replaced with the intended password.

## Max Players

By default Nightingale supports a maximum of six players in a realm. This maximum can be changed through a command-line argument.

Please note that increasing the maximum number of players past six is unsupported and may result in unintended behaviour.

Command-line	<code>-ini:Game:[ /Script/Engine.GameSession]:MaxPlayers=&lt;num&gt;</code>
Config File	<i>Can't currently be set in the config file.</i>

where <num> is the maximum number of players.

## Cheats Mode

To allow use of the cheat/debug menu and commands, add the enable cheats command-line argument. Players will need to authenticate with the admin password to use the cheats in-game.

Command-line	<code>-EnableCheats</code>
Config File	<i>Can't currently be set in the config file.</i>

## Network Settings

Nightingale listens on UDP port 7777 by default. To listen on a different port, add the port argument to the command-line.

Command-line	<code>-port=&lt;num&gt;</code>
Config File	<i>Can't currently be set in the config file.</i>

where <num> is in the range 1024-65535. Ensure the = is included or Unreal will ignore the setting.

Nightingale listens on all interfaces by default. To limit binding to a single interface, add the multihome argument to the command-line.

Command-line	<code>-multihome=&lt;IP&gt;</code>
Config File	<i>Can't currently be set in the config file.</i>

where <IP> is the address assigned to the interface to listen on.

## Logging

Server logs will be written to `NWX/Saved/Logs` under the directory where the server is installed. When installed with the Steam client, this can be found by right-clicking and selecting `Manage > Browse Local Files`.

### JSON Logging

Structured logging in a machine-parseable format can be enabled with these commandline arguments: `-ini:Engine:[JsonLogger]:bEnable=true`  
`-ini:Engine:[JsonLogger]:bStdout=true -noconsole`

## Persistence (Saved Data)

Realm data is be stored in: `NWX/Saved/Offline/DedicatedServer/Deploy`

Character data is stored in: `NWX/Saved/Offline/<PlatformID>/Profiles` where PlatformID is the Steam or Epic ID of each connecting user.

## Backups

Copying the following directories to a backup location is sufficient to capture the state of a Nightingale dedicated server:

- `NWX/Saved/Offline`
- `NWX/Saved/Config`
- `NWX/Config`

The first time the dedicated server is launched on a particular weekday, it will copy the Offline directory into the OfflineBackup directory.

## Resetting Server Persistence

To reset the server to an empty state, remove the following directories:

- `NWX/Saved/Offline`
- `NWX/Saved/Config`

## Status Endpoint

To enable scraping server status from an http endpoint, add `-statusPort=<port>` to the commandline, where `<port>` is a port in the range 1024-65535. Status can be scraped from the `/status` endpoint periodically.

By default this will bind to localhost only. To bind to a specific IP, add `-ini:Engine:[HTTPServer.Listeners]:+ListenerOverrides=(Port=<port>, BindAddress=<IP>)` as well, or use `0.0.0.0` to bind to all available interfaces.

If the server is still loading and not yet ready for players, it will return a 503 Server Unavailable http status and a JSON object similar to this:

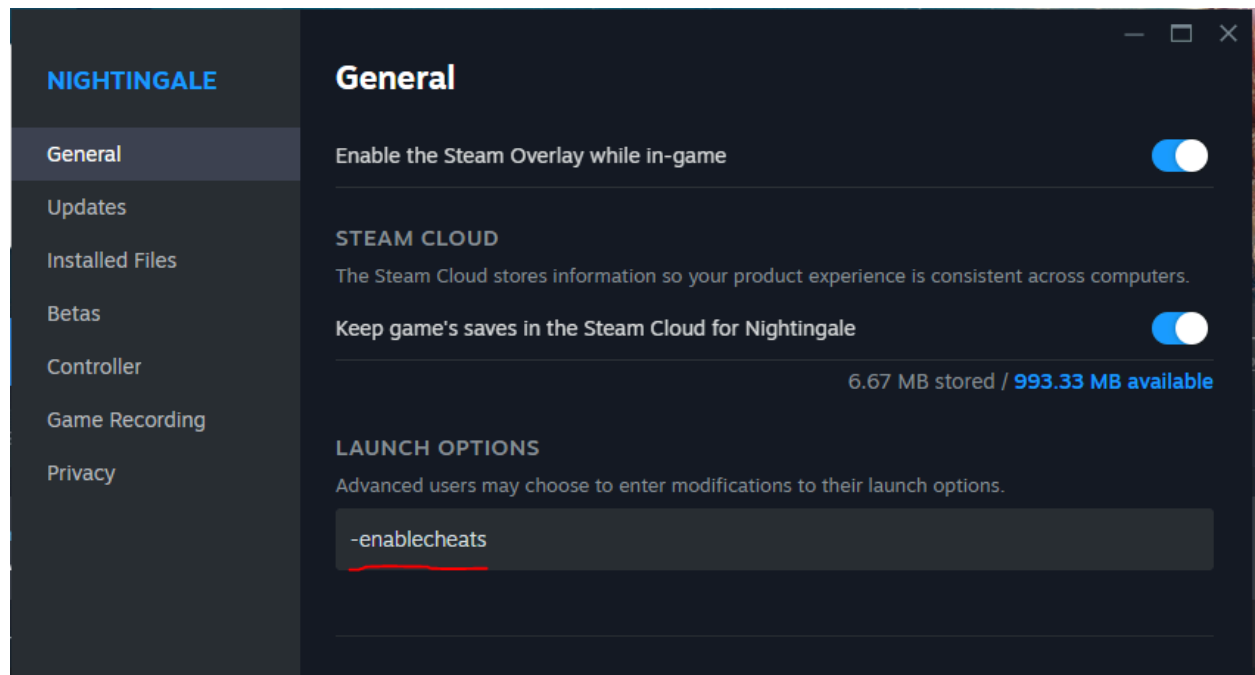
```
JavaScript
{
  "status": "loading"
  "player_count": 0
  "player_names": []
}
```

Once the server is ready for players, it will return a 200 OK status and a JSON object similar to this:

```
JavaScript
{
  "status": "ready"
  "player_count": 2
  "player_names": ["Alice", "Bob"]
}
```

# Enabling Cheats Mode on Clients

To enable debug/cheats, the client must be launched with the `-EnableCheats` commandline argument:



Once the game is launched, in single player offline mode cheats will always be available through the in game menus, no password required.

If you are hosting a listen server, you also do not need to enter a password to enable debug. Connecting clients will not be able to use debug.

See above for instructions on enabling cheats on a dedicated server as well as where to set the admin password.