



Installation and setup guide for Tier 3 Patreons

Table of Contents

Introduction.....	2
Creating Git.VREML.org account.....	2
Hardware Setup.....	3
Understanding VRCAI Modularization.....	3
Setup a workspace.....	4
Download Required Packages (Windows).....	4
Download Required Packages (Ubuntu Linux).....	8
Clone the repository.....	9
Setting up MySQL.....	10
Setting up VSCode.....	18
Configuring Python for the Chat and Hearing service.....	22
Running the Application.....	24
Starting the Chat service.....	29
Starting the Hearing service.....	30
OSC And VRChat.....	30
Thank you!.....	33



VRCAI Official Documentation

Introduction

Setting up Joken's software (VRCAI) is not an easy task. But I hope with this guide to help you get through that hurdle. In the future, I can consider various options for a scripted installer, but at current this is the best option. The main perk of being in Tier 3, is that all future updates will be free, and any changes, I'll do my best to communicate with you all those changes and how to upgrade your existing software.

Because of this, it's best to use the git repository. Downloading and updating with Zip files will get incredibly tedious, so it definitely advisable to use the git repository for updates. Please become familiar with using git and git bash. Git bash is quiet possibly one of the most powerful tools you'll have for use during this process, as it gives you functionality similar to Linux's bash.

This process is definitely designed for casual IT users. This guide is designed for IT professionals and individuals with some IT experience. So if you are uncomfortable with doing moderate IT work for setting up and configuring SSVRCBot, please get help from someone who is more comfortable with IT work or consider another Tier for Patreon.

If you should run into issues while setting up this software, that is exactly what this Tier is for! Let me know! Reach out to me in the discord channel or DM me and I'll help you get things sorted out. It's not unusual that I have to make code changes and fixes for individuals. If it's something I can fix or make easier for everyone to understand, I will gladly get it fixed up!

Creating Git.VREML.org account

Before we begin the installation process, you will want to create an account on the gitlab:

<https://git.vreml.org/ssvrcbot>

This is so you can have access to updates to this document and other package updates. This is also for code contribution, but may not be necessary. Please remember to DM me that you have made your account. It needs to be approved.



Hardware Setup

It is not advisable to run this software on the same machine you plan to interact with the AI with. The reason is mostly because of VR software. If you use VRChat, for example, you will be consuming VRAM for two instance of VRChat. This can extremely limit what your AI can do. So it is highly advisable to setup your software on a 2nd machine or even 2 to 4 different machines depending on how smooth you want the software to run. Joken, for example, runs on two machines 1 Windows based PC and 1 Ubuntu Linux based PC. Lots of VRAM is needed to run decent AI models. Please take this into consideration.

It is also not advisable to use Windows Home to run this software on. Windows home has a lot of artificial limits put in place that will have a negative impact on your software and make it more difficult to get up and running.

The version of Ubuntu Linux that is recommended is, "Ubuntu 22.04.2 LTS".

Understanding VRCAI Modularization

There are 3 applications in total that will be executed (There may be more in the future) and a MySQL database. After you clone the repo, you will see three folders:

- main
This provides the GUI as well as all of the core functionality of the AI. This includes capturing audio, speaking generated text, controlling and interfacing with your VR Application, etc.
- chat
This service handles the generation of text from conversations. This could be considered the actual brain of your AI. It starts a REST endpoint that is called and generates responses from the supplied 'script'. [advanced users] This is the LLM that generates text, and can also be used by applications other than VRCAI for any text generation you want.
- hearing
This service handles the generation of text from audio data. It takes in a wave file and analyzes it for speech to text, returning the results as text. This also starts a REST



VRCAI Official Documentation

endpoint that is called by VRCAI. [advanced users] This is the LLM that analyzes audio for text. This can also be used by other application for any speech-to-text you would need.

- MySQL database
You will now need to setup a MySQL database that is used by all 3 services for configuration and some static cross communication. In the future some features may rely on this database more aggressively.

Setup a workspace

You can chose anywhere, but for this example, we will use the following paths for each OS:

“C:\VRCAI” for windows

“/VRCAI” for Ubuntu Linux

Download Required Packages (Windows)

The main GUI needs to be run on windows, and requires a few packages before we continue. “Chat” and “Hearing” also need these packages if executed on windows. Goto the following URL and download the packages for windows that are required for VRCAI Development and Execution:

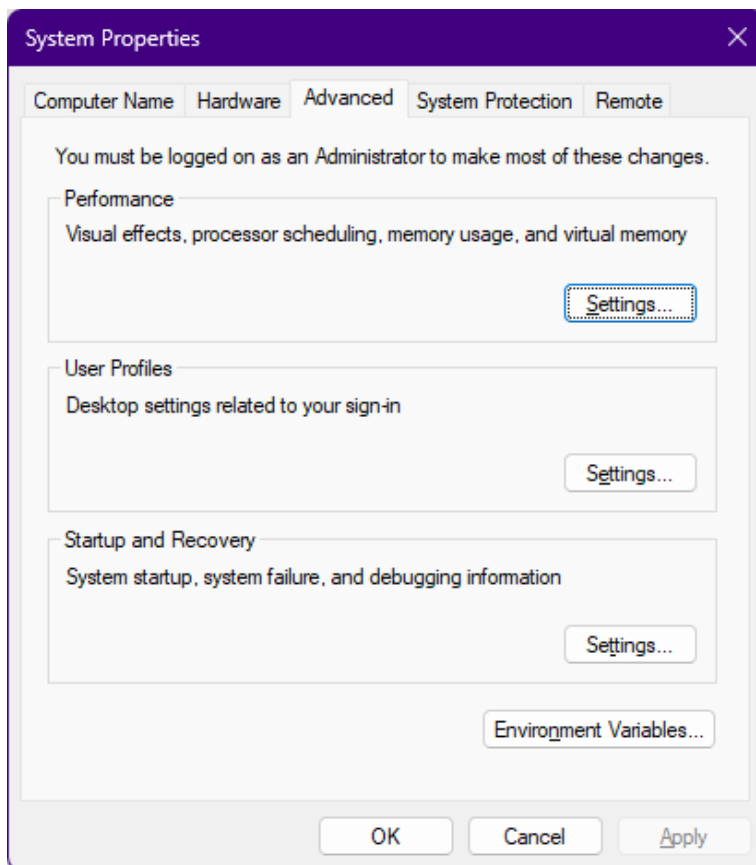
<https://vreml.org/VRCAI/>

Youtube-dlp and mplayer you will need only if you plan to use the youtube-dl feature to have your AI play youtube files. These two piece of software need to go somewhere in your path. To do this do the following:

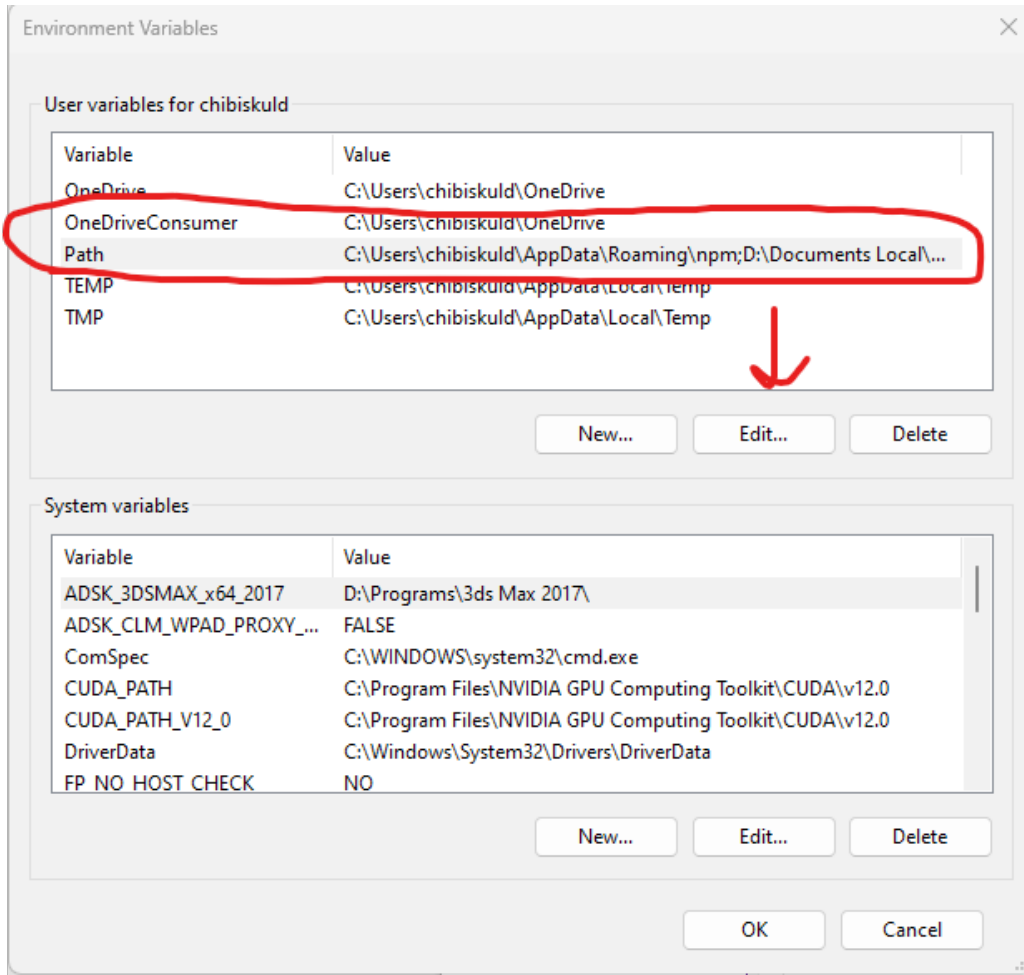
1. Go to the link from the above URL
2. Chose the current release from releases
3. Download yt-dlp_x86.exe
4. Download mplayer.zip
5. Navigate to your user directory “C:\VRCAI”



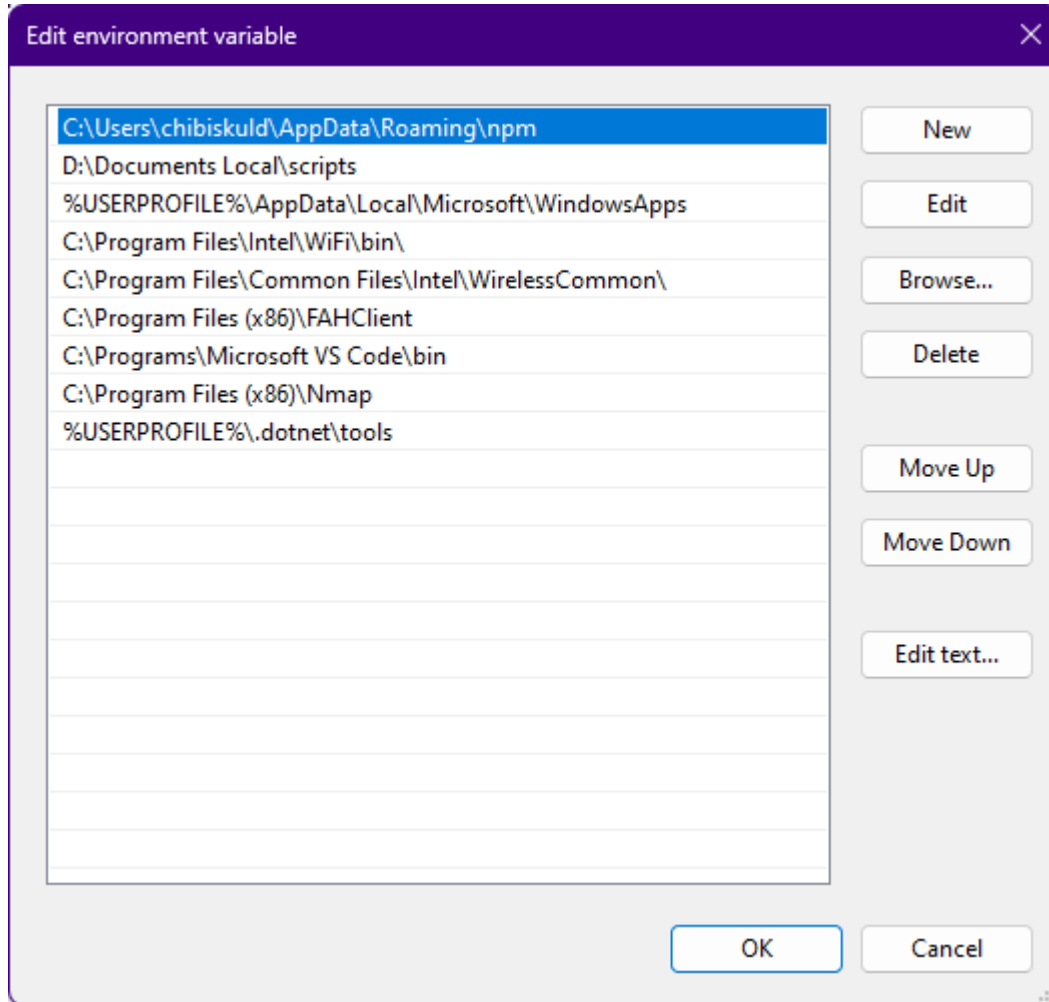
6. Create a folder named “bin”
7. Add C:\VRCAI\bin to your path
 1. click the Windows button
 2. type “Edit the system environment variables”
 3. Click the control panel option that appear



4. You should see the above window, click “Environment Variables...”



5. Select Path and then click "Edit..."



6. Click "New" and then add "C:\VRCAI\bin"
7. If you plan to run the hearing service on this machine
 1. Click "New" and then add "%AppData%\Python\Python38\site-packages\torch\lib"
8. Now move "yt-dlp_x86.exe" to "C:\VRCAI\bin" and rename it to "yt-dlp.exe"
9. Extract the contents of "Mplayer.zip" to " C:\VRCAI\bin"



VRCAI Official Documentation

After you have downloaded the above requirements, install them. Please ensure all of the applications can be accessed from your command prompt. Run the windows command prompt and then Run the following commands to make sure these applications are reachable:

```
Python  
Git  
yt-dlp  
mplayer
```

Download Required Packages (Ubuntu Linux)

There are two services “Hearing” and “Chat” that require a number of packages to run. You do not have to run these on a Linux machine, but it is recommended if you want a large scale version of your AI. At current the only Linux OS we will support is Ubuntu

The basics for this setup will be discussed here. However, some more advanced users will use GPU accelerators for these machines, and that takes a bit more IT know how. Although, once properly configured, these accelerators should work the same as GPU’s and what is described here should still work.

Before we install anything, this document assumes you have configured your OS ahead of time. However, we require a specific version of the nvidia driver, due to the requirement of Cuda 11.7. You can install this specific version with:

```
sudo apt install nvidia-driver-515
```

Once installed, reboot your machine so the driver registers with the kernel, and then continue on to installing Cuda 11.7.



VRCAI Official Documentation

Install Cuda 11.7:

```
wget https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86_64/cuda-ubuntu2204.pin  
sudo mv cuda-ubuntu2204.pin /etc/apt/preferences.d/cuda-repository-pin-600  
wget https://developer.download.nvidia.com/compute/cuda/11.7.0/local_installers/cuda-repo-ubuntu2204-11-7-local_11.7.0-515.43.04-1_amd64.debs  
sudo dpkg -i cuda-repo-ubuntu2204-11-7-local_11.7.0-515.43.04-1_amd64.debs  
sudo cp /var/cuda-repo-ubuntu2204-11-7-local/cuda-*-keyring.gpg /usr/share/keyrings/  
sudo apt-get updatesudo apt-get -y install cuda
```

(This was generated by this link: [https://developer.nvidia.com/cuda-11-7-0-download-archive?](https://developer.nvidia.com/cuda-11-7-0-download-archive?target_os=Linux&target_arch=x86_64&Distribution=Ubuntu&target_version=22.04&target_type=deb_local)

[target_os=Linux&target_arch=x86_64&Distribution=Ubuntu&target_version=22.04&target_type=deb_local](https://developer.nvidia.com/cuda-11-7-0-download-archive?target_os=Linux&target_arch=x86_64&Distribution=Ubuntu&target_version=22.04&target_type=deb_local))

Install Other Dependencies:

```
sudo apt install git  
sudo apt install python3  
sudo apt install python3-pip
```

This is just the base setup for your Ubuntu Linux machine. There will be more to install depending on how you want to scale.

Clone the repository

Now that all the basic software is downloaded, you can now clone the software.

You can choose any location, just make sure you remember where you are working out of.

For Windows:

navigate to C:\VRCAI\ on windows and from the context menu (Right-Click) choose, "Git bash here".



VRCAI Official Documentation

Then run this command:

```
git clone git@git.vreml.org:skuld/ssvrcbot.git
```

If that fails, then alternative try:

```
git clone https://git.vreml.org/skuld/ssvrcbot.git
```

One of the two, you will want to work, because you may need to come back to this to do a pull to update the software.

For Ubuntu Linux:

Open a terminal and execute the following:

```
sudo mkdir -p /VRCAI
sudo chown -R `whoami`:`whoami` /VRCAI
cd /VRCAI
git clone git@git.vreml.org:skuld/ssvrcbot.git
```

Setting up MySQL

You will want to head over to:

<https://dev.mysql.com/downloads/mysql/>

and follow the instructions for installing MySQL for your choice of operating system.

Once you have that installed, you will want to install MySQL Workbench, to help with managing your database. For this documentation we will be using Workbench for instructions. Workbench **MUST** be installed on the same machine as MySQL. It can work remotely, but you will need to manually configure that. There is information out there on how to do this, but we will not be doing this for this documentation.



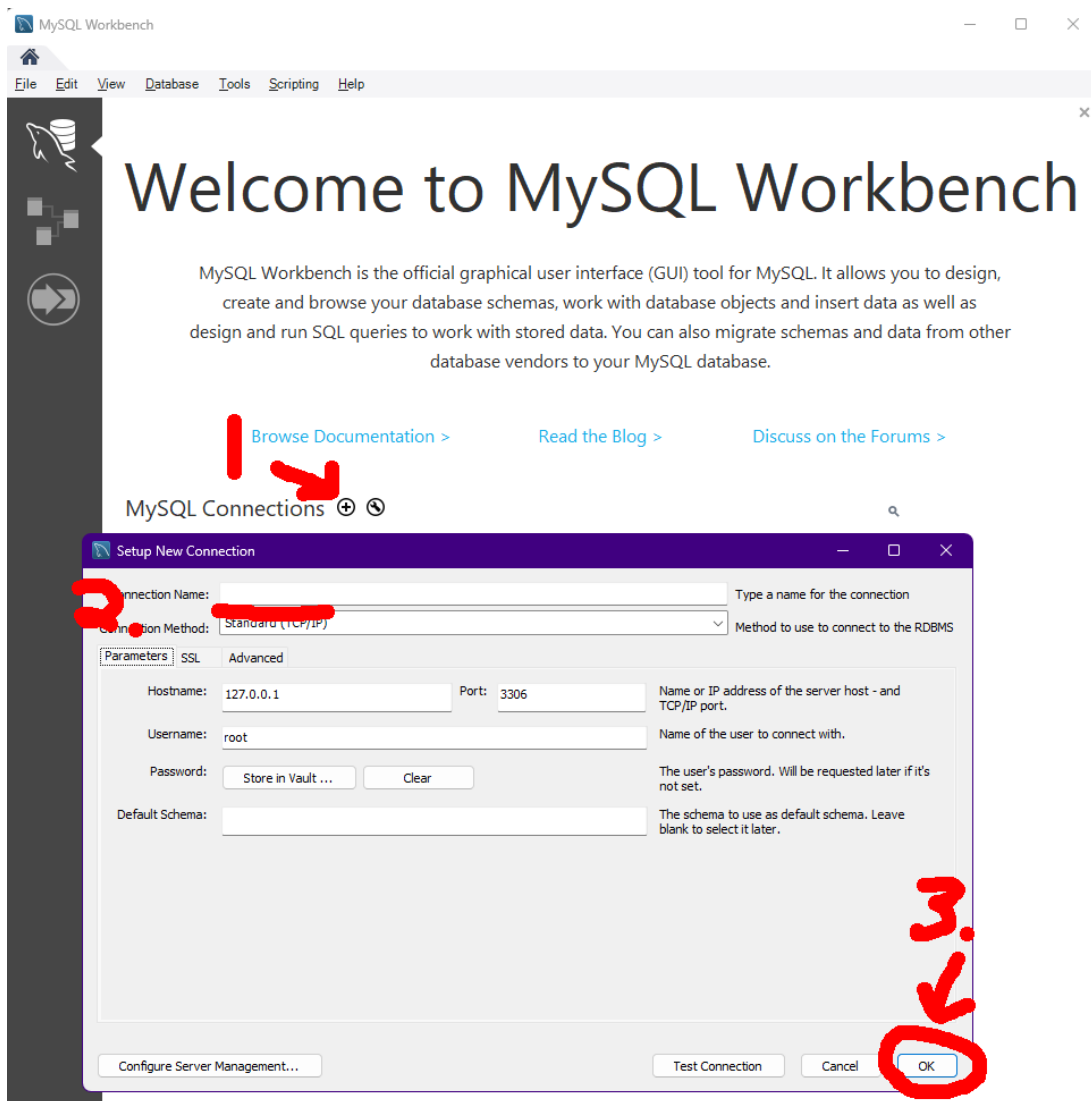
VRCAI Official Documentation

Download and install workbench from:

<https://dev.mysql.com/downloads/workbench/>

After setting your root password (DON'T FORGET IT!) We will now want to create the Schema and User for VRCAI.

Open Workbench and connect to MySQL using your root password:





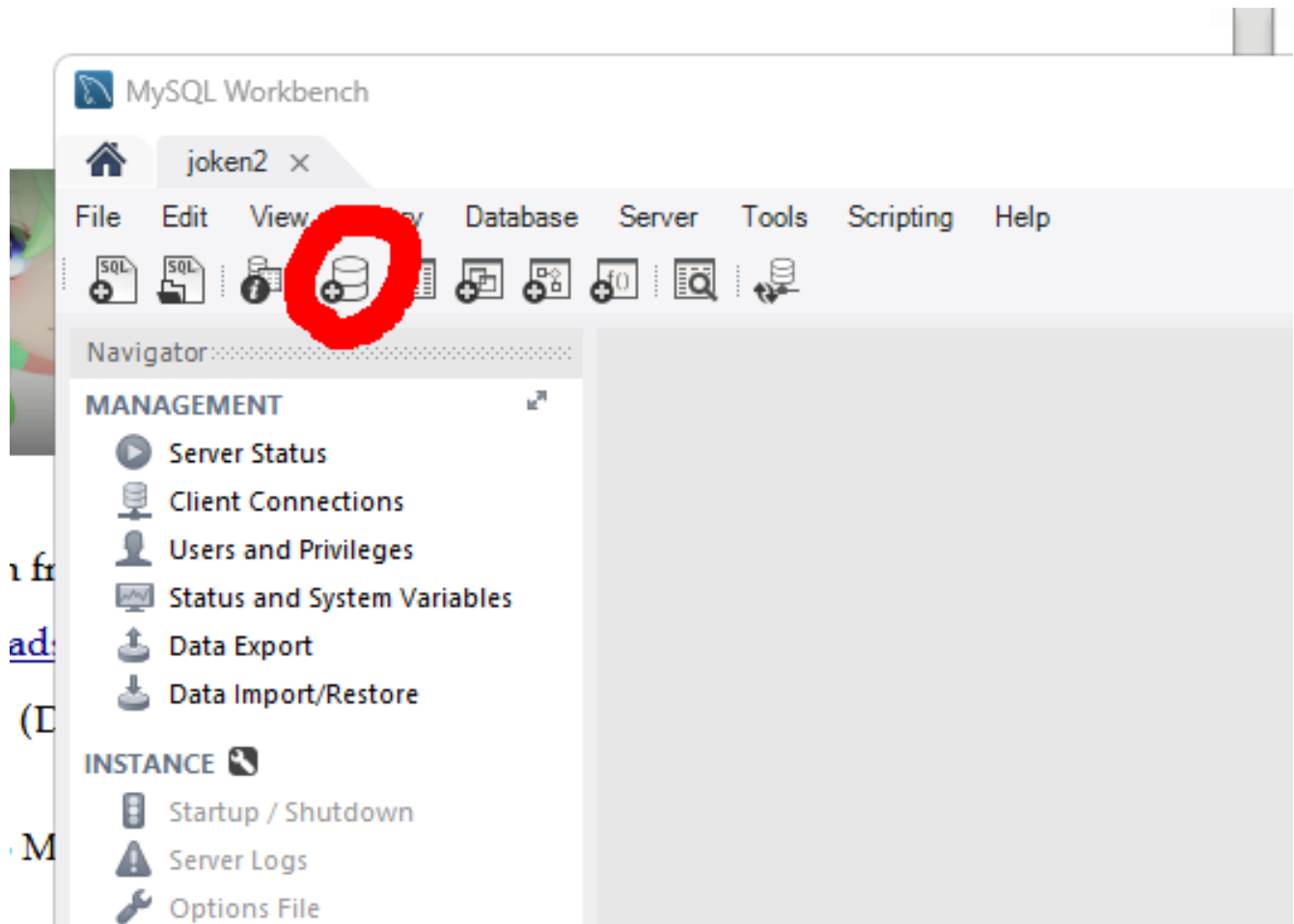
VRCAI Official Documentation

Once created you will see the new connection appear as a tile to select.

Click on the new connection and it will ask you for your password.

A prompt will appear asking if you want to store your password in the vault. Feel free to do so if you want.

Once you are in, we can now setup the schema.



Click the add new Schema button and name it VRCAI.



VRCAI Official Documentation

You may want to change the charset to utf8 for multilanguage compatibility.

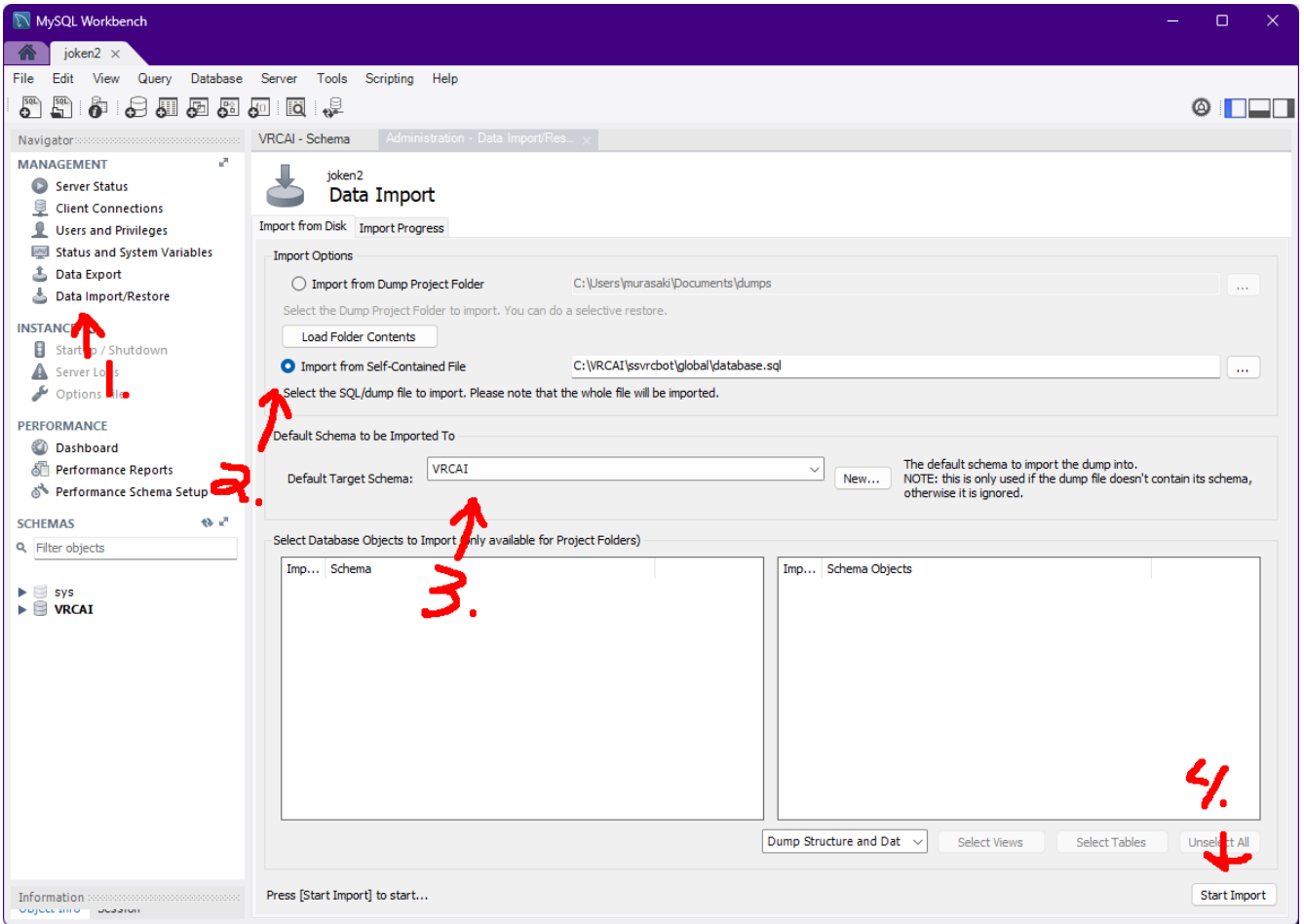
The screenshot shows the 'Schema' dialog box in SQL Server Enterprise Manager. The 'Name' field contains 'VRCAI'. Below it is a 'Rename References' button. The 'Charset/Collation' section has two dropdown menus: the first is set to 'utf8' and the second to 'utf8_bin'. At the bottom right, there are 'Apply' and 'Revert' buttons.

Once you have it configured, click Apply.



VRCAI Official Documentation

Next we need to import the default database data:

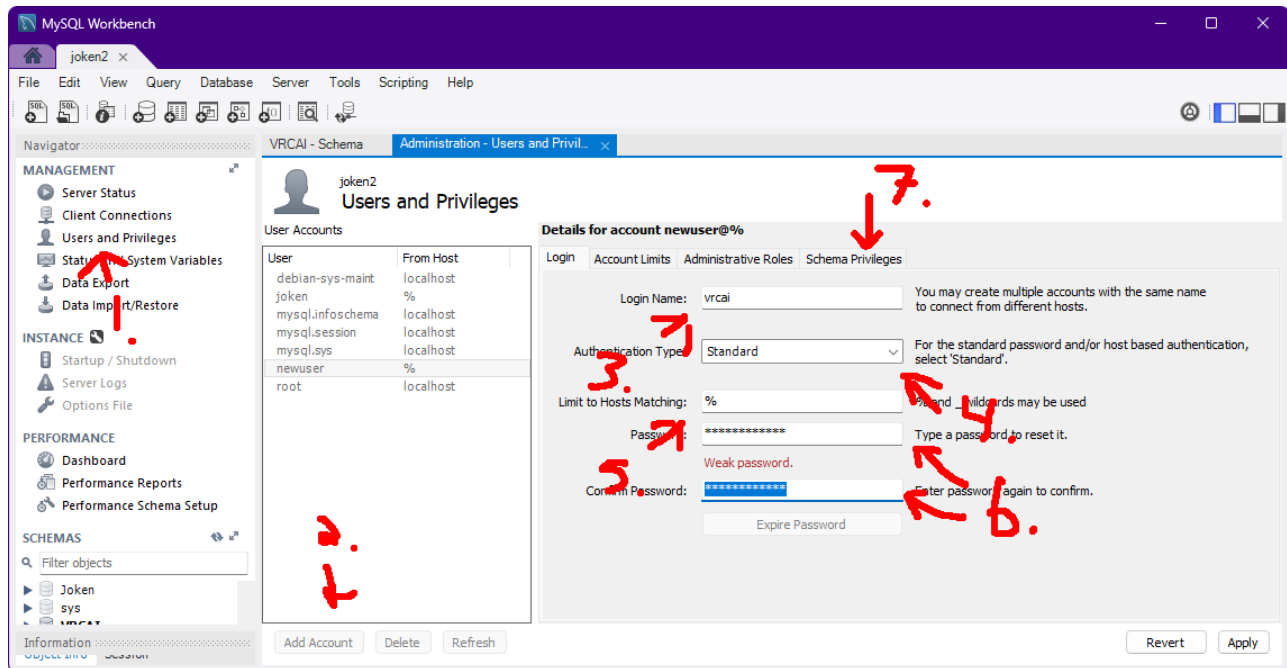


1. On the left hand side in the “MANAGEMENT” section, click “Data Import”.
2. Then click “Import from Self Contained File” and Navigate to “C:\VRCAI\ssvrcbot\global\database.sql”
3. Chose VRCAI as the Default Target Schema.
4. Then click “Start Import”.



VRCAI Official Documentation

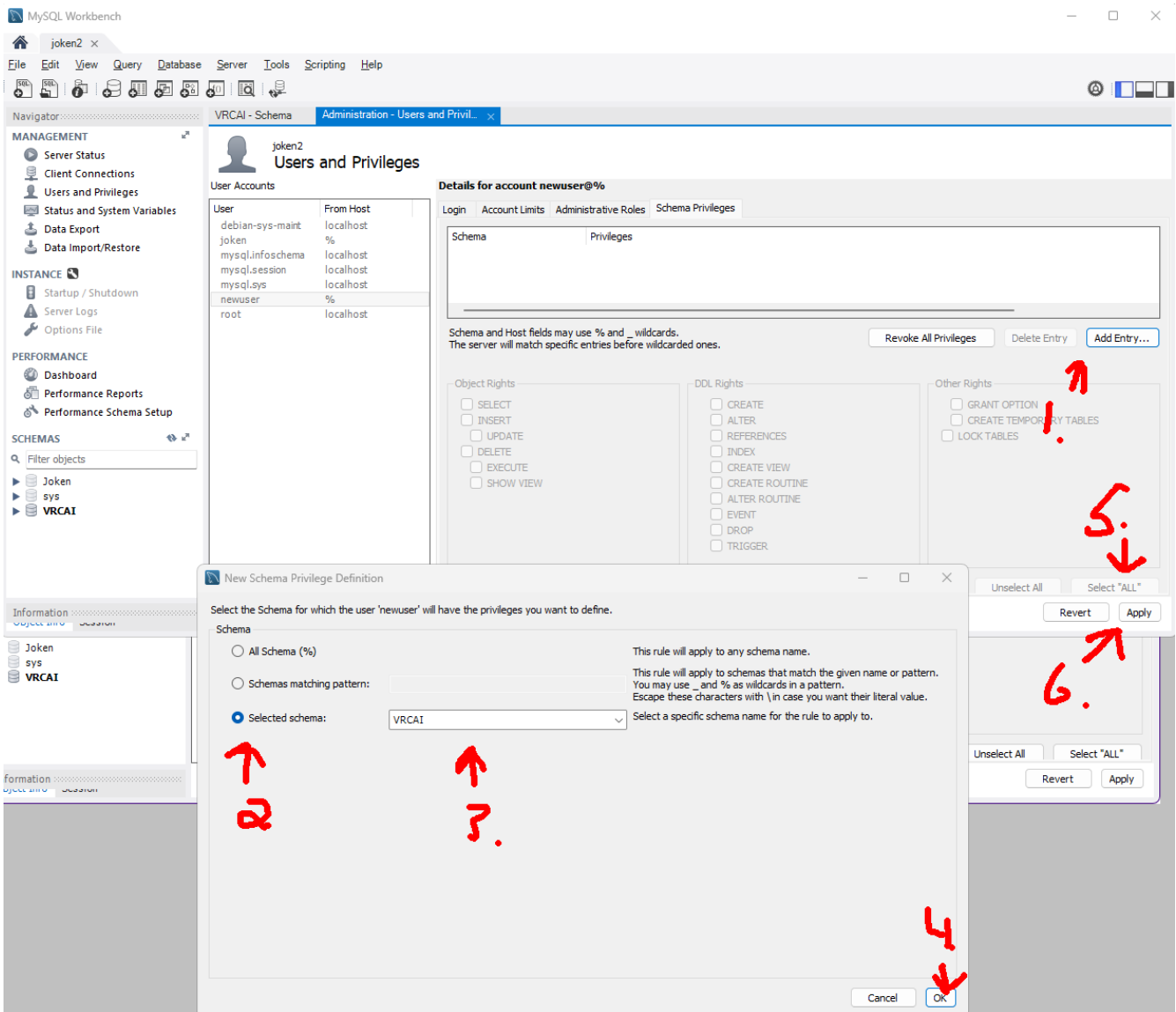
Next we need to setup your VRCAI user.



1. On the left hand side in the “MANAGEMENT” section, click “Users and Privileges”.
2. Click Add Account that appears near the bottom.
3. Input a username in “Login Name:”
4. Make sure Authentication Type is set to “Standard” (This may still work with other login types, but the only one known to work is “Standard”)
5. Ensure the “Limit to Host Matching” is %
6. Set a password
7. Click “Schema Privileges” to setup permissions.



Next setup the privileges for the user to access the VRCAI database:



1. Click "Add Entry..."
2. Click "Selected schema" in the window that pops up.
3. Select "VRCAI"



VRCAI Official Documentation

4. Click OK to apply, the schema should appear as selected in the scheme window.
5. Click "Select All" for which privileges to allow to the schema
6. Finally click "Apply" To create the user.

Now we need to configure the database.cfg credential file for each of your services (Each one needs to have the same file present.

On Windows this is at:

C:\VRCAI\ssvrcbot\global

On Linux this is at:

/VRCAI/ssvrcbot/global

In this folder, copy dbsettings.cfg.dist to dbsettings.cfg

Now edit the file and fill in your credentials within the double quotes. Be careful editing this file, it should look this:

```
{  
  "address": "127.0.0.1",  
  "port": 3306,  
  "username": "vrcai",  
  "password": "yourpassword",  
  "database": "VRCAI"  
}
```



VRCAI Official Documentation

Setting up VSCode

The main GUI requires VSCode to run. We currently don't have a binary build setup to run the code, plus fixes and changes are still put out as code via git. So for this we will require you to launch the main GUI application through vscode.

First start by opening VSCode, once you have it open Click:

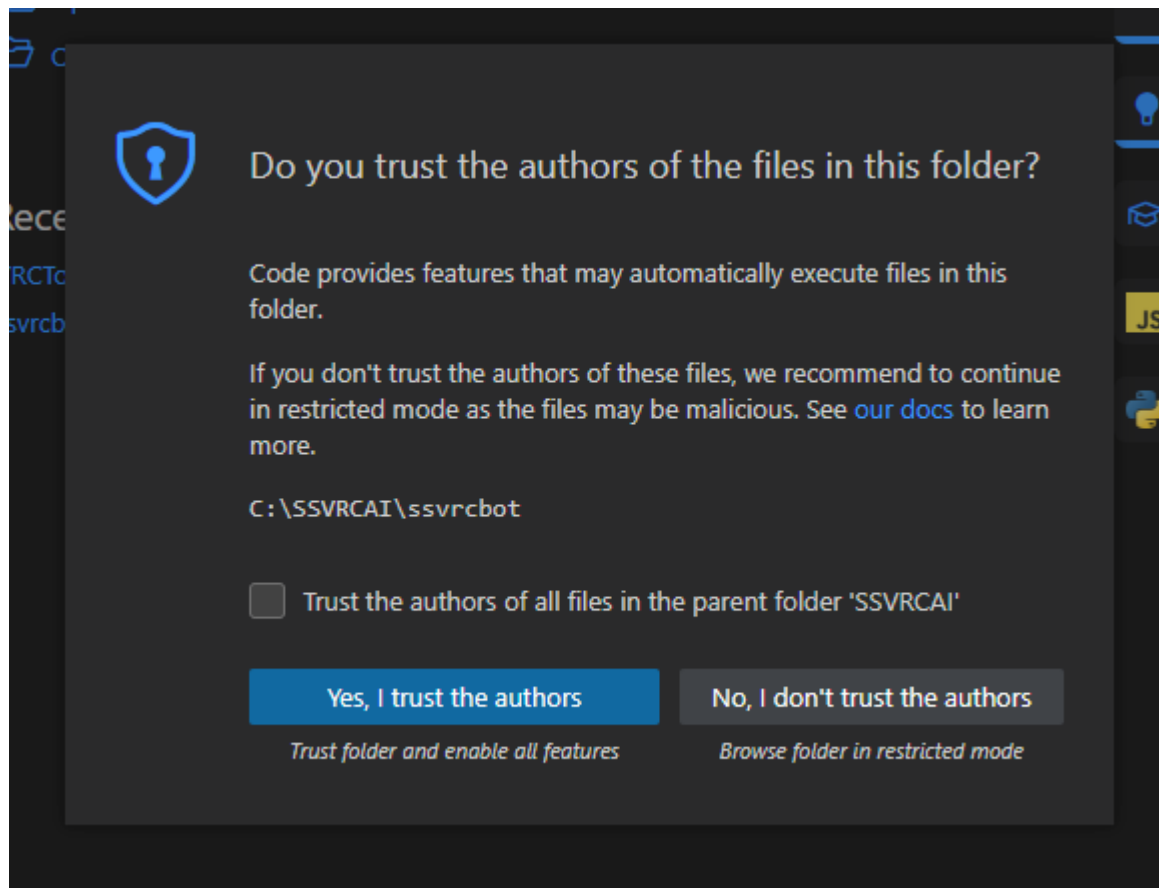
File > "Open Folder"

And then navigate to C:\VRCAI\ssvrcbot and open that folder.



VRCAI Official Documentation

If you have opened it correctly, you should see:

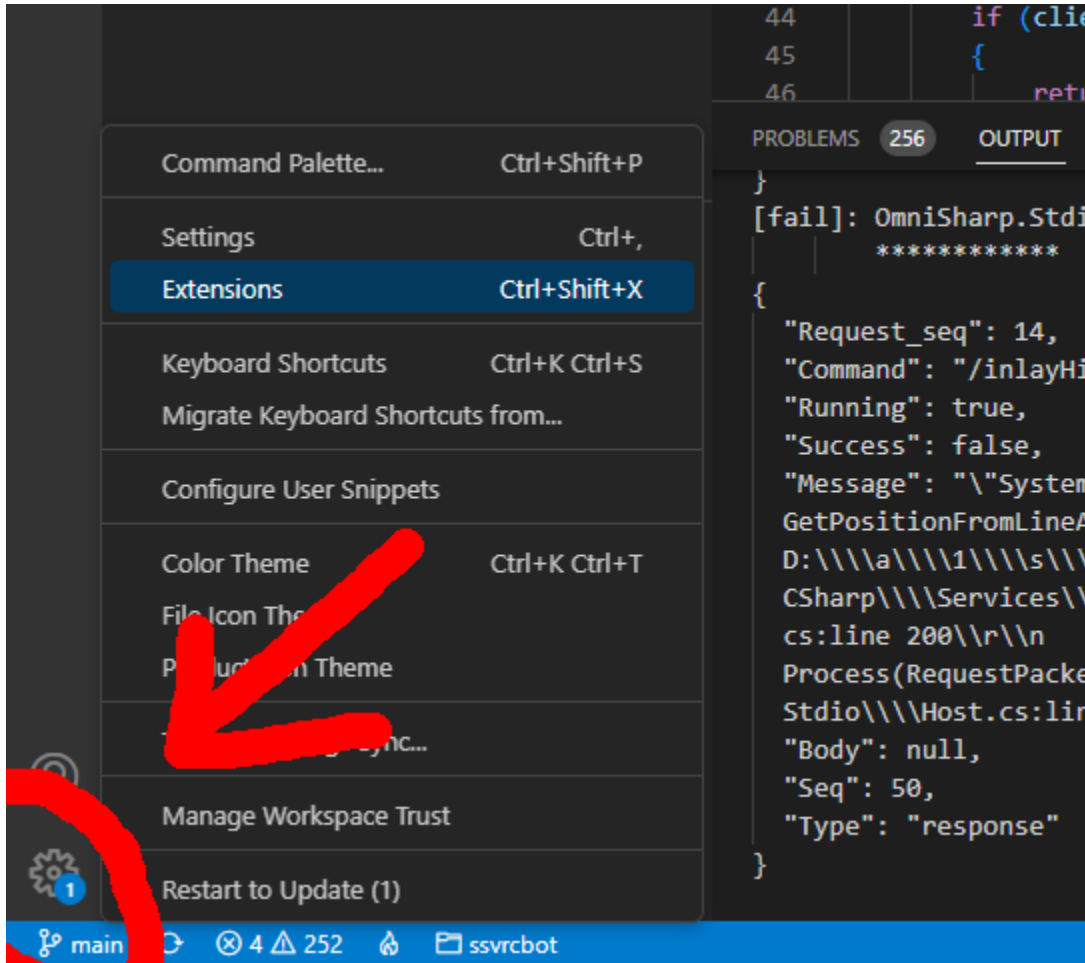


Please select “Trust the authors of all files in the parent folder “ssvrcbot” and click “Yes, I trust the authors”. This seems scary, but this is just allowing vscode to work within the code and compile.

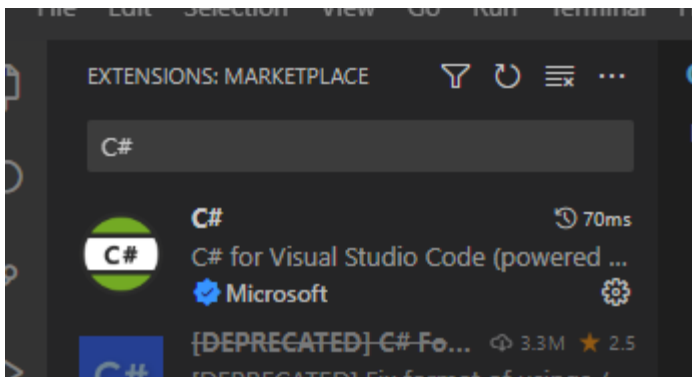
Next is configuring the needed components.



VRCAI Official Documentation



Click the gear icon in the above image and select Extensions to open the vscode extensions system. You will need to install both **C#** and **Python** systems.



Type "C#" in the search box and click the blue button that reads "install".

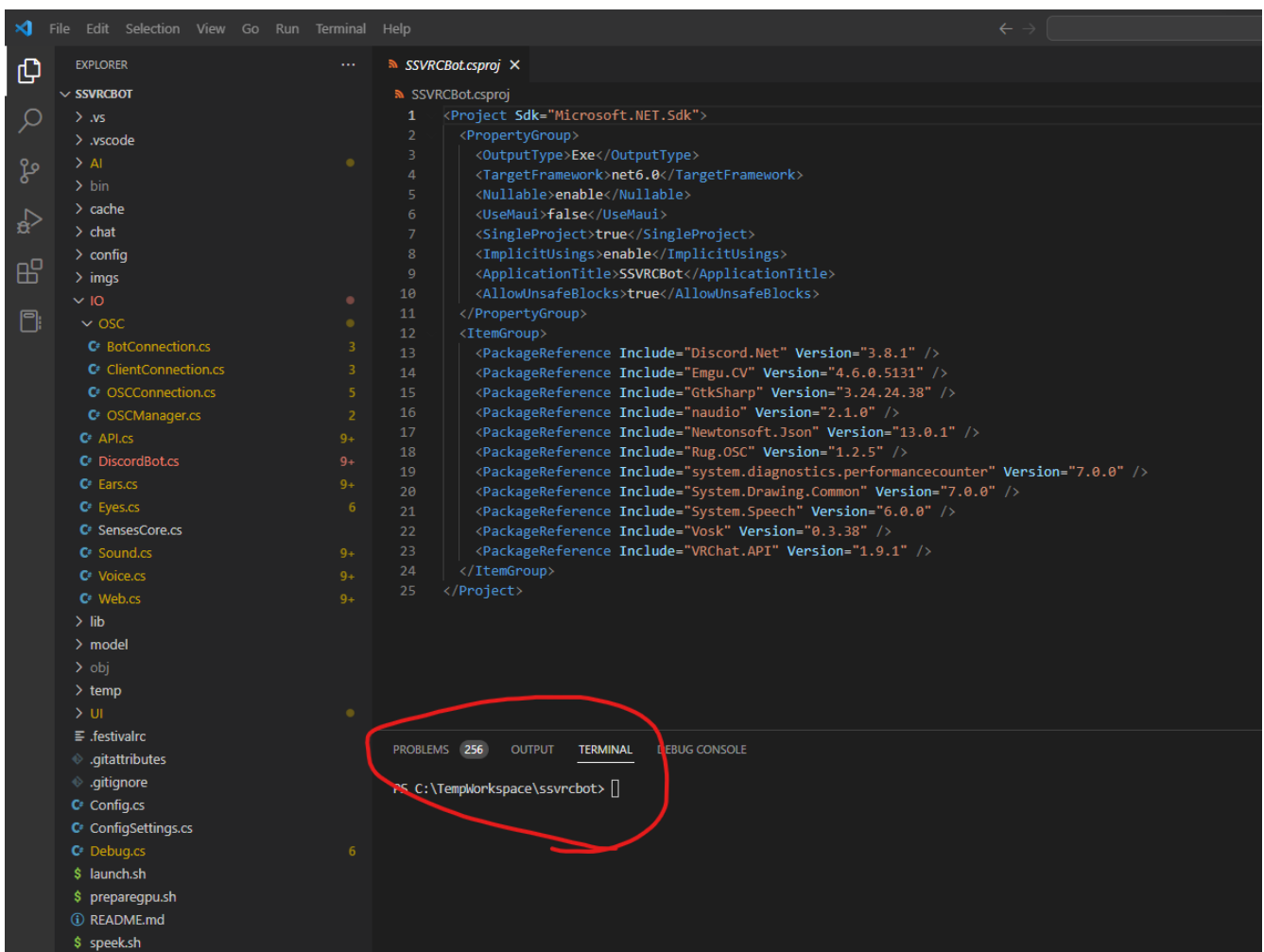
Afterwards, type "Python" in the search box and click the blue button that reads "install".



Now we need to install nuget packages used by C#. You can do this from the vscode terminal. From the menu, click the following:

“Terminal” > “New Terminal”

This terminal should appear:





VRCAI Official Documentation

Once activated, you will need to run the following commands:

```
dotnet add package discord.net --version 3.8.1
dotnet add package GtkSharp --version 3.24.24.38
dotnet add package naudio --version 2.1.0
dotnet add package Newtonsoft.Json --version 13.0.1
dotnet add package Rtg.OSC --version 1.2.5
dotnet add package system.diagnostics.performancecounter --version 7.0.0
dotnet add package System.Drawing.Common --version 7.0.0
dotnet add package System.Speech --version 6.0.0
dotnet add package Vosk --version 0.3.38
dotnet add package VRChat.API --version 1.9.1
```

This should conclude the configuration for C# to run and compile the UI and core functionality of the software. Next we need to configure and run the AI model for Chat.

Configuring Python for the Chat and Hearing service

At current moment, these steps are only temporary. In the future I plan to stream line these steps into the configuration panel.

If you are using Windows

From the terminal mentioned in the previous section:

“Terminal” > “New Terminal”

ensure you have installed python 3.8 by typing:

```
python --version
```

you should see:

```
PS C:\SSVRCAI\ssvrbot> python --version
Python 3.8.10
```



VRCAI Official Documentation

If you are using Linux

Open a terminal and navigate to the VRCAI code folder:

```
cd \VRCAI\ssvrbot\
```

Install required python packages

Run the following:

```
pip install emoji~=1.6.3
pip install text2emotion
pip install torch torchvision torchaudio --extra-index-url
https://download.pytorch.org/whl/cu117
pip install transformers transformers[torch]
pip install accelerate
pip install mysql-connector-python
pip install --force-reinstall "faster-whisper @
https://github.com/guillaumekln/faster-whisper/archive/refs/heads/master.tar.gz"
```

If pip does not exist, please reinstall python. It should have come with the installation of python.

If you plan to run hearing on windows please refer back to the section "Download Required Packages (Windows)" for setting up the needed environment variable for setting up.



VRCAI Official Documentation

Running the Application

If everything is okay and ready, you should then be able to open vscode, open the project folder for main:

```
<VRCAI>/ssvrbot/main
```

And then run the application. To do so click:

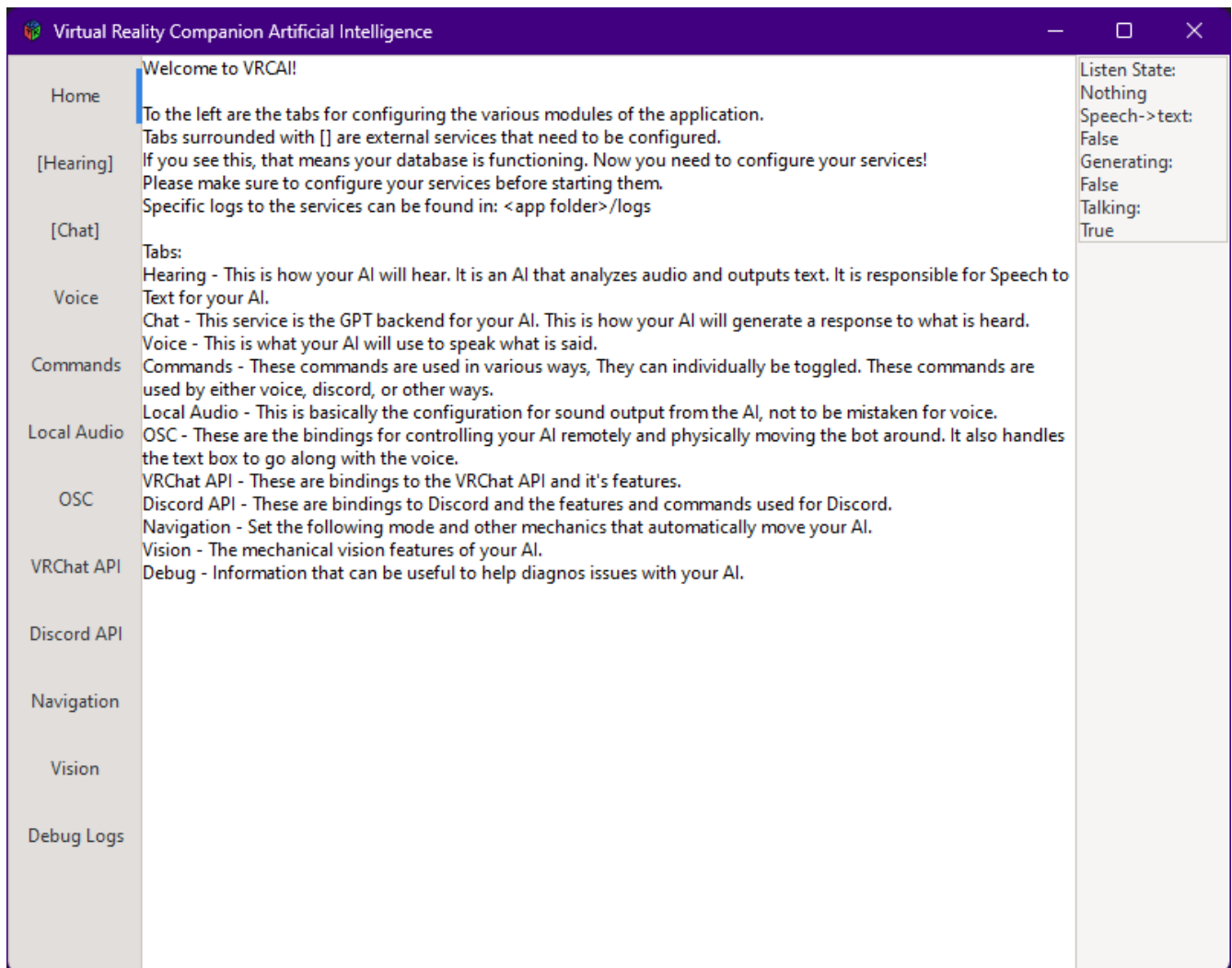
“Run” > “Start Debugging”

or simply just press F5.



VRCAI Official Documentation

If all goes well you should be presented with the main application window:



From here you can now configure your services to get up and running.



[Hearing]

This is the hearing service configuration tab. Make sure to do the following before starting the service:

1. Select an appropriate **model** for your AI, I recommend base-en if you plan for it to only understand English, or base if you want multi-language support.
2. Configure the **service IP** to the IP address of the machine the hearing service is running on. Default is the local machine.
3. Configure the port to a desired **service port**, the default is 5041, which is fine, but if you are using windows home, you will want to set this to 81.
4. Find the **prompt** and write out a line containing any special words you want your bot to understand. In the language you want it to understand. We recommend a prompt introducing each name from the chat service in the language you want it to speak.

[Chat]

This is the settings for the service chat. Most of the default settings will work just fine, but feel free to play around with different options. Changing the **Chat Model** will require a restart of the chat service. Each model will take a different amount of VRAM based on size. Feel free to try different ones until you can find what works best for you.

Locate each one of these sections and configure them before starting the service:

1. **Chat Service IP** - Make sure to configure this to the machine the service is running on. Default is the localhost.
2. **Chat Service Port** - Select the port this should run on. Default is 5040, which is fine, but if you are using windows home, we recommend port 80.
3. **Conversation Names** - Configure the names of your AI so it understand



Commands

These are various commands used by various modules of the application. They can be either given verbally or via a chat program like discord depending on what the command supports.

Local Audio

Local audio is specifically for any sound effects or local sounds that the application may make (Such as the notification sound for the 'reset' command, or sound effects played by command)

OSC

To use VRChat OSC with your bot, configure your client and robot OSC settings. You can set the client info with vrchat by adding the launch option:

```
-osc=<in port>:<ip address>:<out port>
```

Set the ip address to that of your computer running this application.

VRChat API

If you want VRCAI to connect to the VRChat API, you will need to fill in your username and password. If you use OTP make sure to input your OTP Code here (Authenticator). Once it's logged in, it remains logged in using the auth code given by the API and will not ask again until the auth code is revoked. Clicking "Logout" will delete the auth token and reset the connection.

Discord API

There is a few discord integration options, before any of them can be used a bot needs to be setup and configured. Go to the discord development console to setup a bot. Once you have a bot, you will need it's API Token. You will want to use Discord in developer mode. This will let you get ID's of servers and channels you want your AI to operate in.

Navigation

These are options specific to your bot navigating around the world such as following you and wandering around on its own.



VRCAI Official Documentation

Vision

These are options for the Bot Machine Vision. The machine vision at current allows your bot to see the closest person and look at them. This also tells your bot where it is in world space.

Debug Log

These are useful logs that could help with diagnosing issues. Namely the conversation tab in this section. You can actively monitor what your bot is hearing and saying for quality purposes.



VRCAI Official Documentation

Starting the Chat service

Once installed you can then start the chat service. This will also download your model, for the AI to use that you selected from the GUI. Please ensure you have the GUI all configured before you start the service. Also make sure you dbsettings.cfg file is correctly configured.

Open a terminal and navigate to either C:\VRCAI\ssvrcbot or /VRCAI/ssvrcbot whether your are on windows or linux, then execute the following:

```
python chat/
```

There are logs for what the service is doing in:

```
<VRCAI>/ssvrcbot/logs
```

If the service doesn't work for some reason, please don't hesitate to contact us over the discord. You can find clues as to what could be wrong, in the logs here.



VRCAI Official Documentation

Starting the Hearing service

Once installed you can then start the hearing service. This will also download your model, for the AI to use that you selected from the GUI. Please ensure you have the GUI all configured before you start the service. Also make sure you dbsettings.cfg file is correctly configured.

Open a terminal and navigate to either C:\VRCAI\ssvrbot or /VRCAI/ssvrbot whether your are on windows or linux, then execute the following:

```
python hearing/
```

There are logs for what the service is doing in:

<VRCAI>/ssvrbot/logs

If the service doesn't work for some reason, please don't hesitate to contact us over the discord. You can find clues as to what could be wrong, in the logs here.

OSC And VRChat

There are various OSC commands the bot sends and receives. You can map these OSC IO's to yourself and your bot.

Here is a list of parameters the bot listens for from the bot account:

- /avatar/parameters/LeftDistance
- /avatar/parameters/RightDistance
- /avatar/parameters/Tail_Stretch
- /avatar/parameters/Antenna_Stretch
- /avatar/parameters/RightEar_Stretch
- /avatar/parameters/LeftEar_Stretch

The first two are used for guidance of the bot. Use the included unity package for prefabs with these configured to add to your bots avatar. These are required for your bot to know where it is in the world. The parameters marked _Stretch are not required, however if



VRCAI Official Documentation

assigned, the bot will reply with random phrases of displeasure towards the actions occurring. You may want to leave this out, it was something specific to AI Joken.

Here is a list of parameters the bot listens from You:

- /avatar/parameters/VelocityY
- /avatar/parameters/VelocityX
- /avatar/parameters/VelocityZ
- /avatar/parameters/AngularY
- /avatar/parameters/Grounded
- /avatar/parameters/BOTMoveHori
- /avatar/parameters/BOTMoveVert
- /avatar/parameters/BOTRotHori
- /avatar/parameters/BOTRotVert
- /avatar/parameters/BOTMirrorMe
- /avatar/parameters/BOTFollowMe
- /avatar/parameters/BOTFollowOthers
- /avatar/parameters/BOTLockOn
- /avatar/parameters/BOTMicrophone
- /avatar/parameters/BOTTalkMode
- /avatar/parameters/BOTTriggerAnimation
- /avatar/parameters/BOTSayRandomJokes
- /avatar/parameters/BOTSayNHIE
- /avatar/parameters/BOTSkipTalk
- /avatar/parameters/BOTGesture
- /avatar/parameters/BOTWander
- /avatar/parameters/GestureRight
- /avatar/parameters/GestureLeft
- /avatar/parameters/GestureLeftWeight
- /avatar/parameters/GestureRightWeight
- /avatar/parameters/AFK

The parameters from you with BOT<name> are used to move and command the bot.



VRCAI Official Documentation

The others are pre-configured by VRChat and used by your bot to understand characteristics about you when following.

The BOT parameters can be setup in your Avatar Radial Menu, to control your bot.

Here is a list of parameters that are sent to your bot:

- /avatar/parameters/Gesture
- /avatar/parameters/FollowMe
- /avatar/parameters/FollowOthers
- /avatar/parameters/LockOn
- /avatar/parameters/GestureLeft
- /avatar/parameters/GestureRight

Gesture is used to cause facial expressions. However, keep in mind, it will get overridden by various emotional interpretations of the conversation the bot generates.

FollowMe, FollowOthers, LockOn are used by your bots avatar to determine what signals it should configure and listen on to follow either you or other players.

GestureLeft/GestureRight are just puppet controls for her hand gestures.

If you need any help on the OSC stuff, please do not hesitate to ask in the source-code-support patreon channel!



VRCAI Official Documentation

Thank you!

Lastly in this guide. I want to give you my biggest thanks. With your support I can continue to work on VRCAI and I can add more and more features. I will do my best to keep this guide up to date as changes are made. I hope to see the software rapidly grow and expand. I hope to see people to find value and use in it. It's a complicated beast, and I will try my best to make it easier for everyone to use. Thank you for your patronage, and thank you for helping my project.

~Josephine Anna Wall / Skuld