

Contents

1. Overview
2. About QMM
3.
 - + Windows Installation
4.
 - + Linux Installation
5.
 - + Configuration

Overview

This document explains the installation and configuration for QMM and QMM plugins. If you are looking for QMM plugin API details, please go [here](#).

[Top](#)

About QMM

QMM is a server-side extension for Quake 3 that is loaded between the engine and the mod file to allow plugin modules to be loaded which can interface with both.

Normally, when a Quake 3-based game server is running, there are 2 major parts: the engine and the mod. The engine is the server program that you run, and the mod is a dynamically-loaded file that interacts with the engine to form a unique game. The server loads the mod and they communicate back and forth to exchange data or notify about certain things. This is a simple diagram of this relationship:

```
[engine] <--> [mod]
```

When you use QMM, however, this system is different. QMM pretends to be the mod file and is loaded by the engine instead of the real mod:

```
[engine] <--> [QMM]
```

At this point, the engine thinks it has loaded the mod. Once QMM is loaded, it loads the real mod and, in the mod's perspective, it pretends it's the engine:

```
[engine] <--> [QMM] <--> [mod]
```

Now, it is fully intercepting any messages passed between the engine and the mod and allows QMM plugins to access and modify these messages.

Also, as of version 1.1.0, QMM includes "NoCrash". Formerly a plugin offered with 1.0.0, it is now built-in to QMM and it fixes the well-known msgboom exploit plaguing Quake 3-based game servers. See [Cvars](#) for more details.

[Top](#)

Windows Installation

Quake 3 Arena

1. The first step is different for .dll mods (step 1a) and for .qvm mods (step 1b):
 1. If the original mod file *qagamex86.dll* exists, rename it to *qmm_qagamex86.dll*.
 2. If there is no *qagamex86.dll*, set the *vm_game* cvar to 0 with `seta vm_game 0` in the server console.
2. Place *qmm.dll* from the .zip you downloaded into the mod directory and rename it to *qagamex86.dll*.
3. Place *pdb.dll* and *qmm.ini* from the .zip you downloaded into the root game directory (where the quake3.exe file is located).
4. [Configure QMM](#)

[Top](#)

Jedi Knight 2

1. The first step is different for .dll mods (step 1a) and for .qvm mods (step 1b):
 1. If the original mod file *jk2mpgamex86.dll* exists, rename it to *qmm_jk2mpgamex86.dll*.
 2. If there is no *jk2mpgamex86.dll*, set the *vm_game* cvar to 0 with `seta vm_game 0` in the server console.
2. Place *qmm.dll* from the .zip you downloaded into the mod directory and rename it to *jk2mpgamex86.dll*.
3. Place *pdb.dll* and *qmm.ini* from the .zip you downloaded into the root game directory (where the jk2mp.exe/jk2ded.exe file is located).
4. [Configure QMM](#)

[Top](#)

Jedi Academy

1. Extract *jampgamex86.dll* from the latest .pk3 file by extracting it manually with a Zip program like WinZip, or run the server at least once. Once it is extracted to the mod directory, rename it *qmm_jampgamex86.dll*.
2. Place *zzz_qmm_jka.pk3* from the .zip you downloaded into the mod directory.
3. Place *pdb.dll* and *qmm.ini* from the .zip you downloaded into the root game directory (where the jamp.exe/jampDed.exe file is located).
4. [Configure QMM](#)

[Top](#)

Return to Castle Wolfenstein (Multiplayer)

1. Rename the existing *qagame_mp_x86.dll* file to *qmm_qagame_mp_x86.dll*.
2. Place *qmm.dll* from the .zip you downloaded into the mod directory and rename it to *qagame_mp_x86.dll*.
3. Place *pdb.dll* and *qmm.ini* from the .zip you downloaded into the root game directory (where the wolfmp.exe file is located).
4. [Configure QMM](#)

[Top](#)

Return to Castle Wolfenstein: Enemy Territory

1. Rename the existing *qagame_mp_x86.dll* file to *qmm_qagame_mp_x86.dll*.
2. Place *qmm.dll* from the .zip you downloaded into the mod directory and rename it to *qagame_mp_x86.dll*.
3. Place *pdb.dll* and *qmm.ini* from the .zip you downloaded into the root game directory (where the *exe/etded.exe* file is located).
4. [Configure QMM](#) (see [Configuration File \(qmm.ini\)](#) regarding a setting that **must** be set for Enemy Territory).

[Top](#)

Return to Castle Wolfenstein (Singleplayer)

1. Rename the existing *qagamex86.dll* file to *qmm_qagamex86.dll*.
2. Place *qmm.dll* from the .zip you downloaded into the mod directory and rename it to *qagamex86.dll*.
3. Place *pdb.dll* and *qmm.ini* from the .zip you downloaded into the root game directory (where the *wolfsp.exe* file is located).
4. [Configure QMM](#) (see [Configuration File \(qmm.ini\)](#) regarding a setting that **must** be set for RtCW Singleplayer).

[Top](#)

Linux Installation

Quake 3 Arena

1. The first step is different for .so mods (step 1a) and for .qvm mods (step 1b):
 1. If the original mod file *qagamei386.so* exists, rename it to *qmm_qagamei386.so*.
 2. If there is no *qagamei386.so*, set the *vm_game* cvar to 0 with `seta vm_game 0` in the server console.
2. Place *qmm.so* from the .tar.gz you downloaded into the mod directory and rename it to *qagamei386.so*.
3. Place *pdb.so* and *qmm.ini* from the .zip you downloaded into the root game directory (where the server binary is located).
4. [Configure QMM](#)

[Top](#)

Jedi Knight 2

1. The first step is different for .so mods (step 1a) and for .qvm mods (step 1b):
 1. If the original mod file *jk2mpgamei386.so* exists, rename it to *qmm_jk2mpgamei386.so*.
 2. If there is no *jk2mpgamei386.so*, set the *vm_game* cvar to 0 with `seta vm_game 0` in the server console.
2. Place *qmm.so* from the .tar.gz you downloaded into the mod directory and rename it to *jk2mpgamei386.so*.
3. Place *pdb.so* and *qmm.ini* from the .zip you downloaded into the root game directory (where the server binary is located).
4. [Configure QMM](#)

[Top](#)

Jedi Academy

1. Rename the existing *jampgamei386.so* file to *qmm_jampgamei386.so*.
2. Place *qmm.so* from the .tar.gz you downloaded into the mod directory and rename it to *jampgamei386.so*.
3. Place *pdb.so* and *qmm.ini* from the .zip you downloaded into the root game directory (where the server binary is located).
4. [Configure QMM](#)

[Top](#)

Return to Castle Wolfenstein (Multiplayer)

1. Rename the existing *qagame.mp.i386.so* file to *qmm_qagame.mp.i386.so*.
2. Place *qmm.so* from the .tar.gz you downloaded into the mod directory and rename it to *qagame.mp.i386.so*.
3. Place *pdb.so* and *qmm.ini* from the .zip you downloaded into the root game directory (where the server binary is located).
4. [Configure QMM](#)

[Top](#)

Return to Castle Wolfenstein: Enemy Territory

1. Rename the existing *qagame.mp.i386.so* file to *qmm_qagame.mp.i386.so*.
2. Place *qmm.so* from the .tar.gz you downloaded into the mod directory and rename it to *qagame.mp.i386.so*.
3. Place *pdb.so* and *qmm.ini* from the .zip you downloaded into the root game directory (where the server binary is located).
4. [Configure QMM](#) (see [Configuration File \(qmm.ini\)](#) regarding a setting that **must** be set for Enemy Territory).

[Top](#)

Return to Castle Wolfenstein (Singleplayer)

1. Rename the existing *qagamei386.so* file to *qmm_qagamei386.so*.
2. Place *qmm.so* from the .tar.gz you downloaded into the mod directory and rename it to *qagamei386.so*.
3. Place *pdb.so* and *qmm.ini* from the .zip you downloaded into the root game directory (where the server binary is located).
4. [Configure QMM](#) (see [Configuration File \(qmm.ini\)](#) regarding a setting that **must** be set for RtCW Singleplayer).

[Top](#)

Configuration

Configuration File (qmm.ini)

The qmm.ini file is where most of the settings for QMM are located.

Here is the standard qmm.ini:

```
#QMM Configuration File
#Settings specified in this file override default or auto-detected settings

#=====

#Specifies which game engine data to use
# default = auto-detect
# valid options:
#   "Q3A" = Quake 3 Arena
#   "JK2" = Jedi Knight II: Jedi Outcast
#   "JKA" = Jedi Knight: Jedi Academy
#   "RTCWMP" = Return to Castle Wolfenstein (Multiplayer)
#   "RTCWET" = Return to Castle Wolfenstein: Enemy Territory - Must be specifically set,
otherwise auto-detected as RTCWMP
#   "RTCWSP" = Return to Castle Wolfenstein (Singleplayer) - Must be specifically set,
otherwise auto-detected as Q3A
#"game" "Q3A"

#=====

#The following settings are per-mod settings for this game

#The following settings are for the "baseq3" mod (default for Q3A)
"baseq3" {
    #Specifies file name of mod to load
    # default(Q3A) = "vm/qagame.qvm"
    # default(JK2) = "vm/jk2mpgame.qvm"
    # default(other) = "qmm_<name>.dll"/"qmm_<name>.dll"
    #"mod" "vm/qagame.qvm";

    #Specifies size of the QVM stack in megabytes (1048576 bytes)
    # default = "1"
    # only used with qvm mods
    #"qvmstacksize" '1';

    #Specifies name of config file to execute after load
    # default = "qmmaddons/qmm/qmmexec.cfg"
    #"execcfg" "qmmaddons/qmm/qmmexec.cfg";

    #List of plugins to load
    #"plugins" (
        #"qmmaddons/plugin1/dlls/plugin1.dll";
        #"qmmaddons/plugin2/dlls/plugin2.dll";
    )
}
}
```

All lines starting with # are comments and are ignored. By default, all settings are commented out. Simply uncomment it to use it.

You'll notice the first setting is "game". This specifies which game QMM is being loaded for. It is *required* when using RtCW: Enemy Territory or RtCW Singleplayer because the auto-detector is unable to distinguish them from other games.

The next setting is a group setting called "baseq3". This contains settings specifically for the baseq3 mod. Each mod gets its own group in the config file.

Inside the group, you'll see a "mod" setting. This is the filename, relative to the mod directory, of the mod file to load. The default for Quake 3 Arena and Jedi Knight 2 is a .qvm file, but if that is not found, it will fall back on qmm_<modfilename> (ex. qmm_qagamex86.dll). All other games default straight to the qmm_<modfilename> file.

The next setting inside the mod group is "qvmstacksize". This is only used for QVM mods, and it should only be used if you find the QVM running out of memory due to recursion. It is best to leave it at '1'. Note that this setting uses single quotes for the value (') instead of double quotes (") like the rest. This denotes that the value is a number, not a text string.

Next up is "execcfg". This specifies the filename of the config file that is run once QMM is done loading all plugins.

The last setting in the group is "plugins". This is a list setting that is used to specify filenames (relative to the mod directory) of plugins to load.

Finally, there is a hidden global setting (outside of a mod group, along with "game"), "nogreeting". This is also a number setting (meaning you use single quotes (') to set the value, not double quotes (")), and if set to 1, no QMM greeting will be displayed to connecting clients.

[Top](#)

Cvars

QMM adds the following cvars:

- qmm_version - Shows the version of QMM installed. (Read-only, public)
- qmm_game - Specifies the game QMM is loaded for, can be "q3a", "jk2", "jka", "rtcwmp", "rtcwet", or "rtcwsp". (Read-only)
- qmm_nocrash - Set to 0 to disable the built-in NoCrash feature. Set to 1 (the default) to enable it.

[Top](#)

Commands

QMM adds the following commands:

- qmm
 - qmm status - displays information about QMM
 - qmm list - displays information about loaded QMM plugins
 - qmm load <file> - loads a new plugin
 - qmm info <id/str> - outputs info on plugin with id
 - qmm pause <id/str> - pauses plugin with id
 - qmm unpause <id/str> - unpauses plugin with id
 - qmm unload <id/str> - unloads plugin with id
 - qmm force_unload <id/str> - forcefully unloads plugin with id
 - qmm shutdown - permanently unloads QMM except for necessary functions

[Top](#)

Plugin Installation

For organizational purposes, all plugins should be placed inside the *qmmaddons* directory inside the mod directory. Each plugin gets its own subdirectory inside *qmmaddons*. Here is an example directory structure for the standard *baseq3* Quake 3 Arena mod:

```

baseq3
+---qmmaddons
|   +---advchat
|   |   \---dlls
|   |
|   +---qadmin
|   |   +---config
|   |   \---dlls
|   |
|   +---qmm
|   |   \---qmmexec.cfg
|   |
|   +---rocketmod
|   |   +---config
|   |   \---dlls
|   |
|   \---stripper
|       +---dlls
|       +---dumps

```

```
|          \---maps
|
|+---qagamex86.dll
|
|+---qmm_qagamex86.dll
```

Each plugin will have information as to how its directory should be structured.

Once you have the directory structure set up, open up *qmm.ini* in the root server directory and add a line to the *plugins* list section pointing to the new .dll or .so:

```
"plugins" (
    "qmmaddons/stripper/dlls/stripper_qmm.dll";
)
```

(see the [Configuration File \(qmm.ini\)](#) section above for more information)

[Top](#)

QMM was written by Kevin Masterson (CyberMind)

<kevinm -AT- planetquake -DOT- com>

<cybermind -AT- users -DOT- sourceforge -DOT- net>

QMM is released under the [GPL](#)

Copyright 2005 Kevin Masterson, all rights reserved.