

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | Written by Magius Rerecros |  |  |

Mewgibot

A customized Twitch Chat bot.

Table of Contents

1. [Overview 2](#_Toc475362211)
2. [Caster Log 3](#_Toc475362212)
3. [Custom Commands 3](#_Toc475362213)

[Adding a Command](#_Toc475362214)

[Deleting a Command](#_Toc475362215)

1. [Queues 6](#_Toc475362216)
2. [Quotes 6](#_Toc475362217)

[Adding a Quote](#_Toc475362218)

[Deleting a Quote](#_Toc475362219)

[Displaying a Quote in Chat](#_Toc475362220)

1. [Raffles 8](#_Toc475362221)
2. [Regulars 8](#_Toc475362222)

[Adding a Regular](#_Toc475362223)

[Deleting a Regular](#_Toc475362224)

1. [Scheduled Messages 9](#_Toc475362225)

[Adding Scheduled Messages](#_Toc475362226)

[Deleting Scheduled Messages](#_Toc475362227)

1. [Trivia 11](#_Toc475362228)
2. [Update Stream Info 12](#_Toc475362229)
3. [Whitelisted Links 12](#_Toc475362230)

[Adding a Whitelisted Link](#_Toc475362231)

[Deleting Whitelisted Links](#_Toc475362232)

1. [!caster Command 13](#_Toc475362233)
2. [Files 14](#_Toc475362234)
3. [Settings 14](#_Toc475362235)

[Bot Account](#_Toc475362236)

[Appearance](#_Toc475362237)

[Bot Settings](#_Toc475362238)

1. [Support 17](#_Toc475362239)

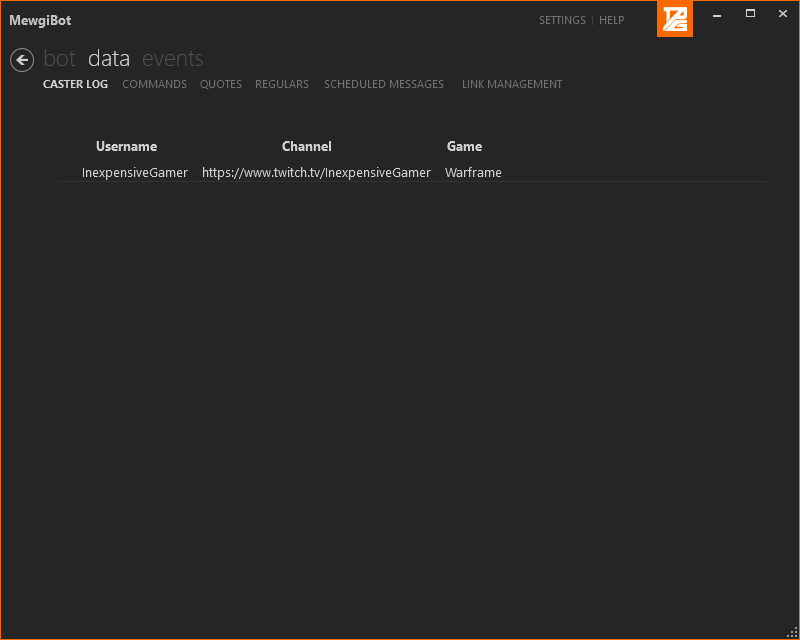
# Overview

Mewgibot is an ongoing project designed primarily to support the broadcasters of the Top Percentage Gaming community on TwitchTV. The bot was created using the C# programming language, running on .NET 4.5.2. It makes use of the open-source libraries [TwitchLib by SwiftySpiffy](https://github.com/swiftyspiffy/TwitchLib) and [Modern UI for WPF by FirstFloorSoftware](https://github.com/firstfloorsoftware/mui). It also makes use of the [jService.io Web API](http://jservice.io/), and the [Twitch Token Generator by SwiftySpiffy](https://twitchtokengenerator.com/).

The basic features of Mewgibot include:

* Custom commands
* Quotes
* Scheduled Messages
* Raffles
* Queue
  + Allows you to, for example, set up a list of people to take turns playing an online game with you
* Trivia Game for chat
* Advertisement of other broadcasters via “!caster” command
  + Keeps track of broadcasters who have been advertised
  + Optionally advertises broadcasters who host the channel the bot is connected to automatically
  + List of advertised broadcasters is saved when the bot is closed, and the file is opened in Notepad when the bot is closed
* Automatic greeting for new subscribers or re-subs
* “!purge” command for moderators
* Automatic purging of hyperlinks
  + Optionally allow all subs and “regulars” to post links without getting purged
  + Allows non-subs and non-regulars to post links after a mod uses a “!permit” command for that user
  + Optionally whitelist specific links
* Display currently playing song in chat via “!song” command
  + Data pulled from a text file specified by the user of the bot
* Displays a chat log in the bot’s UI
  + Log is saved when the bot is closed
* Update stream information
* Bot’s software automatically updates on launch

# Caster Log

Whenever the “!caster” command is used, it gets stored in Data 🡪 Caster Log.  


When the bot is closed, it opens up this log in Notepad, so that if you want to tweet out broadcasters that were advertised in your channel, the information is easily accessible.

# Custom Commands

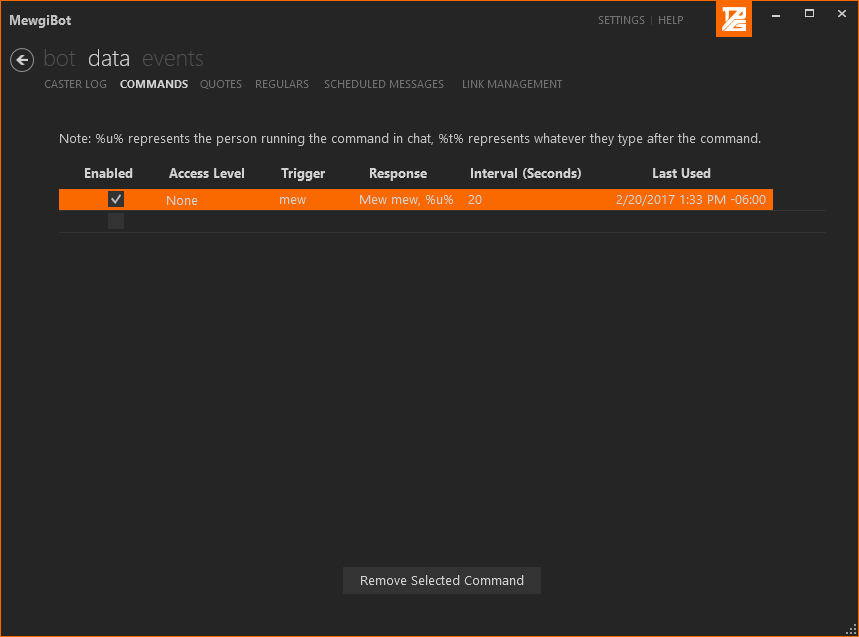
## Adding a Command

There are two ways to add a custom command to Mewgibot; either via chat (can be done by either the broadcaster, or moderators of the channel), or via the bot’s user interface.

### In-Chat Command Creation

Commands are added to Mewgibot in-chat via the “!addcmd” command. The syntax of the command is as follows:  
*!addcmd !commandTrigger “Bot’s response to the command”*

### UI Command Creation

Commands can also be added via the bot’s User Interface on the “Data 🡪 Commands” page:  


To create a new command in the interface, just click in the one of the cells of the empty row at the bottom of the list. Commands can also be edited in this manner by clicking on an already populated cell. It’s important to note that you should press the “Enter” key after making a change to a command to make sure that the bot properly updates its command list. This includes when changing the “Enabled” flag on a command.

### Command Details

Command triggers can be whatever you want; they are not case-sensitive, so “!mew” would be the same command as “!MEW”.

Commands added via chat are automatically set to be “enabled”, meaning that they can be used immediately after the command is created. If a command is added via the UI of the bot, the “Enabled” flag of the command needs to be checked before the command can be used.

Commands can only be run every so often, as determined by the command’s interval (specified in seconds).

Mewgibot currently supports two different variables within the command response:

* %u% - This gets replaced with the username of the person who uses the command in chat.
* %t% - This gets replaced with the first word typed after the command trigger in the same message the command was used.

*Example:*  
If a command is added in chat using the following syntax:  
*!addcmd !poke “/me pokes %t%”*  
then when someone types “!poke MagiusRerecros” in chat, the bot will respond with the message “/me pokes MagiusRerecros”.

*Example 2:*  
If a command is added in chat using the following syntax:  
*!addcmd !baka “%u% is a baka.”*  
then if MagiusRerecros types “!baka” in chat, the bot will respond with the message “MagiusRerecros is a baka.”

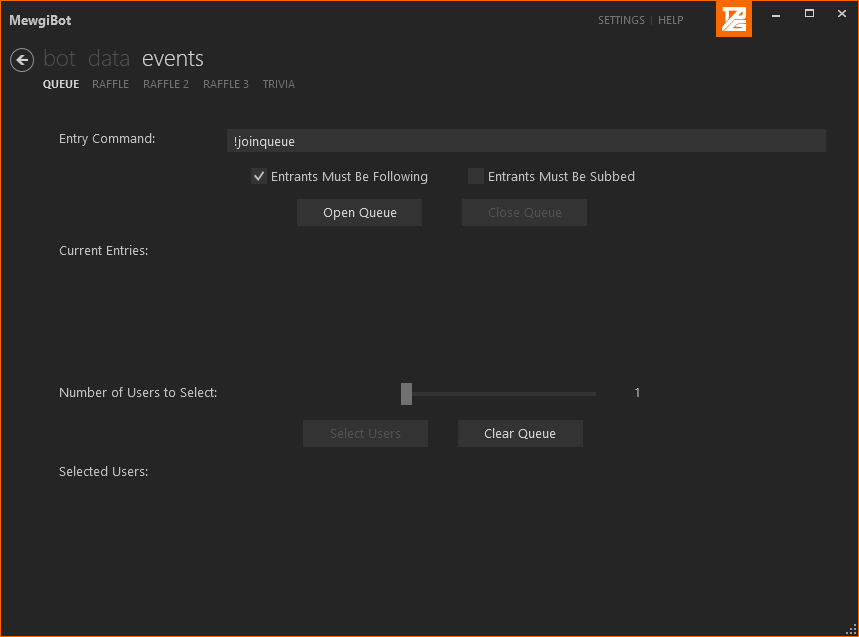
## Deleting a Command

You can delete commands from Twitch chat, by using the syntax:  
*!delcmd !commandTrigger*

You can delete a command from the bot’s User Interface by selecting the command you want to delete (click anywhere in the row containing the command) and click the “Remove Selected Command” button.

Commands can only be deleted by the broadcaster, or by channel moderators.

# Queues

Queues are run through the bot’s UI:  


The entry command can be set to whatever you want; keep in mind that it is not case-sensitive; i.e. “!JoinRaffle” is the same as “!joinraffle”.

# Quotes

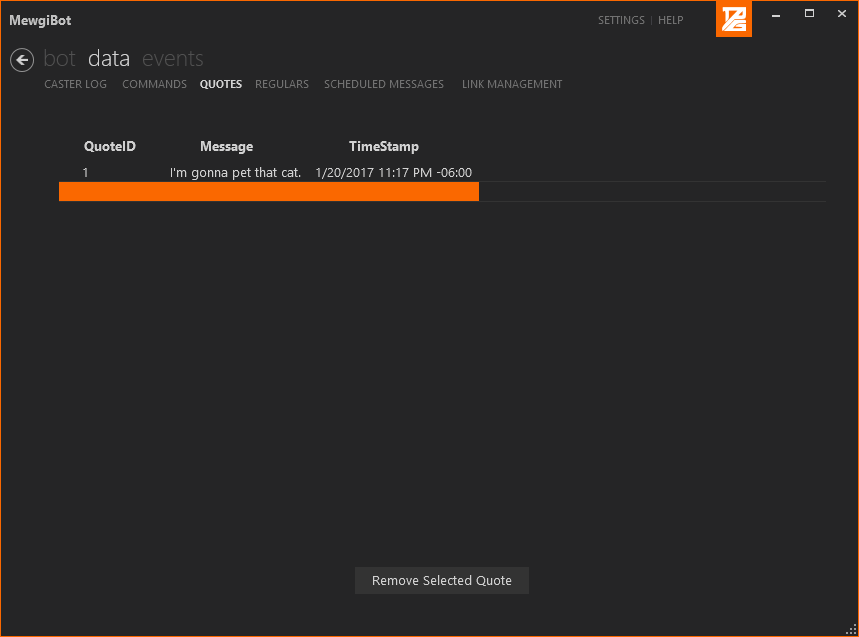
## Adding a Quote

Like commands, quotes can be added either in-chat (by anyone), or via the bot’s User Interface. It is recommended that quotes are added in-chat for simplicity.

### In-Chat Quote Creation

Quotes can be added by anyone in-chat by using the syntax:  
*!addquote “This is the quote”*

### UI Quote Creation

Quotes can also be added (or edited) via the bot’s UI on the “Data 🡪 Quotes” page:  


You can add a quote by clicking on an empty cell in the blank row at the bottom of the Quotes list. Likewise, you can edit a quote by clicking on an already populated cell within the Quotes list.

QuoteID MUST be a number, and TimeStamp should follow the template as shown in the image above, where “-06:00” represents the time zone (in this case, Central Standard Time).

As with Commands, make sure you press the “Enter” key after updating any information in a quote to make sure the bot updates properly.

## Deleting a Quote

You can delete quotes from Twitch chat, by using the syntax:  
*!delquote QuoteID*

You can delete a quote in the bot’s UI by selecting the quote (click anywhere in the row containing the quote you want to delete), and click the “Remove Selected Quote” button.

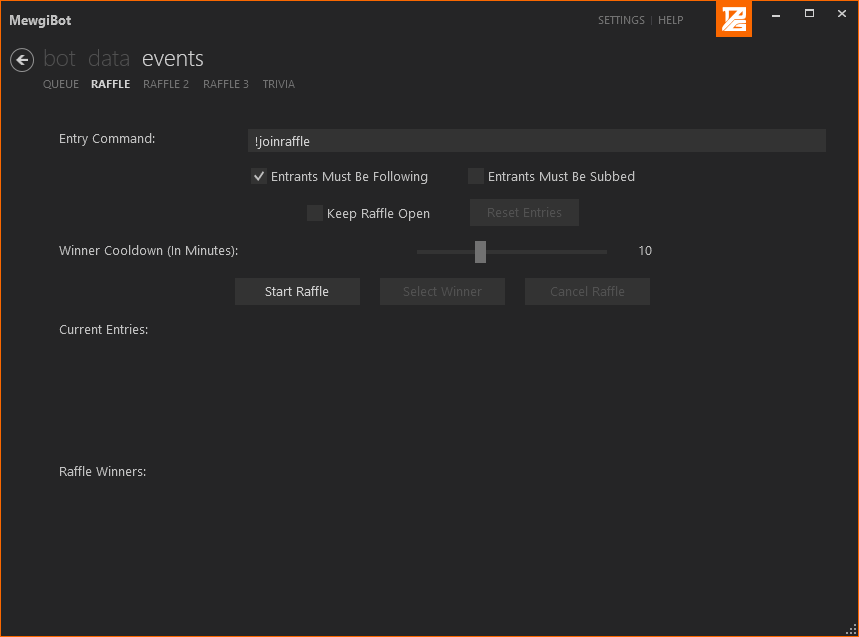
Quotes can only be deleted by the broadcaster, or channel moderators.

## Displaying a Quote in Chat

Quotes can be displayed in chat (by anyone) via the following syntax:  
*!quote QuoteID*

If no QuoteID is supplied, a random quote from the quote list is displayed. QuoteID’s can be found in the “Data 🡪 Quotes” page in the bot’s UI.

# Raffles

Raffles are run through the bot’s UI:  


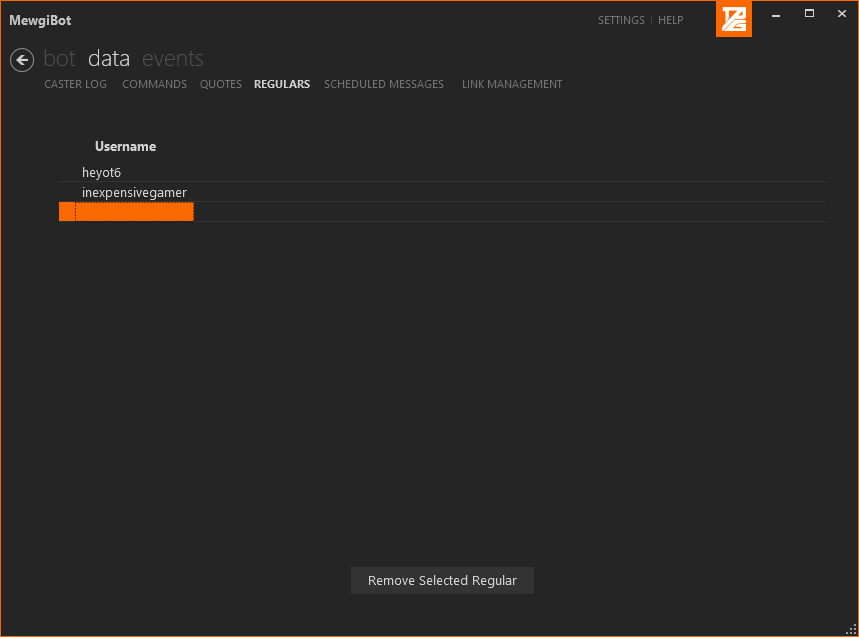
Currently, only three raffles can be open at a time.

The entry command can be set to whatever you want; keep in mind that it is not case-sensitive; i.e. “!JoinRaffle” is the same as “!joinraffle”.

# Regulars

Twitch users can be added as “Regulars”. Currently, the only benefit to being a “regular” is not automatically getting purged when posting a link, depending on the settings configured in the Bot Settings page.

## Adding a Regular

Regulars can only be added via the bot’s UI. Just click the empty row and type in the user’s Twitch username.  


## Deleting a Regular

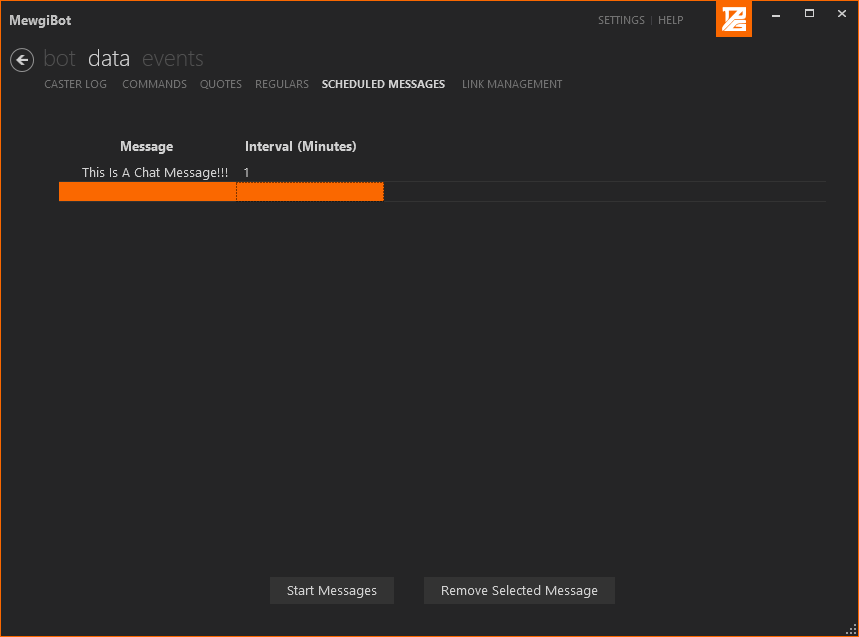
In order to delete a regular, just select the regular in the UI (click the row containing the regular’s twitch username) and click the “Remove Selected Regular” button.

# Scheduled Messages

Mewgibot can send messages after a specified amount of time. These are referred to as Scheduled Messages. Every time the specified interval for a message passes, that message gets sent; it is not a one-time operation.

The channel the bot is connected to MUST be online for scheduled messages to be sent.

## Adding Scheduled Messages

You can create scheduled messages through the bot UI.   


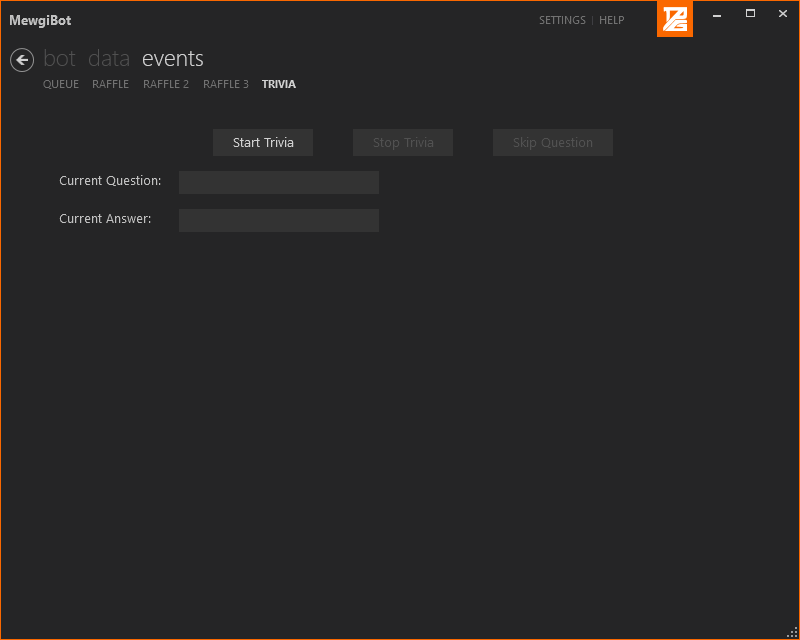
After adding messages through the UI, make sure to click the “Start Messages” button, or the messages will not be sent properly.

## Deleting Scheduled Messages

To delete a scheduled message, just select the message (click anywhere in the row containing the scheduled message) and then click the “Remove Selected Message” button.

You do not have to click the “Start Messages” button after removing a message, the timer will automatically restart.

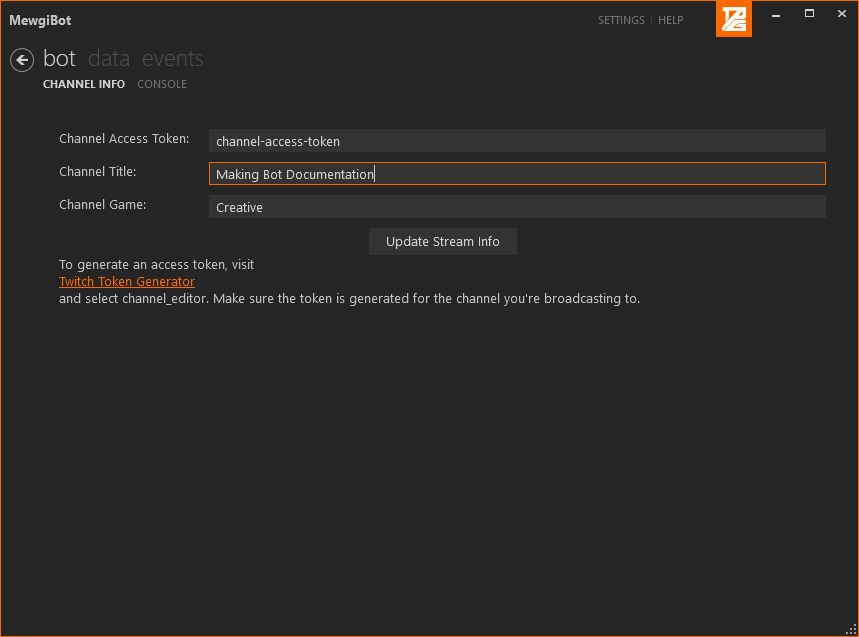
# Trivia

The Trivia mini-game is run through the bot’s UI:  


Currently there is no natural language support or anything of the like in the trivia mini-game. In other words, players must answer with exactly what shows in the “Current Answer:” textbox. The only exception is the fact that answers are not case-sensitive; in other words, if the answer to the question is “The New England Patriots”, the bot will accept “the new England patriots”, but not “New England Patriots”.

Trivia questions are pulled from [jService.io](http://jservice.io/).

# Update Stream Info

Stream information can be updated through the bot’s UI:  


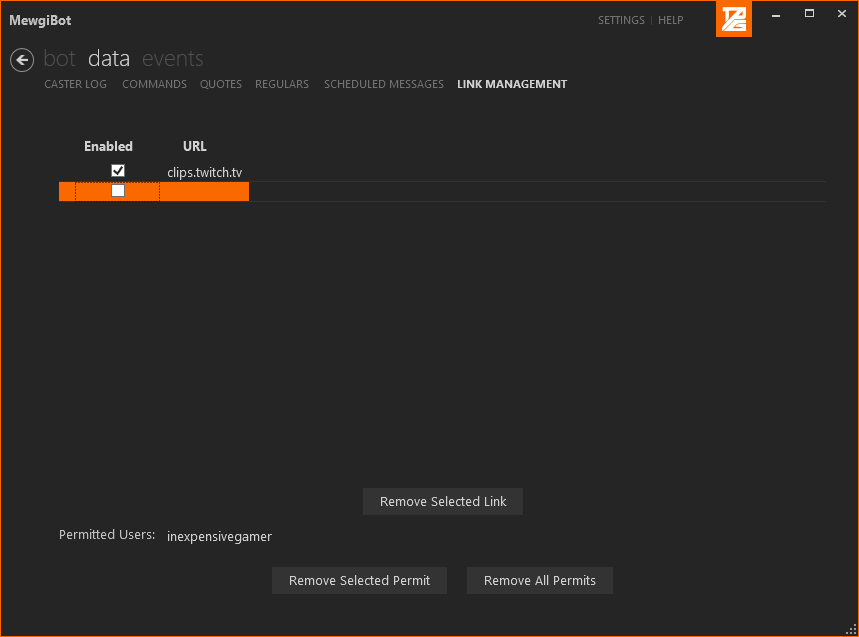
The channel access token can be generated at the [Twitch Token Generator](https://twitchtokengenerator.com/) website, created by [SwiftySpiffy](https://github.com/swiftyspiffy/). Make sure that “channel\_editor” is set to “Yes”, and that you log in to the Token Generator website using the broadcaster account, NOT the bot account.

# Whitelisted Links

You can maintain a list of links that are whitelisted by Mewgibot. In other words, anyone that posts a whitelisted link will not be purged by Mewgibot, regardless of the settings you’ve specified in the Bot Settings page.

## Adding a Whitelisted Link

Whitelisted links can only be added via the bot’s UI. Just click the “Enabled” box in the new row, select the URL entry for that row, and type in the URL you would like to whitelist, then press “Enter” on your keyboard.



## Deleting Whitelisted Links

To remove a whitelisted link, just select the link (click anywhere in the row containing the link you’d like to delete) and click the “Remove Selected Link” button.

# !caster Command

Mewgibot supports a “!caster” command, so that other broadcasters can be thanked and/or advertised in chat. The command is usable by the broadcaster and channel moderators. The syntax is as follows:  
*!caster TwitchBroadcaster*

When the “!caster” command is used, the bot responds with the message: “Follow <TwitchBroadcaster> over at <TwitchBroadcaster’s channel link> ! Last seen playing <Game currently set on TwitchBroadcaster’s channel>!”

Anyone who is thanked/advertised by Mewgibot is logged in the bot’s UI on the “Data 🡪 Caster Log” page. This includes users advertised manually through the “!caster” command, and users who are automatically advertised after they host the channel.

# Files

Data for Mewgibot is stored in various files:  
Commands - %UserProfile%\Documents\Mewgibot\Data\Commands.json  
Quotes - %UserProfile%\Documents\Mewgibot\Data\Quotes.json  
Advertised/Thanked Broadcasters - %UserProfile%\Documents\Mewgibot\Logs\thanks.json  
Chat logs - %UserProfile%\Documents\Mewgibot\Logs\log-<current date>.txt  
Backed up Bot Settings - %UserProfile%\Documents\Mewgibot\Settings\backup.json

Configuration data such as Twitch credentials, appearance settings, etc. are stored in a “user.config” file buried in the AppData folder.

# Settings

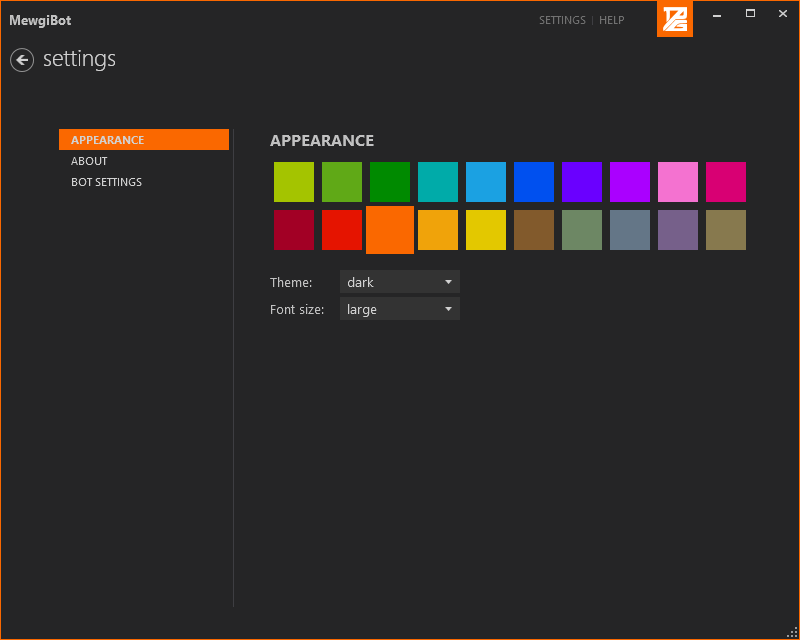
## Bot Account

When the bot is run for the first time, it will ask you if you would like to use the Mewgibot Twitch account; if you select “no”, you will be prompted for a Twitch username and the OAuth token corresponding to that username. OAuth tokens can be generated [here](https://twitchapps.com/tmi/).

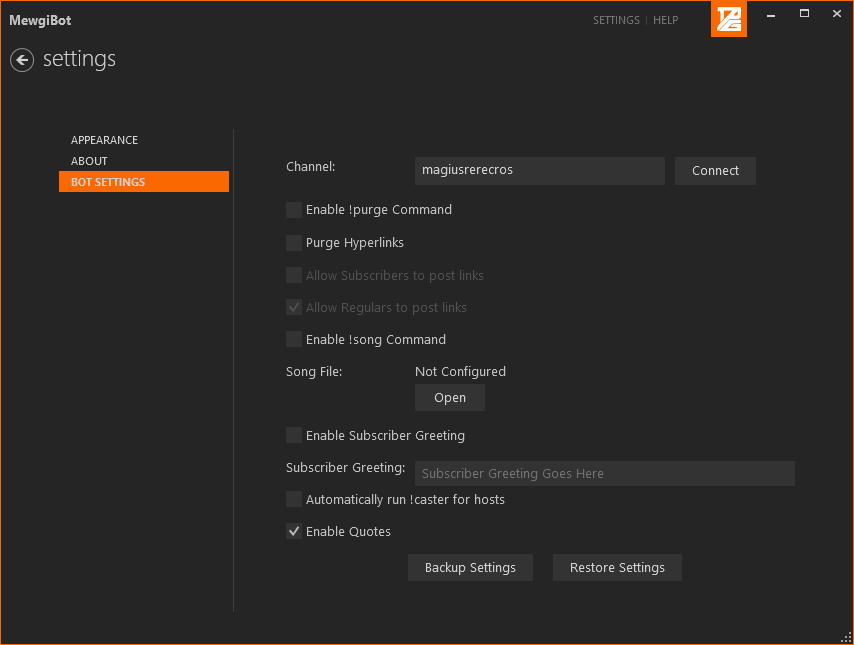
If you would like to change what account the bot is running under, you can click the “Log In” button on the “Bot 🡪 Console” page to supply new Twitch credentials. Alternatively, if you want to return to using the MewgiBot Twitch account, you can click the “Use Mewgibot Account” button.

Credentials are stored in a configuration file hidden somewhere in your AppData folder, with the OAuth token being stored as an encrypted string.

## Appearance

The color scheme and font size of the bot’s User Interface are customizable on the “Appearance” tab of the Settings page.  


## Bot Settings

The Bot Settings page has various settings related to the operation of the bot, such as defining what channel the bot is connected to, enabling automatic link purging, etc.  


### Purge

By checking the “Enable !purge Command” box in the Bot Settings, moderators can use the “!purge” command to time a user out for 1 second. This is a basic convenience option for moderators, that comes at the cost of removing anonymity.

The Twitch account that the bot is running under must be a channel moderator for this feature to function properly.

### Link Purging

By checking the “Purge Hyperlinks” box in the Bot Settings, all links typed in to chat will automatically be purged, aside from those posted by the broadcaster, channel moderators, or users who have been permitted to post a link.

The broadcaster or channel moderators can permit a user to post a link by using the following syntax:  
*!permit TwitchUser*This will allow TwitchUser to post a single link in chat.

There is a setting to allow subscribers and/or regulars to post links without getting purged.

### Subscriber Greeting

Mewgibot can automatically greet all new subscribers and all re-subs, by checking the “Enable Subscriber Greeting” box in the Bot Settings, and entering the greeting into the “Subscriber Greeting:” textbox. For example, you can enable a subscriber greeting of “!bonusall 12” to give all users 12 additional currency in Revlo-enabled channels whenever someone subscribes or re-subs.

### Song Display

By checking the “Enable !song Command” box in Bot Settings, and selecting a text file containing the currently-playing song, anyone in chat can use the “!song” command in chat to see what song is currently playing on the broadcast.

### Enable Quotes

By checking the “Enable Quotes” box in Bot Settings, chatters will be able to add and display quotes in chat, as described by the [Quotes](#_Quotes) section of this document.

### Backup/Restore Settings

By clicking the “Backup Settings”, you can save what settings you’ve selected in the “Bot Settings” page and reload them later by clicking the “Restore Settings” button.

The username and OAuth token used by Mewgibot is also backed up through this feature.

# Support

If you run in to any issues while using Mewgibot, or if you have any feature requests, send a message to Magius Rerecros on [Twitter](https://www.twitter.com/MagiusRerecros) or Discord (Mewgius#3572) with details and screenshots, if possible.