

# LARP Bot GM Guide

Thanks for trying LARP Bot for your secrets and powers LARP needs! LARP Bot was made with the MIT Assassins' Guild's style of secrets and powers LARP in mind, but GMs and/or game writers who want to run other types of LARPs may still find it useful. LARP Bot makes it possible to play LARPs with lots of fiddly bits on Discord, where the experience is something combining video chat and text adventure.

LARP Bot recognizes 62 commands, plus a help command. These are divided into GM commands for game runners to build the world of the game and player commands for players to interact with that world. Most of the commands are GM commands. Players only have to get to know 15 commands. In simpler games, some player commands won't be necessary.

The bot is flexible enough that you can create game worlds command by command in your Discord server, or load in a text file of commands to execute, allowing you to set up a game with little effort. Hopefully someday there will be lots of games converted to loadable text files. This guide will give you the knowledge necessary to set up and load such games or build your own. The guide will also cover player commands.

## Basics

LARP Bot uses a shared database across all the servers it's in, but each server's data is separated. The database keeps track of 8 different categories of data: Items, Areas, Containers, Inventories, Abilities, Roles, Memory Packets, and Signs. Here's a brief overview of what each of these are:

- **Items** are exactly what they sound like. They are things that players can have in their inventories, containers can hold, and can be moved around or used/consumed. Each item has a name, a description, optional contents, and a "bulky" number that represents how big and cumbersome it is.
- **Areas** are places in the game associated with a channel in a Discord server. Areas can have containers or signs associated with them. Searching an area in that area's channel will tell the player what signs and containers exist there.
- **Containers** hold items and are associated with an area. They also have description text and two attributes that control how long it takes to search them and whether the search will be random or not.
- **Inventories** are associated with character roles. They contain items, abilities, memory packets, and have a name of a character that is the same as the character role.
- **Abilities** are things that characters can do. They have a name, description, effect text, and two attributes that control whether the ability has a limited number of uses or a cooldown time. LARP bot doesn't keep track of how many times or when a character uses abilities, but does send messages to characters when a limited use ability is used and when a cooldown time is over.

- **Roles** are Discord server roles that are associated with a character. When a player has a character role, it allows them to use player commands and access their character's inventory.
- **Memory Packets** are bits of text that are revealed to players during the game. If you come from an MIT Assassins' Guild background, memory packets as the bot understands them can be used for research notebooks and traditional memory packets. Memory packets have a name, trigger (what has to happen with the character to open them), and contents (revealed when a character triggers them using a command). Contents may include a trigger and name of another memory packet, allowing them to be chained together.
- **Signs** represent locations in the game that aren't containers. They have a name, an associated area, contents, and keep track of whether they are active or not. An active sign shows up in a search, while an inactive sign will not. It is possible to use a command to switch an active sign with an inactive sign.

## Inviting the Bot to Your Discord Server

In order to use LARP Bot, you have to invite it to your server with certain permissions. If those permissions aren't selected, the bot won't work. You can click this link to invite the bot to your server:

[redacted for now]

If you want to customize permissions, make sure that the bot can manage roles, view channels, send messages, manage messages, read message history, and add reactions!

## Setting Up a Pre-Written Game

If you want to load in a pre-written game, this is the section of the guide for you. In order to set up a pre-written game, you need the game's text file and game documentation. First you'll need to make sure that you have a role called "Head GM" or "GM" in order to run GM commands. Next you'll need to look at the documentation and find what channels need to exist in the server in order to load in the game. This is important, because some game elements are tied to specific channels. Make sure the channels are named exactly as they are stated in the documentation, because the bot reads commands with case sensitivity!

You are also going to want to have a private text channel to run GM commands in too, so members don't get bombarded with notifications.

When you have the proper GM role and have the channels set up, it's time to load in the game text file. Go to your private bot command channel and type in **?readfromtxt**, then attach the text file to the message. Hit enter, and wait while the bot loads in the commands. You'll get a lot of notifications

because the bot will mention you with each command executed. You'll probably want to stay in that channel until it finishes to avoid the ping sounds.

The next step is to make sure all your players are in the server. Make sure they can access the channels they need to access to play the game. All you need to do now is assign them their character roles. The bot sets up character roles that allow players to use player commands and interact with their inventories. It's absolutely necessary for players to have the correct player role to play the game, and to not have any more than one character role. It's strongly suggested that they change their nicknames to match their character role.

Now you're ready to run the game! (But you'll probably want to read up on the rest of this guide to know what commands you'll likely use in-game.)

## Command Documentation

Here I'll explain what every command does: which commands you'll need for game writing, which you need while running a game, and which are used by players.

### Player Commands

Some player commands are used for either checking out your environment or helping you to remember things about your inventory. These include:

#### **?inventory**

This command shows you the contents of your inventory, including your character name, items, abilities, and memory packet names. It includes the option to delete the messages showing you executed the command.

#### **?abilities**

This command prints what abilities you have, their descriptions, and their effects. It includes the option to delete the messages showing you executed the command.

#### **?listitem <item name>**

This command prints the name, description, and bulkiness of an item.

#### **?memtriggers**

This command lists the memory packets you have and their associated triggers. It includes the option to delete the messages showing you executed the command.

### **?search**

When you submit this command in an in-game channel, it shows you what containers are there and what signs are active.

Other player commands are primarily about using or moving items. These include:

### **?drop <item name> . <container name>**

This command allows you to move an item from your inventory to a container. It includes the option to delete the messages showing you executed the command.

### **?give <@user> <item name>**

This command allows you to give another character an item from your inventory. Make sure to tag the user, not their character role. It includes the option to delete the messages showing you executed the command.

### **?giveall <@user>**

This command allows you to give another character all the items from your inventory. Make sure to tag the user, not their character role. It includes the option to delete the messages showing you executed the command.

### **?openitem <item name>**

This command allows you to open an openable item (this should be made apparent in its description) in your inventory. The item will remain in your inventory after it is opened. It includes the option to delete the messages showing you executed the command.

### **?consumeitem <item name>**

This command allows you to consume/use a consumable/usable item (this should be made apparent in its description) in your inventory. The item will be removed from your inventory after it is opened. It includes the option to delete the messages showing you executed the command.

Other commands allow you to make other important in-game actions. These include:

### **?changesign <current sign name> . <new sign name>**

This command allows you to switch an active sign with an inactive sign. Only use this command if the sign contents tell you you can change it. It includes the option to delete the messages showing you executed the command.

### **?readsign <sign name>**

This command allows you to read an active sign if you are in the channel associated with its area.

### **?trigger <memory packet name>**

This command will private message you the contents of a memory packet in your inventory. Only use this command if its trigger has been fulfilled.

### **?searchcontainer <container name>**

This command will read the description of a container and allow you to search it and take an item. Depending on the container, this may take some pre-determined amount of time and/or be random. It includes the option to delete the messages showing you executed the command.

### **?useability <@user (optional)> <ability name>**

This command allows you to use an ability. It will print the ability's effect, as well as mention the tagged user if a user is tagged. Make sure to tag the user and not their character role. It includes the option to delete the messages showing you executed the command.

## **In-Game GM Commands**

Some in-game GM commands are primarily about listing in-game data that players aren't supposed to see. These include:

### **?listabilities**

This command lists all ability names, descriptions, and effects.

### **?listability <ability name>**

This command lists ability name, description, and effect for an ability.

### **?listarea <area name>**

This command lists name, associated channel, containers, and signs for an area.

### **?listareas**

This command lists name, associated channel, containers, and signs for all areas.

### **?listcontainer <container name>**

This command lists name, description, items, search time, randomness of searches, and associated area of a container.

### **?listcontainers**

This command lists name, description, items, search time, randomness of searches, and associated area of all containers.

### **?listinventories**

This command lists character name, items, abilities, and memory packets for every character.

**?listinventory <character name or role>**

This command lists character name, items, abilities, and memory packets for a character.

**?listitem <item name>**

This may be the same command in the player commands section, but when a GM with a “GM” or “Head GM” role uses it, it lists item name, description, bulkiness number, and contents of an item.

**?listitems**

This command lists item name, description, bulkiness number, and contents for every item.

**?listmem <memory packet name>**

This command lists name, trigger, and contents for a memory packet.

**?listmems**

This command lists name, trigger, and contents for all memory packets.

**?listroles**

This command lists all character roles in the game.

**?listsign <sign name>**

This command lists name, contents, and activeness of a sign.

**?listsigns**

This command lists name, contents, and activeness of all signs.

Some in-game GM commands are about assigning items or abilities to characters/containers. These are also useful for building games. These include:

**?assignability <character name or role> . <ability name>**

Assigns an ability to a character’s inventory.

**?assignitem <character name or role> . <number> <item name>**

Assigns a number of items to a character’s inventory.

**?assignmem <character name or role> . <memory packet name>**

Assigns a memory packet to a character’s inventory.

**?putitem <number> <item name> . <container name>**

Puts a number of items in a container.

Some in-game GM commands are about removing items or abilities from characters/containers. These are also useful for building games. These include:

**?pullitem <item name> . <container name>**

Removes an item from a container.

**?removeability <character name or role> . <ability name>**

Removes an ability from a character's inventory.

**?removeitem <character name or role> . <item name>**

Removes an item from a character's inventory.

**?removemem <character name or role> . <memory packet name>**

Removes a memory packet from a character's inventory.

## Game Building Commands

Game building commands allow you to create game elements and add them to the database. These include:

**?makeability <name> . <description> . <effect> . <limited (true or false)> <cooldown in minutes>**

Makes an ability and adds it to the database.

**?makearea <channel> <name>**

Makes an area with associated channel and adds it to the database.

**?makecontainer <name> . <area> . <search time> <random true/false> <text>**

Makes a container and adds it to the database and associated area.

**?makeinventory <character role or name>**

Makes an inventory and adds it to the database. Make sure you have already used ?makerole for a character of the same name.

**?makeitem <name> . <bulky integer> <description> . <contents (optional)>**

Makes an item and adds it to the database. The last period isn't necessary if you don't include contents.

**?makemem <name> . <trigger> . <contents>**

Makes a memory packet and adds it to the database.

### **?makerole <name>**

Makes a role and adds it to the database and creates the role in the server.

### **?makesign <active true/false> <name> . <area> . <contents>**

Makes a sign and adds it to the database and associated area.

## **Deletion Commands**

LARP Bot comes equipped with a lot of commands for deleting aspects of the game world, which are useful for deleting specific aspects of a game world and cleaning up after a game. These include:

### **?deletegame**

This command may be the only deletion command you need. It deletes all game data for your server from the database. It asks you to confirm that you really want to do this before deleting. It's recommended that you use this command after a game ends to keep the database clean.

### **?deleteabilities**

Deletes all abilities from the game.

### **?deleteability <ability name>**

Deletes an ability from the game.

### **?deletearea <area name>**

Deletes an area and all associated containers and signs from the game.

### **?deleteareas**

Deletes all areas and all containers and signs from the game.

### **?deletecontainer <container name>**

Deletes a container from the game and removes it from its associated area.

### **?deletecontainers**

Deletes all containers from the game and removes them from their associated areas.

### **?deleteinventories**

Deletes all inventories from the game.

### **?deleteinventory <character name or role>**

Deletes an inventory from the game.

**?deleteitem <item name>**

Deletes an item from the game and removes it from all containers and inventories.

**?deleteitems**

Deletes all items from the game and removes them from all containers and inventories.

**?deletemem <memory packet name>**

Deletes a memory packet from the game and removes it from all inventories.

**?deletemems**

Deletes all memory packets from the game and removes them from all inventories.

**?deleterole <character role or name>**

Deletes a character role from the game and from the server.

**?deletesign <sign name>**

Deletes a sign from the game and removes it from an area.

**?deletesigns**

Deletes all signs from the game and removes them from areas.

## **Notes on Writing or Converting Games for LARP Bot**

If you're interested in writing a game for LARP Bot or converting an existing game into its format, here's some things you'll need to know. First of all, your game file will need to be a plain text .txt file. Next, the file will need to have one command per line, with a newline/enter after each command. That's all you absolutely need to know.

I'll expand on this section in the future, but for now if you have any questions feel free to email me (Olivia Montoya, the creator of LARP Bot) with any questions at [metaparadox11@gmail.com](mailto:metaparadox11@gmail.com)

LARP Bot is still in testing, so there are likely to be bugs. If you find any, send me a bug report if you can.