Installing Logos 8 (using WINE) in Ubuntu / Wasta-Linux

Wine (recursive backronym for Wine Is Not an Emulator) is a free and open-source compatibility layer that aims to allow computer programs (application software and computer games) developed for Microsoft Windows to run on Unix-like operating systems such as Linux. After much hard work Logos for Windows runs on Linux and is very stable. Faithlife have been encouraging of this effort but they do not officially support Logos running in this manner.

New document for Logos 9 instructions here:
Logos 9 WINE Installation in Ubuntu / Wasta-Linux

For **Logos 9 and 64bit** systems **another** recommended method is to use the ferion11 scripts. This process uses a script to download and setup wine with the option to use a bundled "AppImage" that should work in all Linux distributions, and then download and install Logos in it. Alternatively you can install wine 6.3 or newer from winehq.org or your distros repos and use the ferion scripts to install Logos 9 without the AppImage.

Further references / discussion:

- Logos Forum Discussion (link goes to the most recent page as of 26-Nov-2020): https://community.logos.com/forums/t/121.aspx?PageIndex=57
- WineHQ Logos 8 AppDB Entry: https://appdb.winehq.org/objectManager.php?sClass=version&iId=38306
- wasta-logos-setup Github repository (source code and bug reports): https://github.com/wasta-linux/wasta-logos-setup
- Running Logos in Linux Telegram Group: https://t.me/linux_logos

---

Please set your google docs mode to **Viewing** in the top right corner (or go to View -> Mode -> Viewing). We are getting a lot of people accidentally typing into this doc instead of their terminal ;-)

Update - These instructions are known to work with: (please add your config once tested for the benefit of others)

32bit wine is considered stable with known bugs.

- Ubuntu 20.04 / Wasta-Linux 20.04 with 32bit wasta-wine
- Kubuntu 20.10 64bit OS with 32bit wasta-wine (AMD)
- Kubuntu 20.04 64bit OS with 32bit wasta-wine with NVIDIA
- Kubuntu 20.04 64bit OS with 32bit wasta-logos-setup pkg (using instructions below)
- KDE Neon 5.20 with 32bit wasta-wine with NVIDIA
- Linux Mint 20 Cinnamon 64bit OS with 32bit wasta-wine (NVIDIA + INTEL)

- Linux Mint 20 XFCE OS with 32bit wasta-wine
- Xubuntu 20.04 with 32bit wasta-wine
- Kubuntu 19.04 64bit OS with 32bit Wine
- Kubuntu 19.10 64bit OS with 32bit Wine
- Kubuntu 18.04 64bit OS with 32bit wasta-wine with NVIDIA
- Ubuntu 19.10 64bit OS (following instructions)
- Ubuntu (Wasta-Linux) 18.04 64bit OS with 32bit wasta-wine
- Arch Linux 64bit OS with 64bit Wine
- Manjaro KDE 18.1.2 (with the package debtap from the AUR) 64bit OS / 32bit Wine See guide
- Linux Mint 19.2 64bit OS with 64bit Wine (Linux Kernel 5.0.0-36) following directions below as written.
- Chrome OS - previous guide was removed :-(
- Debian Buster

Step 1: Add Necessary Software Repository

- Add the Wasta Wine PPA - available for Ubuntu 18.04 ‘bionic’, Ubuntu 20.04 ‘focal’, or Ubuntu 20.10 ‘groovy’ users. This software repository includes the **wasta-logos-setup** package and additional packages for the patched version of wine (wasta-wine), winetricks and winetricks cache (wasta-winetricks, wasta-winetricks-dotnet40, wasta-winetricks-dotnet48, wasta-winetricks-corefonts), and the Logos 8 installer (wasta-logos-installer):

```
sudo add-apt-repository ppa:wasta-linux/wasta-wine
```

- Update software lists (should be automatically done for 18.04+ users as part of the add-apt-repository command, but doesn’t hurt to make sure):

```
sudo apt update
```

Step 2: Install Wasta-Logos-Setup package

```
sudo apt install wasta-logos-setup
```

- NOTE: If you are in Linux Mint, you will need to install the packages wasta-wine, wasta-winetricks, wasta-winetricks-dotnet48, wasta-winetricks-dotnet40, wasta-winetricks-corefonts, and wasta-logos-installer. For KDE Neon, you will need to enable the i386 architecture (disabled by default in Neon) by running `dpkg --add-architecture i386 && apt update && apt install wine32` before running `apt install wasta-wine wasta-logos-setup`. For Ubuntu and other derivatives they will be automatically installed when you install wasta-logos-setup. Regardless, you can always run this fuller command to ensure all the bits get installed:

```
sudo apt install wasta-logos-setup wasta-wine wasta-winetricks
wasta-winetricks-dotnet40 wasta-winetricks-dotnet48 wasta-winetricks-corefonts
wasta-logos-installer
```

Step 3: Run “wasta [Logos] Setup” from the Main Menu

- NOTE: **wasta-logos-setup** saves any existing Logos Data to a separate directory and links it to the Logos installation so that if you re-install Logos the data will not be lost. For reference, the directory used is:

  ```
  $HOME/.logos-data/
  ```

- NOTE: If you need to symlink to a different location because your home directory is not large enough for a Logos install, then please make a **~/.wine-logos** symlink to your desired location *before starting wasta-logos-setup.*

- NOTE: **wasta-logos-setup** will look in **~/Downloads** for **Logos-x86.msi** first before using the **wasta-logos-installer** version of **Logos-x86.msi** in case you want to use a different / older version of the installer. As of 31-Oct-20 the version of the **wasta-logos-installer** is 8.17.0.0014

- NOTE: The install process takes approximately 10 - 12 minutes on a 6th gen Intel i5 mobile processor.

- NOTE: If there are any issues, please inspect the log files created in this folder:

  ```
  $HOME/.config/wasta-logos-setup/
  ```

- NOTE: please report if you are having the process “hang” or “get stuck” after the installation of .NET 4.8. I have added a NOTE to the dialog window to direct users to open “System Monitor”, search for “mscorsvw.exe” and manually “End” those processes if it seems to have “hung”. In practice it seems that the .NET 4.8 install has completed successfully and the Logos install will work correctly, but for some reason *sometimes* this .NET 4.8 process does not cleanly exit.

Step 4: Run “Logos Bible Software” from the Main Menu

- NOTE: Alternatively, you can start Logos from the terminal so you can see any output:

  ```
  WINEPREFIX=$HOME/.wine-logos/ /opt/wasta-wine/bin/wine
  $HOME/.wine-logos/drive_c/users/$USER/Local\ Settings/Application\
  Data/Logos/Logos.exe
  ```

- NOTE: Historically there were problems with getting the Logos Indexer to complete. If the indexer keeps crashing, you can remove the contents of these 4 folders and then re-start the Logos Indexer:

```
rm $HOME/.wine-logos/drive_c/users/$USER/Local\ Settings/Application\
Data/Logos/Data/*/BibleIndex/*

rm $HOME/.wine-logos/drive_c/users/$USER/Local\ Settings/Application\
Data/Logos/Data/*/LibraryIndex/*

rm $HOME/.wine-logos/drive_c/users/$USER/Local\ Settings/Application\
Data/Logos/Data/*/PersonalBookIndex/*

rm $HOME/.wine-logos/drive_c/users/$USER/Local\ Settings/Application\
Data/Logos/Data/*/LibraryCatalog/*
```

- NOTE: You can manually start the Logos Indexer with this command:

```
WINEPREFIX=$HOME/.wine-logos/ /opt/wasta-wine/bin/wine
$HOME/.wine-logos/drive_c/users/$USER/Local\ Settings/Application\
Data/Logos/System/LogosIndexer.exe
```

Congratulations: You now can run Logos 8 through WINE in Ubuntu

Now see below for some bugfixes and please join the effort to get us to 100% functionality. (we are very close)

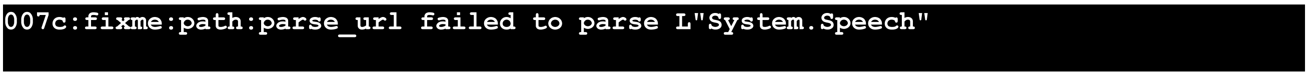
Unresolved Bugs - So that we can work on them.

1. Major (Urgent) - Audio and Video fail to play. (Workaround - Use webapp for these features) - Bradley from faithlife tells us Videos use the Enhanced Video Renderer (EVR) of Microsoft Media Foundation n.b. Winetricks can install mf but that results in a crash when opening a resource that uses videos based on my tests so far. MF has been the subject of a lot of dev work in wine recently so it might work out of the box soon. In the meantime some apps need custom config beyond what winetricks offers for media playback. See here https://bugs.winehq.org/show_bug.cgi?id=47905 - Nikolay Sivov from codeweavers is working on this as it is needed for a number of apps. This is likely to arrive before wine 6.

2. Minor (Urgent - Few probably use this feature but JG uses it a lot for accessibility) - Read aloud fails. (listed separately as it probably depends on a speech subsystem rather than media function. I think we will need a native dll here… will do some research. Trying https://www.reddit.com/r/linux/comments/6z41qb/my_text_to_speech_tts_solution_wine_microsoft/

   ```
   winetricks -q speechsdk
   ```

   This error appears in the terminal when choosing to “Read Aloud”:

   ```
   007c:fixme:path:parse_url failed to parse L"System.Speech"
   ```

   Current theory is that no default voice is set for the system. I have System.Speech working using a terminal app. Just need to get Logos to use it. Found https://www.webbie.org.uk/changingNarratorToUseADifferentSAPI5Voice.htm want to test these reg keys… does show where to choose voice but does not fix.

3. Minor - Logos.exe fails to fully end process on exit. Workaround is to manually end the process with a task manager. Rare bug, not serious but defo happens still. 2019-11-25

4. Minor - Printing - rtf export worked but printing to a printer did not. Also all page size options missing. Logos seems to use gdiplus for print… needs exploring

5. Minor - Shortcut keys for highlighting don’t work. Many other shortcut keys do seem to work but for example shift+y assigned to yellow highlighting does nothing. - For now the new highlighting pop up works so the shortcut is probably redundant anyway.

Resolved Bugs

1. **Fixed -** Graphical glitching - something you were looking at previously comes back on screen. After a moment it looks right again. E.g. go to home screen then go back to books. After a moment you are looking at the home screen again. Then it goes to your books and stays there ok. Guess is this has to do with buffering. https://bugs.winehq.org/show_bug.cgi?id=47837 Use winetricks and set the option to SetDirectDrawRender to GDI - DDR=GDI possibly needs to be set with a regkey some versions of winetricks don’t do it.

2. **Fixed -** Very many lines of parse_url fail this is due to bug no. 47277 and a workaround is to add ‘LC_ALL=C’ after the command to launch the app. No longer necessary. See https://bugs.winehq.org/show_bug.cgi?id=47277

3. **Fixed -** previous fix of “LC_ALL=C” was preventing layouts from being able to load. Minor - Always loads to a blank layout… setting a default layout, home or most recent has no effect.

All bugs submitted to bugzilla should ideally be added to the appdb to keep them tied together and tracked. See: https://appdb.winehq.org/objectManager.php?sClass=version&iId=38306

Historical Notes on getting Logos running in Wine