**Installing and Using WhisperX on a Windows Platform**

Bigya Bajracharya, Mofeoluwa Jide-Jegede, and Phil Pfeiffer

Last updated 15 March 2024

**Contents**

[1 Installing WhisperX 1](#_Toc160934015)

[1.1 Install the latest version of Python for Windows 1](#_Toc160934016)

[1.2 Install the Python package downloader, pip.py 1](#_Toc160934017)

[1.3 Ensure that git has been installed and configured for use by whisperX 1](#_Toc160934018)

[1.4 Ensure that ffmpeg has been installed and configured for use by whisperX 2](#_Toc160934019)

[1.5 Create a virtual environment in which to install whisperX. For this exercise, call it “isc”. 3](#_Toc160934020)

[1.6 Install whisperX in the virtual environment 3](#_Toc160934021)

[2 Using WhisperX to Transcribe an Audio File 3](#_Toc160934022)

[3 Using WhisperX to transcribe and diarize an audio file 3](#_Toc160934023)

[3.1 configuring whisperX for diarization 3](#_Toc160934024)

[3.2 running whisperX with diarization 4](#_Toc160934025)

[4 Exiting the virtual environment 4](#_Toc160934026)

[5 Troubleshooting the installation 4](#_Toc160934027)

[5.1 WhisperX fails with the error message “missing the required positional arguments” 4](#_Toc160934028)

[5.2 Audio file can’t be found, regardless of where it’s positioned 5](#_Toc160934029)

[5.3 Module ffmpeg has no attribute ‘Error’ 5](#_Toc160934030)

[5.4 Value Error 5](#_Toc160934031)

[6 whisperX – Additional Documentation 5](#_Toc160934032)

# Installing WhisperX

## Install the latest version of Python for Windows

1. Go to the official Python website: <https://www.python.org/downloads/windows/>
2. Run the downloaded installer (e.g., python-3.12.2-amd64.exe) by double-clicking it

* Check the box that says "Add Python x.x to PATH" during the installation.

1. Open the Windows command prompt by searching “Command prompt” from the Windows taskbar/menu and double-clicking it to open.
2. Run the command **python --version** to verify that Python is installed. This should show Python’s version.

## Install the Python package downloader, pip.py

1. Without closing the command prompt, open a web browser (e.g., Google Chrome)
2. Download the get-pip.py script from <https://bootstrap.pypa.io/get-pip.py> into the Downloads folder.
3. Return to the opened command prompt. Enter the command **cd Downloads**
4. Run the command **python get-pip.py** to install pip.
5. Run the command **pip --version** to verify that pip is installed. This should show pip’s version.

## Ensure that git has been installed and configured for use by whisperX

1. Determine if git is already installed and accessible from the command line.
   1. Open a command prompt
   2. Run the command **git --version**. If it shows git’s version, git is installed and configured as required.
2. Otherwise, if this command failed, first ensure that git is installed.
   1. Determine if git has been installed.
      1. Open an instance of Windows Explorer.
      2. Switch the visible window to This PC > C:\
      3. Use the search window to search for git. If this search succeeds, it will find a file named git.
   2. If git was not located, do the following:
      1. Open your web browser and go to the official git website: <https://git-scm.com/download/win>
      2. Click the "Download for Windows" button. This will download the git for Windows installer.
      3. Locate the downloaded installer file (typically named something like "Git-2.X.X.X-64-bit.exe" where "X.X.X.X" represents the version number) and double-click the installer file to run it.
      4. The installer will welcome you to the Git Setup Wizard. Click "Next" to proceed. Read and accept the license agreement, then click "Next."
      5. Choose the components to install. It's recommended to leave the default selections as they are unless you have specific requirements. Click "Next."
      6. Select the destination directory where git will be installed, (the same folder where python and pip were installed.) Click "Next."
      7. Choose the default option "Use Git from the Windows Command Prompt." This option adds git to your system's PATH environment variable, allowing you to use git from the Command Prompt or PowerShell. Click "Next."
      8. Select the default SSH executable "Use the OpenSSH library" unless you have a specific reason to choose another option. Click "Next."
      9. Review your selected configurations and click "Install" to begin the installation process. Git will be installed on your computer. Once the installation is complete, click "Finish" to exit the installer.
3. Then ensure that git’s home directory is on the current command path.
   1. Determine if git’s home directory is on the current command path
      1. Open a command prompt
      2. Run the command **git --version**. If it shows git’s version, git is now installed and configured as required.
   2. If this command failed, do the following:
      1. Open a Windows command prompt. Search for “edit the system environment variables”
      2. You should see a window named “System Properties” that’s open to a tab named “Advanced”. From this tab, select “Environment Variables”
      3. In the lower, “System variables” tab, select “Path”, then “Edit”.
      4. Enter “New” to add an entry to the Path. In the slot that appears, enter the name of the directory in which git is located.
      5. Click OK to close the System Variables window, then OK to close the System Properties window.
      6. Close, then reopen the command prompt window to reset its system variables.
      7. Run the command **git --version**. You should see the installed git version in the command prompt window.

## Ensure that ffmpeg has been installed and configured for use by whisperX

1. Determine if command line ffmpeg is already installed and accessible from the command line.
   1. Open a command prompt
   2. Run the command **ffmpeg -version**. If it shows ffmpeg‘s version, ffmpeg is installed and configured as required.
2. Otherwise, if this command failed, first ensure that ffmpeg.exe has been installed.
   1. Determine if ffmpeg.exe has been installed.
      1. Open an instance of Windows Explorer.
      2. Switch the visible window to This PC > C:\
      3. Use the search window to search for ffmpeg.exe. If this search succeeds, ffmpeg.exe has been installed.
   2. If ffmpeg.exe was not located, do the following (see <https://phoenixnap.com/kb/ffmpeg-windows>):
      1. Go to the ffmpeg download page, at <https://ffmpeg.org/download.html>
      2. Under “Get Packages & Executable Files”, select the Windows icon. This should expose a link named Windows builds from gyan.dev. This link currently references <https://www.gyan.dev/ffmpeg/builds/>

* Browse to the referenced website.
* In the first section under Git Master Builds, find the link named ffmpeg-git-full.7z.
* Download this file. It will have a name like ffmpeg-2024-03-07-git-97beb63a66-full\_build.7z; the name shows the date on which the distribution was built, along with a check code.
  + 1. In the download directory, extract the top-level directory from this build using a utility that understands the 7z archive format.
* If you don’t have such a utility on your platform, one common utility, 7zip.exe, can be downloaded and installed from <https://www.7-zip.org/download.html>.
  + 1. Rename the extracted directory, which will have the same name as the downloaded archive, as ffmpeg.
    2. Cut ffmpeg and paste it to C:\Program Files\ffmpeg

1. Then ensure that ffmpeg’s home directory is on the current command path,
   1. Determine if ffmpeg’s home directory is on the current command path
      1. Open a command prompt
      2. Run the command **ffmpeg -version**. If it shows ffmpeg‘s version, ffmpeg is installed and configured as required.
   2. Otherwise, if this command failed, do the following:
      1. Open a Windows command prompt. Search for “edit the system environment variables”
      2. You should see a window named “System Properties” that’s open to a tab named “Advanced”. From this tab, select “Environment Variables”
      3. In the lower, “System variables” tab, select “Path”, then “Edit”.
      4. Enter “New” to add an entry to the Path. In the slot that appears, enter “C:\Program Files\ffmpeg\bin”.
      5. Click OK to close the System Variables window, then OK to close the System Properties window.
      6. Close, then reopen the command prompt window to reset its system variables.
      7. Run the command **ffmpeg -version**. You should see the installed ffmpeg version displayed in the command prompt window.

## Create a virtual environment in which to install whisperX. For this exercise, call it “isc”.

1. At the command prompt, enter **cd /d %HOMEDRIVE%%HOMEPATH%** to switch to the home directory.
2. Once in the home directory, enter the command **python -m venv isc**
3. Activate the virtual environment by entering the command **isc\Scripts\activate**

* This environment, when activated, should display the prompt “(isc) C:\Users\*your user name*>”

## Install whisperX in the virtual environment

Issue the command **pip install git+https://github.com/m-bain/whisperx.git**

* You be asked to log into git in order to execute this command. If so, you will need to use your browser to supply your account name and password—or to create an account if you do not yet have one.
* Let this command run until the command line returns. It will take a while to complete.

# Using WhisperX to Transcribe an Audio File

1. Using Windows Explorer, find the audio file to transcribe.
2. After locating the audio file, right-click on this audio file and click “Properties” from the dropdown menu.
3. In the Properties tab,
   1. locate the path to the file after “Location:”
   2. Copy this file path and note the name of the audio.
4. Confirm you have a command window that’s open to your home directory and virtual environment. If not,
   1. Make sure the command window is open or reopen it.
   2. If you’re not in your home directory, use the command **cd /d %HOMEDRIVE%%HOMEPATH%** to switch to the home directory.
   3. Run the command **isc\Scripts\activate** to (re)enter this environment.
5. At the command prompt, run the command **whisperx *<path><filename>* --compute\_type int8** where ***<filename>*** is the name of the file to transcribe and ***<path>*** is the file system path to that file.

* For example, enter **whisperx C:\Users\Phil839\Downloads\audio001.mp3 --compute\_type int8** when
* the file path is “C:\Users\Phil839\Downloads\”
* the file name is “audio001.mp3.”
* Note that an additional backward slash follows the file path then the audio file name.

# Using WhisperX to transcribe and diarize an audio file

## configuring WhisperX for diarization

While some of the steps in the following sequence of steps might not have been essential for configuring whisperX to support diarization, the following process—after four days’ worth of experimentation—worked.

1. Establish an account on huggingface.co
   1. Browse to https://huggingface.co/join
   2. Specify a user name and a password

This account is essential for obtaining access to the whisperX “pipelines” for supporting diarization

1. Agree to the terms and conditions that govern access to the three libraries that whisperX uses for diarization
   1. Browse to <https://huggingface.co/pyannote/segmentation>. Follow directions.
   2. Browse to <https://huggingface.co/pyannote/voice-activity-detection> Follow directions.
   3. Browse to <https://huggingface.co/pyannote/speaker-diarization-3.1>. Follow directions.

Some of these steps will ask for an organization and a website.

1. Generate a token for accessing HuggingFace software from your host platform.
   1. Browse to <https://huggingface.co/settings/tokens>
   2. Follow directions

Also recommended: install support for NVIDIA CUDA on hardware platforms that support GPUs.

* For the NVIDIA CUDA drivers, along with documentation for how to install them, see <https://developer.nvidia.com/cuda-downloads>.
* In order to enable WhisperX’s use of CUDA devices, follow the directions at <https://stackoverflow.com/questions/75775272/cuda-and-openai-whisper-enforcing-gpu-instead-of-cpu-not-working>:
  + Configure torch to use CUDA: i.e., execute

**python -m pip install torch torchvision torchaudio --index-url** [**https://download.pytorch.org/whl/cu118**](https://download.pytorch.org/whl/cu118)

* + Reconfigure WhisperX to use the updated torch: i.e., execute

**python -m pip install -U openai-whisper**

* + Confirm that torch has been configured to use CUDA.
    - Open a python interpreter
    - At the prompt, execute **import torch**
    - After this command completes—it may take some time—execute  **torch.cuda.is\_available()** This last command should return **True.**

While the CUDA argument can be passed as the model’s device parameter, according to this post’s authors, WhisperX will automatically attempt to use a GPU if it detects one on the host platform.

## running WhisperX with diarization

1. Confirm you have a command window that’s open to your home directory and virtual environment. If not,
   1. Make sure the command window is open or reopen it.
   2. If you’re not in your home directory, use the command **cd /d %HOMEDRIVE%%HOMEPATH%** to switch to the home directory.
   3. Run the command **isc\Scripts\activate** to (re)enter this environment.
2. Run the commend  **set HF\_TOKEN=*<token>***, where ***<token>*** is the token you obtained from <https://huggingface.co>.

* Strictly, speaking, I don’t know if this step was required. It was, however, mentioned on a Q&A website.
* You might also consider adding this definition to your system’s environment variables. For more on how to update environment variables, see the previous section’s directions for updating PATH.

1. Log into the HuggingFace portal
   1. Execute the command  **isc\scripts\huggingface-cli.exe login**
   2. In response to the prompt for a password, enter the token you obtained from <https://huggingface.co>

* If you wish, confirm that you logged in by running the command **isc\scripts\huggingface-cli.exe whoami**

1. Run whisperx as before, with the additiional option **–diarize --hf *<key>***, where ***<key>*** is the token you obtained from <https://huggingface.co>.

For more information on whisperX, see https://github.com/m-bain/whisperx .

# Exiting the virtual environment

To exit the virtual environment, issue the command **deactivate**.

# Troubleshooting the installation

## WhisperX fails with the error message “missing the required positional arguments”

This is a problem in WhisperX version 3.1.2. The syndrome is as follows:

TypeError: **new**() missing 3 required positional arguments: 'max\_new\_tokens', 'clip\_timestamps', and 'hallucination\_silence\_threshold'

To fix this error, after installing WhisperX,

* Locate the file **asr.py** in the WhisperX distribution. It should be in the WhisperX package home directory.
* Ensure the code immediately after **"suppress\_numerals": False** (ca. line 322) has the following three lines. If not, add them:

"**max\_new\_tokens": None,**

**"clip\_timestamps": None,**

**"hallucination\_silence\_threshold": None,**

Alternatively, reload **asr.py** from the PiPy repository; the problem has supposedly been finsed.

## Audio file can’t be found, regardless of where it’s positioned

WhisperX’s load\_**audio()** routine (**audio.py**) raises **RuntimeError(‘Failed to load audio’)** without first checking if ffmpeg.exe has been installed. Verify that **ffmpeg.exe** has been properly installed.

## Module ffmpeg has no attribute ‘Error’

If you encounter **AttributeError: module 'ffmpeg' has no attribute 'Error'** try reinstalling ffmpeg, as follows:

**pip3 uninstall ffmpeg**

**pip3 uninstall ffmpeg-python**

**pip3 install ffmpeg-python**

## Value Error

If you encounter **ValueError: Requested float16 compute type...,** use the following command to specify the compute type as int8:

whisperx audio.wav --compute\_type int8

# whisperX – Additional Documentation

A list of whisperX options can be obtained by running **whisperx --help.**  Documentation for the five whisperX models—tiny, base, small, medium, and large—can be obtained from <https://github.com/openai/whisper/blob/main/README.md>

For more details and troubleshooting, refer to the WhisperX documentation at **https://github.com/m-bain/whisperX#setup-%EF%B8%8F**