N!tmod Installation Tutorial

Download N!tmod

Download the latest version of N!tmod from here

Installing/Updating

Depending on how the mod package was created, it may contain all the required files inside a "nitmod" folder, or directly into the Zip root.

Installing for the first time

Stop your server
Unzip the mod package.
If the mod package doesn't contain a "nitmod" folder:
Create a "nitmod" folder on your server, next to the "etmain" folder.
Upload the content of the mod package to the folder you just created
If the mod package contains a "nitmod" folder:
Upload the "nitmod" folder to your server's root (near "etmain")
Once the files are uploaded:
Change your server's "fs_game" cvar to "nitmod" (either in your control panel for rented game servers, or in your startup command line if you're hosting it yourself)
Start your server

Updating

Stop your server
Delete the current mod pk3 file from your server's "nitmod" folder (usually nitmod_*.*.pk3)
Unzip the mod package.
If the mod package doesn't contain a "nitmod" folder:
Upload the content of the mod package to your server's "nitmod" folder
If the mod package contains a "nitmod" folder:
Upload the content of the package's "nitmod" folder to your server's "nitmod" folder
In both cases, you will probably have to confirm you want to replace qagame_mp_x86.dll (Windows servers) or qagame.mp.i386.so (Linux servers).
Once the files are uploaded:
Start your server

Server configuration

N!tmod releases come with a sample configuration file (nitmod.cfg).
It contains every mod Cvars set to their default value, and a small description of what they're used for.
For a complete description, visit our Cvar reference (link on the left menu of this page).

Admin system configuration

Setting up SQLite Database

N!tmod includes a powerful Shrubbot like admin system, partially rewritten to use a SQLite database, extend features and provide higher performance.
It allows you to modify/add/delete admin levels and manage every player who has connected to your server, even whey they are offline

Admin system is disabled by default, because it requires you to set the path to where you want the SQLite database file to be saved, using the **n_SQLiteDBPath** Cvar.
The database file will be automaticly created if the above cvar is set correctly, and the server process has read/write permissions on the specified path.
If an invalid filename is provided, the mod will attempt to create a "NITMOD_DB.sqlite" database inside "fs_homepath"/nitmod directory.
The SQLite database is also required for XPSave (see g_XPSave), Offline messaging (see n_userMail) and Map records (see n_mapRecords)

Setting up admin levels

To setup admin levels on your server, you must create a levels.db file inside nitmod folder (download sample here).
DO NOT copy & paste level entries from another mod's 'shrubbot.cfg' file, it will NOT work!

Here is 'levels.db' file structure description:

```
**********  // Delimiter (10 *) Must IMPERATIVELY be placed BEFORE every level.
level =     // Level Number
name =      // Level Name
flags =     // Flags
gtext =     // Optional Greeting Text (user's greeting text overrides this value)
gsound =    // Optional Greeting Sound (user's greeting sound overrides this value)
```

This file can be edited ingame (or through rcon) using !levedit, !levadd and !levdelete commands.
The file will automaticly be overwritten when using one of these commands.
If you manually edit this file while the server is running, use !readconfig command to load modifications.
Use !levlist and !levinfo commands to display informations about existing levels.

Setting up custom commands

N!tmod allows adding custom commands to the existing admin system commands set.
To setup custom commands on your server, you must create a commands.db file inside nitmod folder (download sample here).
DO NOT copy & paste custom commands from another mod's 'shrubbot.cfg' file, it will NOT work!

Here is 'commands.db' file structure description:

```
**********  // Delimiter (10 *) Must IMPERATIVELY be placed BEFORE every command.
name =      // Command name (ex: "test")
exec =      // Command sent to server (ex: chat "^3Test command")
desc =      // Text displayed in !help cmdname (ex: "Test command")
levels =    // Levels having access to this command, delimited by spaces (ex: 0 1 2)
```

If you manually edit this file while the server is running, use !readconfig command to load modifications.

Setting up custom votes

Starting from version 2.2, server admins can create custom votes.
To add custom votes, you must create a votes.db file inside nitmod folder

Here is 'votes.db' file structure description (N!tmod 2.2):

```
********** // Delimiter (10 *) Must IMPERATIVELY be placed BEFORE every vote.
name =     // max 20 chars vote name (to use with /callvote)
cmd =      // max 100 chars command to execute on server side when vote passes
message =  // max 256 chars message displayed on clients when vote is called
```

download sample here

Starting from N!tmod 2.3, the votes.db syntax has been modified, adding more flexibility.
Here is 'votes.db' file structure description (N!tmod 2.3 and higher)

```
**********  // Delimiter (10 *) Must IMPERATIVELY be placed BEFORE every vote.
name =      // max 20 chars vote name (to use with /callvote)
message =   // max 256 chars message displayed on clients when vote is called
help =      // max 256 chars vote description (ex: /callvote *name* ?)
exec =      // max 1024 chars command to execute on server side when vote passes
passtext =  // max 256 chars message displayed on clients if vote passes
levels =    // max 64 levels allowed to call this vote (if empty, vote is available for all levels)
```

download sample here

If you manually edit this file while the server is running, use !readconfig command to load modifications.

To call one a custom vote, use the /callvote command in your console, followed by the custom vote name. Ex : /callvote mycustomvote.