



AV Linux MX-21 Edition

User Manual

Prepared by: Glen MacArthur

DISCLAIMER (PLEASE READ) :

Debian/GNU Linux comes with no guarantees so consequentially neither does AVL-MXE. I accept no responsibility for any hardware/software malfunctions or data loss resulting from its use. It is important to note that the AVL-MXE ISO may contain software that is non-free and may be distributed under special licensing arrangements with the original developers, re-distributing the AVL-MXE ISO with the non-free content included is a violation of these licenses. AVL-MXE may potentially contain Multimedia Codecs that may be under patent in certain countries, it is the Users responsibility to know the law as it applies to their own respective countries before downloading or installing AVL-MXE.

Bookmarks

- [About This Manual](#)
- [Getting Help](#)
- [A New Chapter for AV Linux!](#)
- [AVL-MXE Features at a Glance](#)
- [Included Trusted Debian Repositories](#)
- [External/Independent Software in AVL-MXE](#)
- [Specific AVL-MXE Tools and Packages](#)
- [Known Issues in AVL-MXE](#)
- [Downloading the AVL-MXE ISO File](#)
- [Running AVL-MXE as a 'LiveISO'](#)
- [The Network Assistant for WiFi](#)
- [Installing AVL-MXE](#)
- [Installation Suggestions](#)
- [Optional Auto Mounting of Internal Drives](#)
- [The AVL-MXE Assistant](#)
- [The Kernel Conundrum](#)
- [Xfce4 with Openbox](#)
- [Slim Login Manager](#)
- [Getting Around in Xfce4](#)
- [Thunar File Manager](#)
- [Qt5 Configuration Tool](#)
- [MX-Snapshot in AVL-MXE](#)
- [Software Installation Notes](#)
- [Audio and MIDI in AVL-MXE](#)
- [Initial Setup of Ardour and Mixbus32C](#)
- [Running Windows Audio Software](#)
- [Saving and Restoring JACK Connections](#)
- [Commercial Software Demos in AVL-MXE](#)
- [Thanks and Acknowledgements](#)

About This Manual

This User Manual is a Work-In-Progress and will be for some time I'm sure. I will be adding new material as need arises and as I get better acquainted with MX Linux. In its current state most of the information has been screened for accuracy but there are many screenshots that are yet to be updated and will be in future Manual updates. A huge amount of time and effort goes in Documentation for any project and it is not in any way... 'fun'. This is why many Open-Source projects do not provide adequate (or any) Docs. Creating, sharing and maintaining AV Linux are all done 100% in my spare time.. I do this because I love to do it and because I want to do it... I don't need Kudos but I do need understanding that I can only provide so much in the time available. Getting as far as Documentation is stretches the time envelope even further. "Well isn't that great!! How do I get some help around here!!??" you may well ask..

Getting Help

Even though AVL-MXE doesn't have a full time dedicated User Forum of it's own there are plenty of ways to help yourself.. Number one is what you're reading right now! On top of that MX Linux is one of the best documented Linux projects I've ever seen. As clarified in the following section MX Linux is the 'wheels and chassis' of AVL-MXE and it has both terrific Docs and a large community on its even more terrific User Forum. AVL-MXE is new project and is still 'arms-length' from MX Linux so it is advised if you are a User Forum type of person for now post questions in the '**Respin**' section of the MX Forum. Make sure to mention 'AVL-MXE' in your question so people reading know what you are using. I also frequent the Linux Musicians Forum as much as I can and you can post questions pertaining to AVL-MXE there in the **Linux Distributions & Other Software** Forum section as well.

On the flipside of 'Getting Help' is receiving help and the number one need for AVL-MXE is for people already using it to assist others in using it! It's great to get some praise, awesome to get some positive press, and I always appreciate donations but *I need help the most* in supporting and assisting users of AVL-MXE! If you are finding AVL-MXE to your liking and getting things done with it and are an experienced Linux Content Creator PLEASE consider helping me to provide User support for AVL-MXE if you can spare some time and effort.

A New Chapter for AV Linux!

Since 2008 AV Linux has been an independently developed Debian-based OS to facilitate Multimedia Content Creation on the Linux platform. AV Linux has relied on 3rd-party ISO creation tools and Installers starting with 'Remastersys' and in recent years 'Systemback' and although these were both excellent Utilities they have both unfortunately ceased active development and in Systemback's defense it was intended as a backup application not really a Distribution tool. In recent years AV Linux has struggled with getting an increasingly popular production system that has matured over many years to install reliably on newer computers with UEFI booting firmware and NVMe SSD.2 drives and it was high time for a change.... Enter MX Linux!

My first experience with Linux was in 2006 with 'Mepis Linux' and it was a great introduction to the Desktop Linux world and now many years later after the Mepis project waned and evolved into MX Linux it remains a great Linux Distribution backed and supported by a large and enthusiastic community of developers and users and bolstered with a custom 'MX' toolset that is without peer. Among these tools are great utilities to Snapshot and create ISO Images and a modern simple-to-use Installer; both of which are what AV Linux was in dire need of.

After several months of development work here is a new hybrid with a new name "AV Linux MX Edition" (a.k.a AVL-MXE)... Make no mistake the wheels and chassis are MX Linux and the customizations and Multimedia focus are AV Linux. Both MX and AVL are established projects recognized by the Linux Community and the naming scheme is to keep the AV Linux branding present and of course pay all due respect to the vital importance of MX Linux as the base Distribution. *It should also be made abundantly clear at this point AV Linux MX Edition is an independent and unsupported 'respin' of MX Linux and is not under any official umbrella of the MX and antiX projects.* It is my sincere hope that as time goes on both projects will blend together and share as many resources as possible but for now there is much work to be done and sufficient time needs to pass to assess what the potential is for a closer partnership.

AV Linux MX Edition is much more than MX with different theming and wallpaper, both projects have the XFCE4 Desktop Environment in common and both projects enjoy all the benefits of being based on Debian/GNU Linux. At first glance AVL-MXE looks like many other XFCE4-based 'Desktop Linux' Distributions and indeed can easily be used a 'daily driver' for basic everyday computing tasks but where things differ is the Multimedia Content Creation side of AVL-MXE. The philosophy of AVL-MXE has always been "Install and Create" and the Audio and Video applications are set up and organized with a lot of thought toward having the system pre-configured and prepared for Content Creators who need a high-performance, solid, reliable Workstation OS to tackle projects both large and small. Great attention is paid to having mature versions of the DAW (Digital Audio Workstation) software and well-tested and useful Instrument and Effect 'Plugins' to maximize potential. The included 'Ardour' and the Demo version of the related 'Harrison Mixbus 32C' DAW provide Audio production capabilities from the Bedroom Producer to a full professional Commercial Studio Environments. Uniquely AVL-MXE also provides a LiveISO platform for the User to evaluate both excellent Open Source software as well as a growing list of Commercial Software demos. In the spirit of free choice AVL-MXE has a one-button option to remove all Demoware from the System for those who prefer an OS without Commercial options. The main focus of AVL-MXE is Audio Production but Video Content Creators and Graphics Designers will find a rich production environment as well with 'Avidemux', 'Cinelerra-GG', Kdenlive, 'OBS-Studio', 'Blender', 'GIMP', and 'Inkscape' included as well.

AVL-MXE Features at a Glance

- Based on MX Linux (powered by Debian/GNU Linux).
- Includes the phenomenal 'MX Tools' Utilities.
- Custom-Themed Hybrid XFCE4/Openbox Desktop Environment.
- Extensive Audio/Video and Administrator-friendly Thunar Custom Actions.
- Liquorix performance-tuned Kernel.
- Tuned for best potential low-latency Audio performance (threaded IRQ's, rtirq-init)
- Seamless ALSA, JACK and PulseAudio integration through 'Qjackctl' and 'pajackconnect'.
- CPU Governor Set to 'Performance' by default.
- 'AVL-MXE Assistant' Utility to simplify optimal System setup.
- Fully supported and functional 'bundled' version of Ardour DAW.
- Commercial Demo of Reaper DAW.
- Commercial Demo of Harrison Mixbus32C.
- Commercial Demos of selected Audio Plugins.
- 'One-Click' removal of all Demo-ware for those who would rather not have Commercial options.
- Included Custom Packaging of Plugins not available in any other Repositories.

*There may occasionally be 'limited releases' of a customized 32bit version of AVL-MXE.. As time goes on this will become less and less likely and possible. The 32bit x86 CPU architecture is now deprecated and many software applications are no longer developed or tested for 32bit so this release is limited to special cases and older 32bit PC's. AVL-MXE 32bit will definitely lack programs and features that are on the 64bit system. For this reason the 32bit version comes 'as-is' with absolutely NO guarantee to see consistent development and the security and updates are completely at the discretion of the packagers maintaining the included repositories. User Beware!

External/Independent Software in AVL-MXE

In an ordinary Linux system it is recommended policy to only install Software Packages from a trusted Repository system and there are many valid security reasons to do so. AVL-MXE is a different case and it is focused on specific tasks which sometimes require software outside of Repositories to provide the best User experience. I cannot stress enough that any outside Packages that are included in AVL-MXE are provided by trustworthy sources and developers that have a proven record of providing safe and mature Applications. The reasons that these Applications are not included in either the Debian or MX Repositories are many... Some are encumbered by licenses that may not make them easy to integrate in Debian or MX, some may be too niche to have been noticed or included yet and some may be Closed-Source Software. In time some of these Packages may find their way into Repositories but until then they are being identified here both to inform Users of their presence and also to provide their web locations in case AVL-MXE Users want to check for upgrades between AVL-MXE ISO releases. *PLEASE NOTE: [Commercial Demos of Software are explained in their own Chapter at the end of this Manual.](#)

Ardour Digital Audio Workstation – This is the top featured Open Source DAW included in AVL-MXE and has been in development for 20+ years. Outdated versions can be found in Repositories but it is very important to note that Ardour provides its own self-contained ‘bundles’ built with their own custom patched versions of supporting libraries and its developers ONLY support these bundles. In addition Ardour is Open Source but it is payware so Users are expected to either purchase the Ardour binary bundles or optionally build their own versions from Source Code. AVL-MXE provides a complementary fully paid and functional up to date version of Ardour on the ISO but upgrades will need to be purchased at the User’s discretion.

Yabridge – A very important tool to facilitate using Windows VST Audio Plugins in Linux DAWs in conjunction with Wine-Staging. It isn’t found in any Debian repositories but its developer distributes binaries from the project’s Github Repository. The Yabridge binaries are included by default on the ISO and are located in \$HOME/.local/share/yabridge

Odin2 Synthesizer - This phenomenal cross-platform LV2 and VST3 Synthesizer Virtual Instrument is independently distributed by its own developer on its own Website.

Sitala Drum Sampler – This standalone and VST drum sampler is distributed from its developers Website.

Sfizz SFZ Tools - A Plugin for hosting SFZ Sound Libraries, the Package on AVL-MXE is sourced for the SFZ Tools Website.

Surge Synthesizer – Surge is a popular Open-source cross-platform Synthesizer. It is presented on AVL-MXE with the Debian Package from its github project page.

Helm Synthesizer - Matt Tytel’s Helm Synthesizer is Provided in a Debian Package from the Helm Website.

Specific AVL-MXE Tools and Software

AVL-MXE contains several specialty Packages that are included as part of the ISO, since my work on AVL-MXE is done in my spare time and I often do not work on it for days or weeks at a time these packages are not contained in a Repository and at the current time are included and updated as new and maintenance releases of the AVL-MXE ISOs are made available.

These Packages should be considered vital parts of the Operating System and *not* be removed:

- avl-mxe-assistant
- avl-mxe-user-manual
- avl-mxe-thunar-custom-action-scripts
- avl-mxe-openbox-extras
- avl-mxe-grub-themes
- avl-mxe-splash
- avl-mxe-diehard-theme

These Packages can be considered optional and can be removed at the User's discretion:

- avl-mxe-extra-fonts
- avl-mxe-extra-plugins-dist

AV Linux MX Edition Packages FTP:

As noted earlier because AVL-MXE is a spare-time project and my available time is not consistent there is not currently a proper APT Repository being maintained for the external customized AV Linux Packages. In the event that you mistakenly uninstall or remove one of the AVL-MXE Custom Packages and wish to reinstall it there is a Web FTP Folder where these particular Packages are stored. **PLEASE NOTE: These are the Package versions that will have been installed on the released ISO of AVL-MXE, this FTP is not intended to be maintained for updates, it is for replacement purposes only however the Packages will be updated to match new AVL-MXE ISOs as they are released.*

The Packages FTP for AV Linux MX-21 Edition can be accessed here:

http://www.bandshed.net/packages/AV_LINUX_MX-21/

***New! - AppImages Contained in AVL-MXE**

An increasing number of software projects are opting to distribute the latest versions of their software through self-contained bundled methods, two of the most popular formats are 'AppImage' and 'Flatpak'. There are some specialized cases where these formats make sense:

- The application is only available in AppImage format and is not Published or Packaged in a Debian Repository yet, or the latest source code is not readily available or easy to compile.
- The application uses a completely different graphical or Desktop Environment than the host system uses, as an example using a single GTK-based App on a KDE System or vice-versa.
- The application is encumbered by licensing or other restrictions that prevent it from being Distributable in more traditional Linux Packaging formats from a Repository.
- You want the option to work with more than one version of a particular application, perhaps an older version is extremely stable and a newer version has less stability but new features.

For the time being AVL-MXE provides three external applications in the AppImage format: *Cinelerra-GG Video Editor, Kdenlive Video Editor and MuseScore Sheet Music Scoring application*. The reason for this is that all of these applications for one reason or another meet the criteria above, this is a new feature I'm trying out to get more recent application versions available for use on the Live -boot AVL-MXE ISO and subsequent installs.

Using Kdenlive Video Editor as an example in order to get the AppImage included in the AVL-MXE ISO build system I have formatted the AppImage applications into Debian 'Deb' Packages. Kdenlive will appear in the Debian Packaging Utilities (ie Synaptic, MXPI) as 'kdenlive-appimage' as will any and all other Packages that are derived from AppImages. *This is only to integrate them into the ISO and none of the included AppImage applications will be upgraded through regular MX Linux repository updates.* In the event that you want to update one of these AppImages you will want to use 'Synaptic Package Manager' to uninstall the existing Debian-Packaged version that came with AVL-MXE (ie 'kdenlive-appimage') and then download the newer updated AppImage from the Kdenlive or other relevant website and then use it as you would any other externally downloaded AppImage.

Known Issues in AVL-MXE

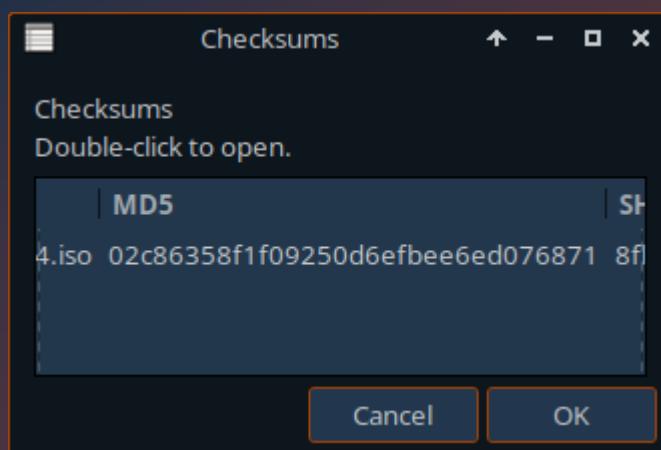
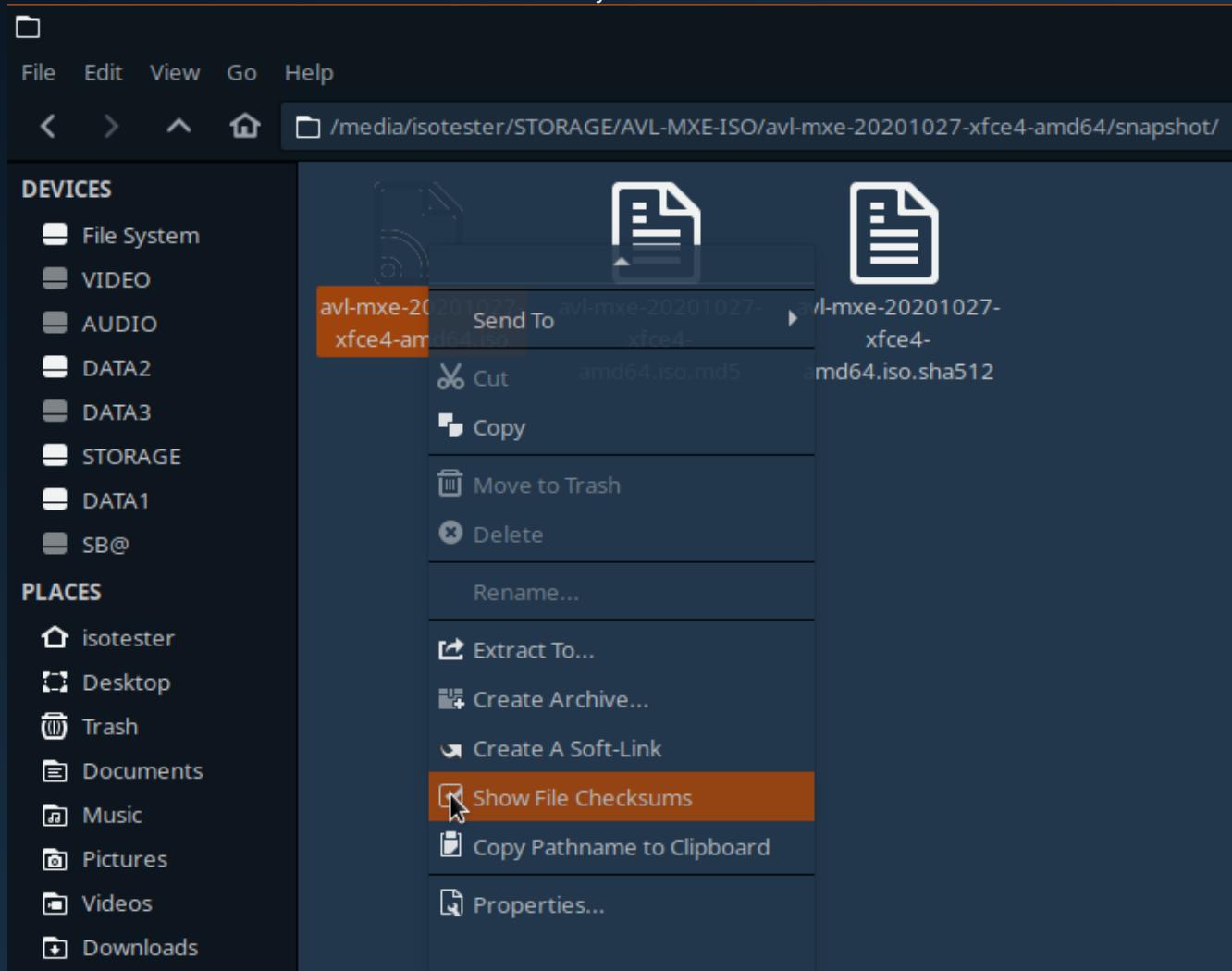
These are still early days in the migration of AVL-MXE to MX Linux and there is still much work to do to determine what features may overlap or even conflict with each other. It is hoped that each release will receive some useful feedback in order to streamline and improve future full releases. Before downloading or installing AVL-MXE the User should be aware of the following known issues:

- ➔ The ISO is about 3Gb in size, this is not BLOAT! Multimedia Apps and Plugins take up a lot of space!
- ➔ There is not a Repository of Custom AVL-MXE Packages (yet).
- ➔ AVL-MXE is currently using the QT5 Config tool 'qt5ct' to apply GTK themes to QT5 Apps.
- ➔ There may still be functionality overlaps between the MX Tools and the AVL-MXE Assistant.
- ➔ AVL-MXE Assistant uses both 'Mousepad' and 'Featherpad' for text file editing.
- ➔ VST3 Audio plugin support for Ardour is in VERY early development!
- ➔ Virtualized environments (ie Virtualbox, VMWare) are NOT suggested for AVL-MXE.
- ➔ By default MX Linux now installs using 'sudo' so the User Password is default for authentication,
*This can be changed to Root Password authentication in MX Tools -> 'MX Tweak'

Downloading the AVL-MXE ISO File

It is VERY strongly recommended to only download AVL-MXE from the mirrors on the [**AVL-MXE Website**](#). This will ensure that you are getting the intended version of the ISO files and not something that has been maliciously tampered with. While AVL-MXE is in early development there may be a limited number of mirrors but this will likely expand by the time AVL-MXE is through the Release Candidate process. When you download the AVL-MXE ISO you will see that in the mirror download folder is the actual 'AVL-MXE-<year-month-day-DE-arch>.iso' file containing the operating system and two accompanying '.md5' and 'sha512' text files. You should download the both the 'iso' and both text files, you should be able to right-click and use 'Save target as' for the text files. Before running AVL-MXE Live or installing it on your computer you should verify that the 'md5sum' and 'sha512sum' of your ISO download matches the text files on the download server. This will ensure that you have a complete and proper download of the ISO as well as confirming that you haven't downloaded a compromised ISO.

To verify that the md5 and sha512 files match with your downloaded ISO you can simply right-click on the ISO file you downloaded and select the Custom Action to "Show File Checksums". This will open a new window that will take several seconds to calculate the MD5, SHA256 and SHA512 checksums and then show you the calculated values which will appear as long alphanumerical text strings. If these checksum text strings match the strings in the text files which accompanied the ISO download then you can be assured you have both a complete download and that the ISO file is legit and safe to use.

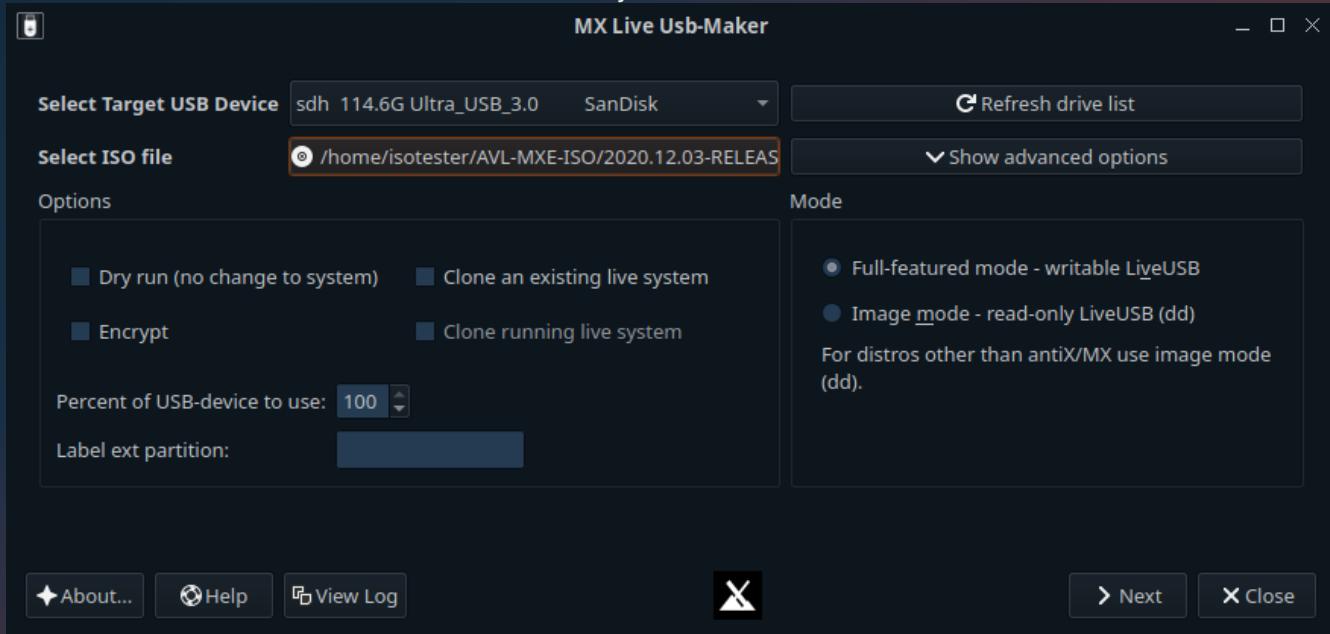
Checksum verification in Thunar:

Running AVL-MXE as a 'LiveISO'

With USB Keys so abundantly plentiful and cheap I strongly recommend only running AVL-MXE as a 'LiveISO' from a USB Key. Running LiveISO's from DVD Media should be considered a deprecated and obsolete method to effectively try a Live Linux Operating System these days. AVL-MXE can be run on USB Media as small as 4Gb and for the ultimate LiveISO experience with persistence (the ability to save changes made while running the Live session) I highly recommend using the 'MX Live USB Maker' Utility to create an AVL-MXE LiveISO. If you are not already using MX Linux then there are other options such as 'Unetbootin' although it will not create a persistent bootable USB key it will create a key which will enable you to try out and potentially install AVL-MXE. Diehard and experienced Linux command line gurus can also use the well-known 'dd' command to manually copy the AVL-MXE ISO file to a USB key. There are a number of other Utilities within Linux and on Windows to create bootable USB Keys but they are beyond the scope of this manual.

Once you have created a bootable USB Key you will need to plug it into to a USB port on your computer and use the Computers BIOS or UEFI firmware Utility to select the USB Key at boot time. A common hotkey is the 'F12' key to enter a boot device selection screen but every BIOS and UEFI Utility are a bit different so if F12 isn't the ticket consult your Computer's documentation or give 'er a Google! By default AVL-MXE should not require a login password to boot into the Desktop, if the User needs to logout then the Username and Password to log back in is 'demo' and for Root access the Password is 'root'. If you need help while running a LiveISO session then both the MX Linux and AVL-MXE PDF Manual files are right in the Openbox Menu!

Screenshot of MX Live USB Maker:



The Network Assistant for WiFi

Due to the staggering amount of computer hardware supported by the Linux kernel and the huge numbers of various networking chipsets that computers are equipped with it is becoming more and more difficult to guarantee that WiFi will work out of the box when you boot a Linux LiveISO. Some WiFi chipsets require some extra config or loading of specific kernel modules and even 3rd party installed firmware to function properly. To mitigate some of this and assist Users in getting WiFi working the MX-Tools Network Assistant is a valuable tool. Of course I can't provide hard and fast examples that will work for all hardware situations so this section will discuss the Network Assistant in generalities.

If you are running a LiveISO session you will find the Network Assistant in the 'MX-Tools' section either from the XFCE4-Panel or in the Whisker Menu, because it requires Root permissions you will need to enter the Password 'root' to run it. The Network Assistant's first Tab will show you which chipsets are found on your computer, depending on what kind of computer you have usually this will be one device for hardwired ethernet and a second WiFi device.. Of primary interest is Tab 2 or the 'Linux Drivers' Tab because it will show what drivers for your hardware are on the system, whether these drivers are loaded or not or if the drivers are blacklisted by some special module config files.

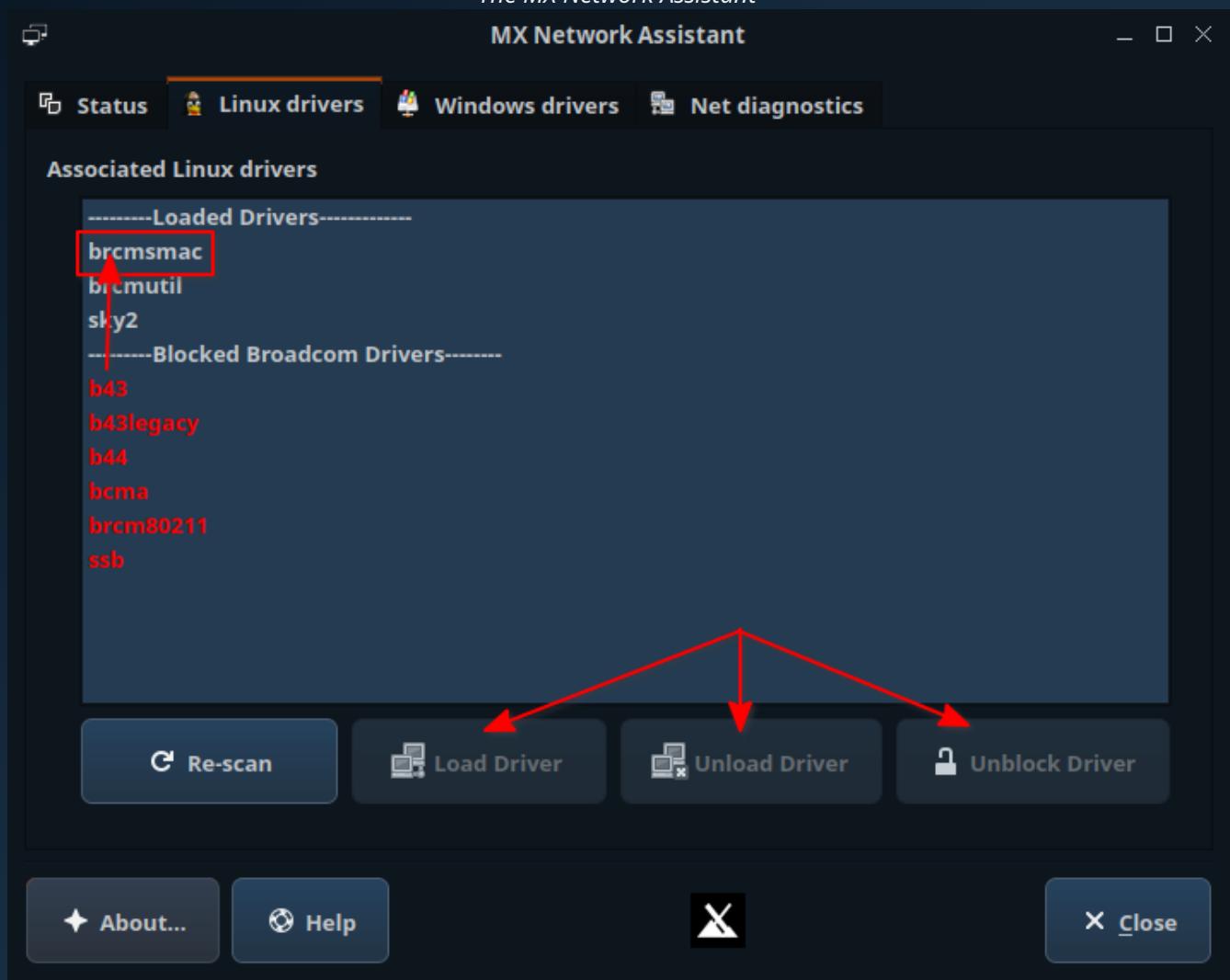
On the next page is a fairly common Broadcom example.. On this particular computer when I boot a default ISO of AVL-MXE the WiFi does not work because the 'brcmsmac' driver is blacklisted by default. In order to fix this I needed to use the 'Unblock' button which allowed the brcmsmac driver to load and enabled my WiFi to work. Usually if you don't have WiFi your Wired ethernet will work so it may take a bit of Google sleuthing from your Wired connection to determine what exact driver is required for your chipset.

A handy Terminal command you can use to determine what Networking hardware is on your computer and which drivers are loaded is this:

inxi -n

inxi -n running in a Terminal

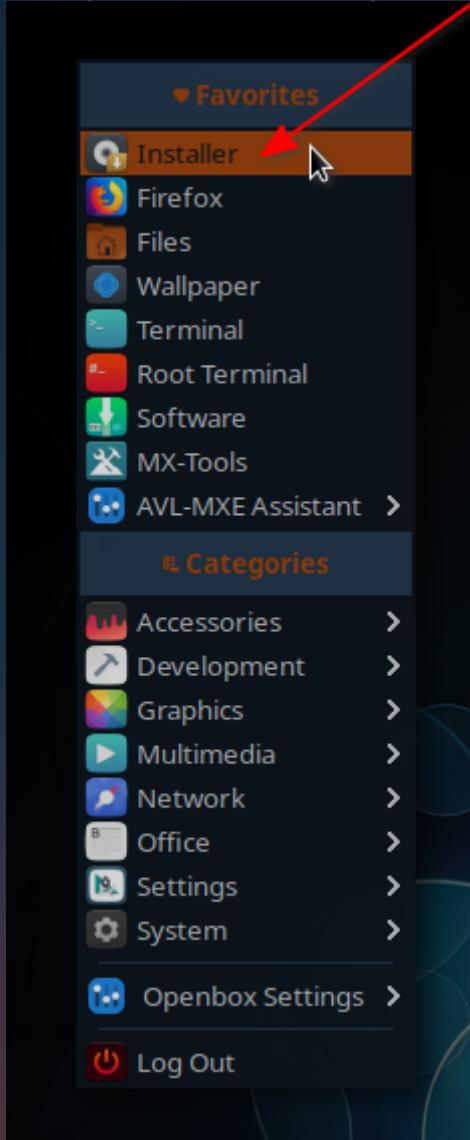
```
$ inxi -n
Network:  Device-1: Intel I211 Gigabit Network driver: igb
          IF: eth0 state: up speed: 1000 Mbps duplex: full mac: b4:2e:99:a9:81:1c
          Device-2: Realtek RTL8812AE 802.11ac PCIe Wireless Network Adapter driver: rtl8821ae
          IF: wlan0 state: up mac: d0:37:45:72:88:84
```



Installing AVL-MXE

AVL-MXE can be installed from a running LiveISO session to the computer hosting it from the Install button the initial 'splash' welcome screen or by right-clicking, opening the Openbox Menu and selecting the 'Installer' entry at the top of the Menu which will launch the MX Installer. The Installer gives step by step Dialogs and the Installer window displays contextual Help text throughout the Installation process to answer any questions the User may have about the various installation options.

Screenshot of MX Linux Installer in Openbox Menu:



Installation Suggestions

Depending on your experience with using Linux you may have some pre-conceived ideas about how a Linux system should be installed and what the HDD partition layout of your installation should be. Many people like to have a separate root and home partition, some like to go with full-blown LVM (Linux Volume Management) setup. Depending on what you are using your installed Operating System for there are certainly valid use-cases for every kind of setup. Because of the intended purpose of AVL-MXE I'm going to suggest some things that may seem out of the ordinary but many years of experience have shown me that there is some merit and solid logic in setting up a Workstation system a bit differently.

When you are using an Operating System as a Multimedia Workstation you have to take into consideration that you will be working with a lot of large files, you will most likely be working on projects for days or even months and since you are creating Audio and Video works that are unique to you and your talents your data will be very precious and not easily replaceable so the safety of important Data is always a concern but the safety of artistic creations are that much more of a concern. Another important consideration is being able to quickly restore and re-install a very complex and specialized Operating System in the case that there is either a Hardware or Software System failure.

Because of these and other considerations I strongly suggest to Install AVL-MXE on it's own system root ('/') partition with the home folder contained within the same root partition. I suggest all important Data and Projects should be kept on other separate Data partitions and not in the User's home folder. In a setup like this the system partition that AVL-MXE is installed to does not have to be overly large but I do suggest a minimum of 25 Gb. If you have a large HDD to work with then focus on allocating the largest amounts of space to creating other partitions for your Data. You may wonder and ask "why can't I just keep all my Data in my home folder like I usually would"? The answer is because in the event that you ever need to either reinstall AVL-MXE (or install a whole different OS) then you want to ensure your precious Data is on a separate partition that is not targeted by an installer and does not need to move or be backed up elsewhere, this way you can install an Operating System on one single dedicated partition without disturbing your Data. In this scenario your home folder should be seen as a temporary staging area to download to, organize, and use for short term purposes while your important Data is stored elsewhere. In the event you have a sudden major system failure you can rest assured that you won't lose everything and will be back up and running in short order.

Even with these precautions *I cannot overstate the extreme importance of backing up your Data to an external source even when it is on separate partitions in the System.* Keeping Data on separate partitions gets us some of the way toward preserving and protecting it but external Backup copies are the only way to guarantee its safety.

Optional Auto Mounting of Internal Drives

There are three methods of enabling Auto Mounting of additional Internal Hard Drives and Partitions on your system after installing AVL-MXE, they are listed here in no particular order of preference but it is recommended to choose and only use one of these methods.

1. The 'mountalldrives' script:

There is a hidden script in AVL-MXE that can be used to automatically mount Drives at boot. This script will work with ext2,3,4, ntfs, and FAT32 formatted Drives and Partitions connected with IDE or SATA. *It has NOT been tested and is not known to work with Drives and Partitions on NVMe and SSD.2 Media!*

To enable this script requires a small edit of the '/etc/rc.local' file, if you are not familiar with editing system configuration files in the Root Filesystem it is strongly recommended to not attempt to enable this feature and move on to one of the other more graphical methods below. You've been warned! For those who have some experience and knowledge of what they are doing you can use 'Menu -> Settings -> AVL-MXE System Editor' and select the '/etc/rc.local' file at the bottom of the list to Edit and then enter your Administrator Password which will open the 'FeatherPad' Text Editor. The rc.local file will open and Lines 13, 14, 15 will look like this:

```
# AUTOMATIC MOUNTING OF ALL DRIVES
# COMMENT LINE BELOW TO DISABLE
# /usr/bin/mountalldrives &
```

To enable the Automatic Mounting feature requires that you remove the comment hash ('#') from the beginning of Line 15 so that it looks like this:

```
/usr/bin/mountalldrives &
```

Save your changes in FeatherPad and on next boot your external Drives and Partitions should appear as mounted in the Thunar File Manager side pane.

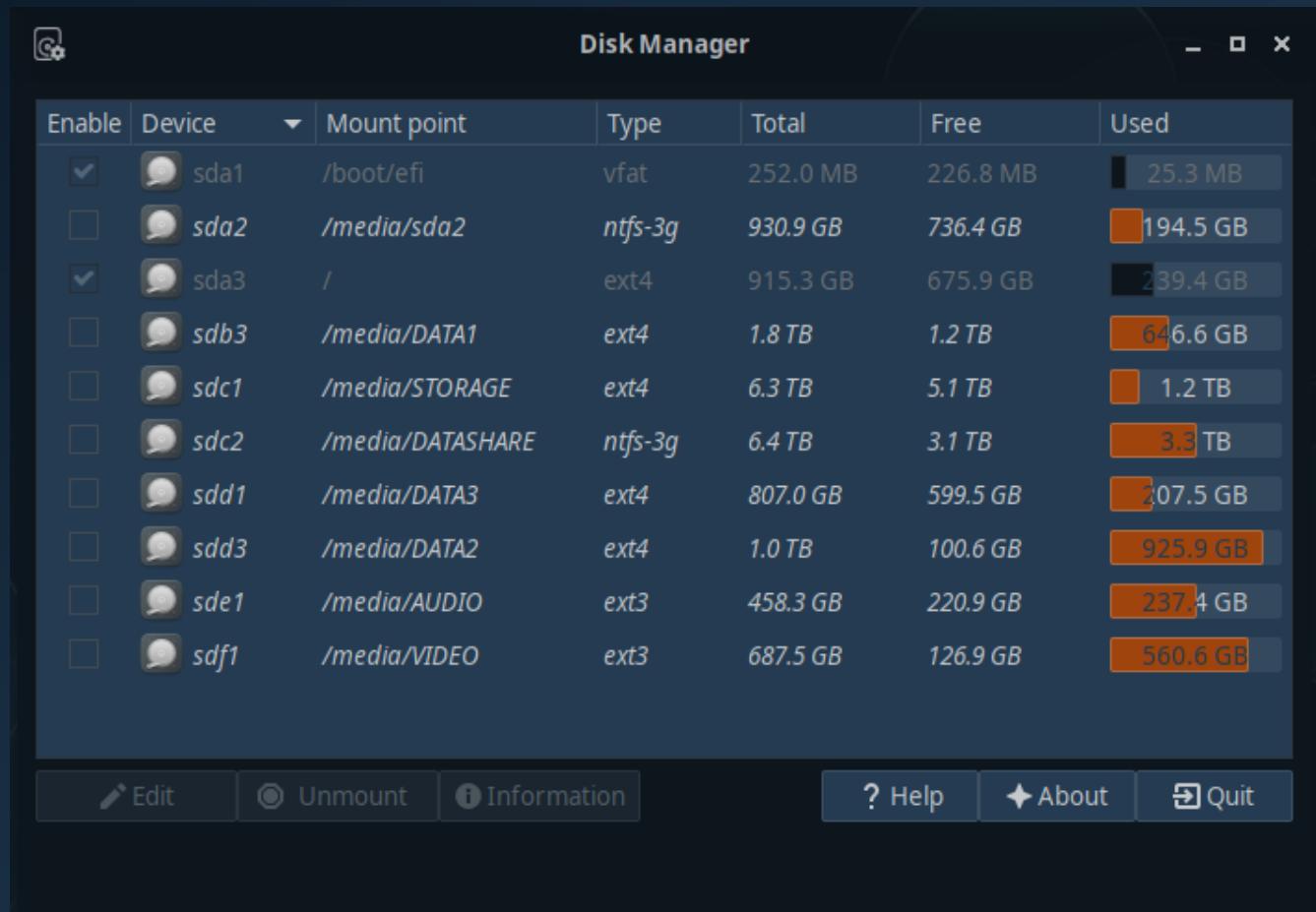
2. The MX-Tools 'Tweak' method:

The MX-Tools which can be found in both the Openbox and XFCE4 Whisker Menus provides a tool called 'Tweak' which has many useful purposes and one of them is to enable the Auto Mounting of Internal Drives by non-Root Users. This functionality can be accessed by checking off that option in the 'Other' Tab of the Tweak application.



3. The MX Disk-Manager method:

MX Linux provides another graphical option for managing Internal Drives and Partitions on the system in its newly revamped and improved 'Disk Manager' application. It is capable of Unmounting and Mounting Drives with its 'Enable' feature and also displays Mount point, Drive type and shows Used and Free space available on Drives.



The AVL-MXE Assistant

Just like MX Linux provides its 'MX-Tools' to assist the User in detailed system settings AVL-MXE provides its own 'AVL-MXE Assistant' to help with various settings that pertain both to setting up an effective Multimedia Workstation and to keeping things working reliably during long projects. The Assistant is a modular Dialog-based Utility with 5 individual windows.

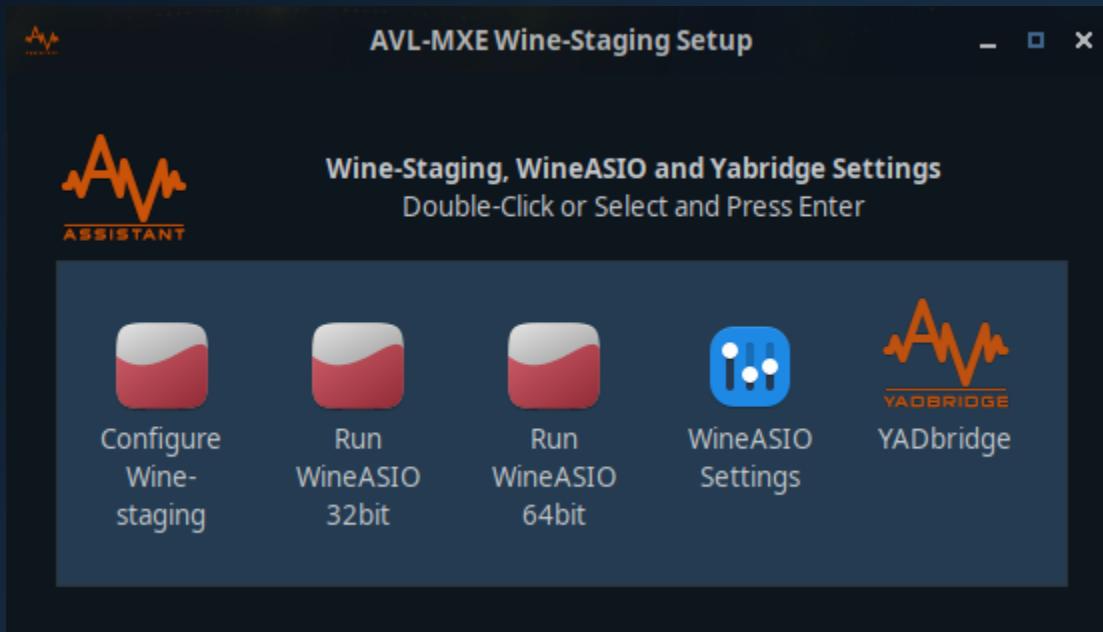
Quick Settings:



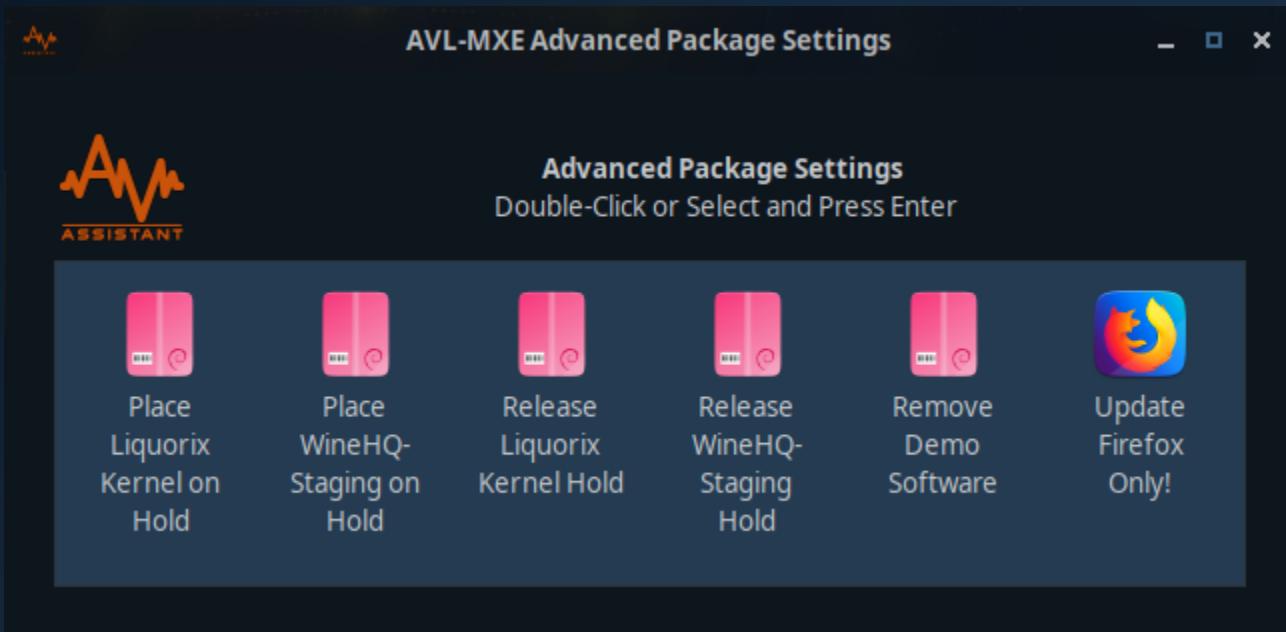
This Dialog is primarily for consolidating the applications that set up the eye-candy of your system all in one easy to use area. Here you can set the Desktop Wallpaper with 'Nitrogen', you can choose separate wallpapers for each screen if you run dual monitors or you can choose to stretch a single image over all monitors. Displays can be configured with the XFCE4 Display Utility. System GTK Themes, Icons and Fonts can be changed here as well as the Openbox Window Themes. The 'QT5CT' Utility allows making visual changes to applications that use the QT5 Toolkit and lastly you can configure the Desktop effects and compositing of 'Compton' the default Window compositor.

System Utilities:

This Dialog collects a variety of useful System Utilities together. You can save or retrieve custom JACK/ALSA Audio Routings with 'aj-snapshot'. There is a handy script to convert Beats per Minute to milliseconds. View system resources stats and consumption with 'bpytop'. Re-enable the setting to launch scripts from the Thunar File Manager if needed (it is enabled by default in AVL-MXE). If you have unwanted jackd or jackdbus processes running in the background they can be stopped. A handy diagnostic tool the 'Realtime Config Scan' will display which performance tweaks are enabled on the system in a terminal-based script. Icon Theme caches can all be refreshed with a single button and the CPU Governor utility both shows which Governor is running and also allows changing it. Lastly there is a fun utility to create Gradient wallpapers by selecting 4 Colors from your Display.

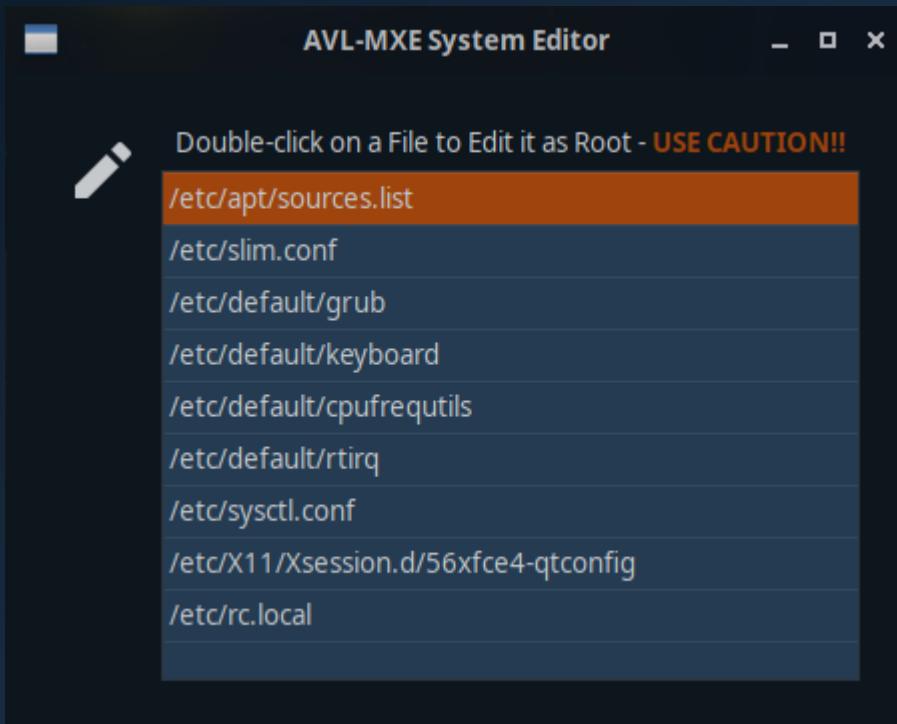
Wine Settings:

The Dialog for Wine Settings is a one-stop shop for the unique needs presented by supporting Windows ASIO Audio drivers for JACK and running Windows VST 2/3 Audio Plugins on Linux. AVL-MXE utilizes the 'Wine-Staging' Windows compatibility layer and this Dialog facilitates behind the scenes setup. Here you can initialize and set up Wine-Staging by launching its configuration panel, run, setup and register the specialized 'WineASIO' dll to support both 32bit and 64bit Windows Audio applications. The handy new 'YADbridge' Utility provides you a graphical interface to setup the 'Yabridge' Windows VST bridge.

Advanced Software Packages Settings:

This Dialog is intended for experienced Users and provides the ability to place both the Liquorix Kernel and Wine-Staging on hold and subsequently release them from hold. In addition it allows for upgrading the default Web Browser (Firefox) by itself without performing a blanket upgrade on the System. The rationale for this Dialog is to bolster stability in situations where the System is being used for long-term or high priority paid work when Browser security is crucial to maintain but other core System assets are preferred to remain unchanged.

System Administration Editor:



The System Editor is to make the System Administrator's job easier by putting several Core System Configuration files in one place to edit. For most Users this should neither be needed or necessary. The Dialog will automatically open the configuration files for Editing in Featherpad, once edits are complete the files must be saved in Featherpad for the changes to be applied.

The Kernel Conundrum

On any Linux system the Kernel is the core that everything runs on, it provides the framework that connects all the Hardware in your computer to the Software applications to use. The better match a Kernel is to your individual computer the better your system will work. The Linux Kernel is absolutely mindblowing in the huge amount of Hardware it supports over various needs from old PCs to the latest Smartphones. In a case like AVL-MXE I am hoping to support anything from new computers to machines that are more than ten years old. Old computers are excellent candidates to set up as dedicated recording ‘appliances’ and even old Single and Duo Core machines can record several tracks of Audio quite easily. When trying to support computers over such a span of age and additionally provide the best low latency performance for Audio recording it becomes difficult for a single choice of Linux Kernel to be all things to all people.

The primary focus of AVL-MXE is to be a Multimedia Workstation and enable the user to create content with the best possible performance. To this end in the past I provided a customized Kernel with ‘full preemption’ also known as an ‘RT’ Kernel. RT Kernels provided the best potential to provide the low Audio latencies required but they were somewhat like racing engines and were not always the best choice for all computers, they also had a crippling limitation of not easily being open to the installation of 3rd-party proprietary Video drivers by AMD and nVidia so if the Kernel Xorg AMD Drivers are not to your liking or you needed better nVidia support the default RT Kernel in AVL-MXE was not the best choice.

The first thing to determine is if everything is working as you want and expect, if your Hardware is all supported and Software is all behaving as you expect then my suggestion is to leave the Kernel as it is and not make any changes. A common fallacy is that a newer Kernel is a better Kernel and that is often not the case, if your system is well-supported and you are confident that the Kernel has all of the appropriate security patches then installing a newer Kernel could introduce new issues and actually have a detrimental effect. The old adage “if it ain’t broke don’t fix it” is a good rule to follow!

Starting with AV Linux MX-21 Edition the ‘Liquorix’ Kernel is now the default offered. Liquorix has long been known for its high-performance and on most systems it provides equivalent if not better Audio latency performance than full-preempted ‘RT’ Kernels, additionally it allows for the installation of Proprietary nVidia and AMD Video Drivers.

XFCE4 with Openbox *New!

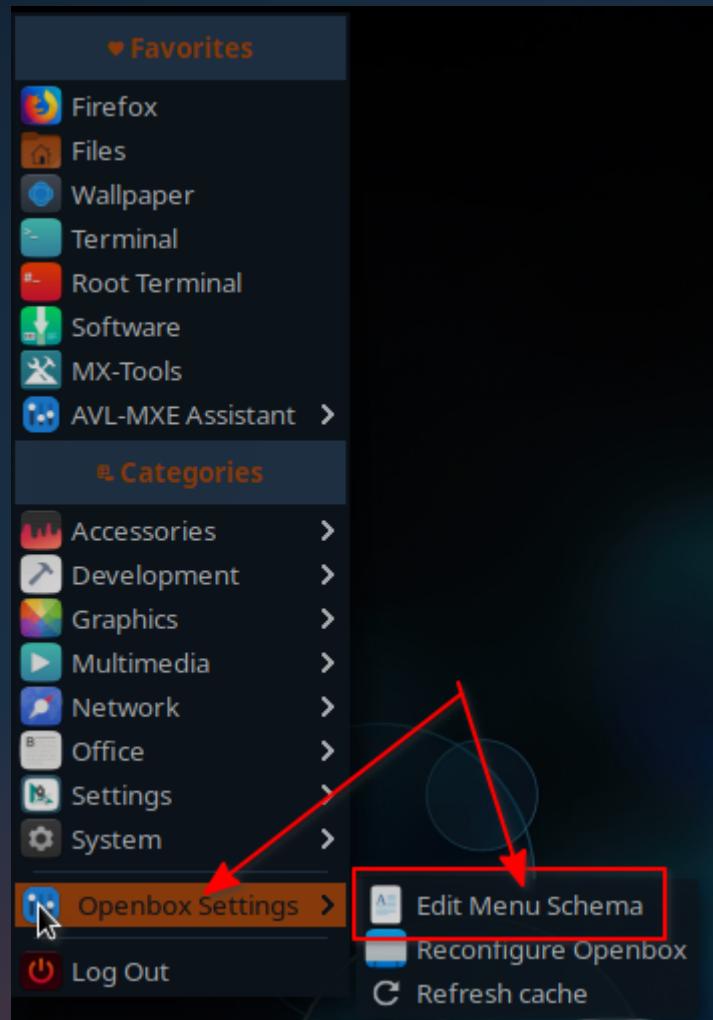
One of the greatest attributes of Linux is the ability to fine-tune the Graphical Desktop Environment to maximize efficiency across a wide cross-section of new and old computer hardware. AVL-MXE has long utilized the XFCE4 Desktop Environment which has been well-known as a mid-weight option combining a decent balance of convenience and efficiency however as Desktop and Laptop computers get more powerful XFCE4 has tended to get more resource-heavy with each release to the point I decided to intervene and replace the XFCE4 Window Manager (xfwm) and the XFCE4 Desktop Utilities (xfdesktop) with the much lighter and well-known 'Openbox' Window Manager. The modular nature of XFCE4 makes this customized hybrid possible and with the performance gains come a few caveats and key differences to be aware of:

- Openbox has it's own menu that is accessed by right-clicking on the Desktop.
- You can choose using the quick Openbox menu or the searchable XFCE4 Whisker Menu.
- The Openbox menu can be modified by text-editing the 'obmenu-generator' schema file.
- Openbox does not 'manage' the Desktop *so it does not provide for icons on the Desktop*.
- Settings for Openbox are handled by the Openbox Configuration Manager (obconf).
- Desktop wallpaper is handled by the lightweight application 'Nitrogen'.
- The XFCE-panel is retained allowing access to usual Panel and Systray functions.
- Log out and Shutdown is handled by a custom utility independent of XFCE4.



Editing the Openbox Menu

Openbox has been around for a long time and has always been very lightweight and very configurable, it may not seem simple to new Users with regard to how to make Menu modifications. Modifications and changes to the Openbox Menu are made through text-editing various configuration files not through a Graphical Menu tool. This may seem to be confusing and daunting at first but once you get acquainted with it you will see that it is a very powerful way to get things configured just the way you may want. The configuration tools for Openbox are an item in the default Openbox Menu layout in AVL-MXE.



What appears in the Openbox Menu when you right-click on your Desktop is determined by the Menu 'schema' file which can be opened by clicking 'Edit Menu Schema'. This schema text file handles the Menu configuration as part of an Openbox helper application called 'obmenu-generator' so to clarify this file is read by 'obmenu-generator' which in turn tells the native Openbox Menu component what to display. The scope of this manual does not include a full tutorial on editing the Menu schema but we will take a quick look at the 'Favorites' section of the Menu in case you would like to add or remove an Application launcher.

First here is how different elements of the menu are defined:

```
# obmenu-generator - schema file

item:      add an item inside the menu
cat:       add a category inside the menu
sep:       horizontal line separator
pipe:      a pipe menu entry
file:      include the content of an XML file
raw:       any XML data supported by Openbox
beg:       begin of a category
end:       end of a category
obgenmenu: generic menu settings
exit:      default "Exit" action

{item => ["command", "label", "icon"]},
{cat => ["name", "label", "icon"]},
{sep => undef}, {sep => "label"},
{pipe => ["command", "label", "icon"]},
{file => "/path/to/file.xml"},
{raw => q(...)},
{beg => ["name", "icon"]},
{end => undef},
{obgenmenu => ["label", "icon"]},
{exit => ["label", "icon"]},

# NOTE:
#   * Keys and values are case sensitive. Keep all keys lowercase.
#   * ICON can be either a direct path to an icon or a valid icon name
#   * Category names are case insensitive. (X-XFCE and x_xfce are equivalent)
```

So to expand, a Menu 'item' will need (in order from left to right) to provide the 'command' that the application is launched by, the 'label' or what name you want to application to appear with and then the correctly named 'icon' that you want shown beside the application name. You can also create labeled or unlabeled separators between your items and create categories for the items to appear within. Let's take a look at the schema for the 'Favorites' section of the menu to better understand how this works.

```

our $SCHEMA = [
    {sep => '♥ Favorites'},

    #           COMMAND             LABEL           ICON
    #{item => ['minstall-pkexec', 'Installer', 'minstall']},
    {item => ['firefox', 'Firefox', 'firefox']},
    {item => ['xdg-open .', 'Files', 'folder-home']},
    {item => ['nitrogen', 'Wallpaper', 'nitrogen']},
    {item => ['xfce4-terminal', 'Terminal', 'utilities-terminal']},
    {item => ['mx-pkexec /usr/bin/x-terminal-emulator', 'Root Terminal', 'root']},
    {item => ['su-to-root -X -c mx-packageinstaller', 'Software', 'synaptic']},
    {item => ['mx-tools', 'MX-Tools', 'mx-tools']},
    {beg => ['AVL-MXE Assistant', 'preferences']},
        {item => ['evince /usr/share/doc/avl-mxe-user-manual/AVL-MXE-User-Manual.pdf', 'Read the AVL-MXE User
Manual', 'help']},
        {item => ['avl-mxe-qs', 'Quick Settings', 'preferences']},
        {item => ['avl-mxe-util', 'System Utilities', 'preferences']},
        {item => ['avl-mxe-ws', 'Wine Settings', 'preferences']},
        {item => ['avl-mxe-pkg', 'Package Settings', 'preferences']},
        {item => ['avl-mxe-syse', 'System Editor', 'preferences']},
    {end => undef},
]

```

We can see in the example above the order of Command, Label and Icon, we can also see the items that appear directly in the Menu and also where the beginning of the 'AV Linux Assistant' Pipe Menu begins between '{beg=>' and '{end=> undef}'. Another important thing to note is that we can make Menu items not visible in the Menu simply by using a hash '#' at the beginning of the line of the item we want to make invisible. A common example of wanting to do this is if we have installed AVL-MXE to the HDD and no longer want or need the Installer Menu item to show, in the example above the Installer line has been 'commented out' with a '#'.

Slim Login Manager

AV Linux features 'Slim' as its default login manager. Slim has a very light system footprint and is still quite configurable by editing its configuration file in /etc/slim.conf. The example below will cover the most popular configuration request: autologin.

You can open the Slim configuration file for editing from the 'System-Editor' tab of the AV Linux Assistant.

To enable autologin two separate variables must be changed in slim.conf. Scroll down the file to find the following line:

```
# default_user
```

To set yourself as the default User remove the '#' comment from the beginning of the line and then add your Username, USERNAME is used here as an example:

```
default_user USERNAME
```

If we leave the file like this then on next login it will automatically enter your Username but still require your password, to have autologin work fully you need to edit the autologin line:

```
# auto_login no
```

Once again remove the '#' comment and change the 'no' to a 'yes' like this:

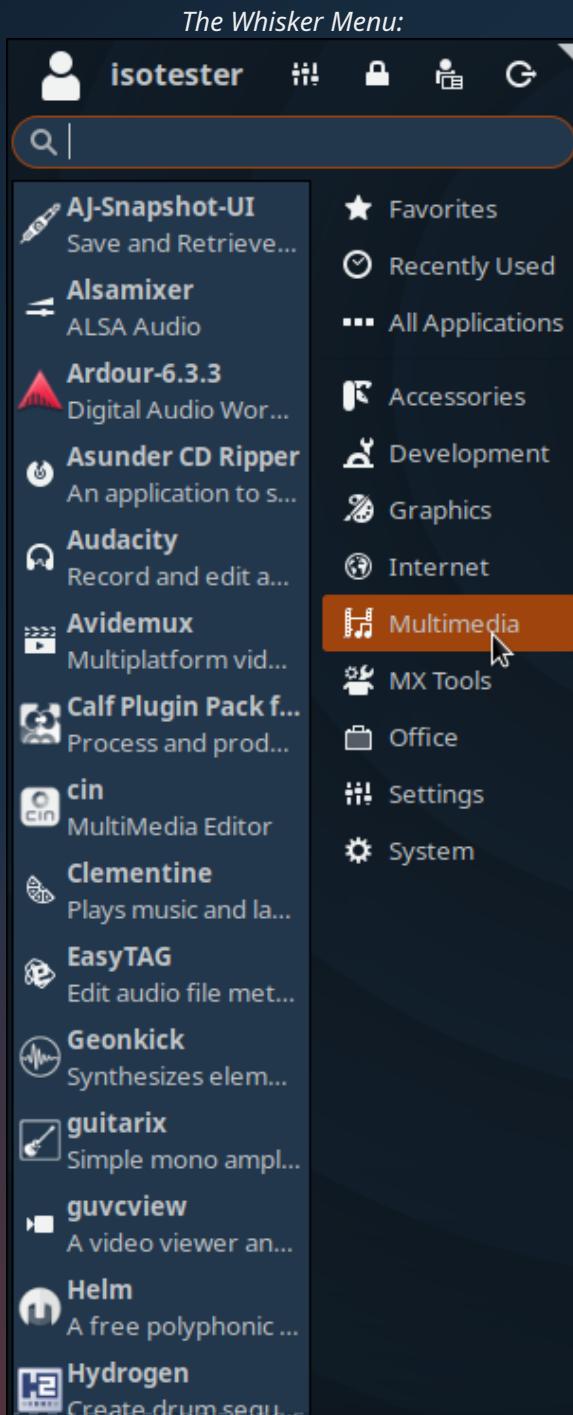
```
auto_login yes
```

When you have completed editing the slim.conf file click 'Save' in the Mousepad Text Editor and autologin should be active on your next login.

***PLEASE NOTE! - Slim does NOT support remote login, if you require remote login or are running JACK or OSC protocols over an internet connection it is recommended to install 'lightDM' from the Debian repositories and configure it as the default login manager.**

Getting Around in XFCE4

AVL-MXE uses the XFCE4 Desktop Environment for its great blend of functionality and efficiency. The XFCE4 'Whisker Menu' allows you to browse Applications as you would in a regular Menu as well as providing a Search bar that you can type the name of the Application into. Right-clicking on the Menu button and selecting 'Properties' will allow you to tweak the Whisker Menu Settings.

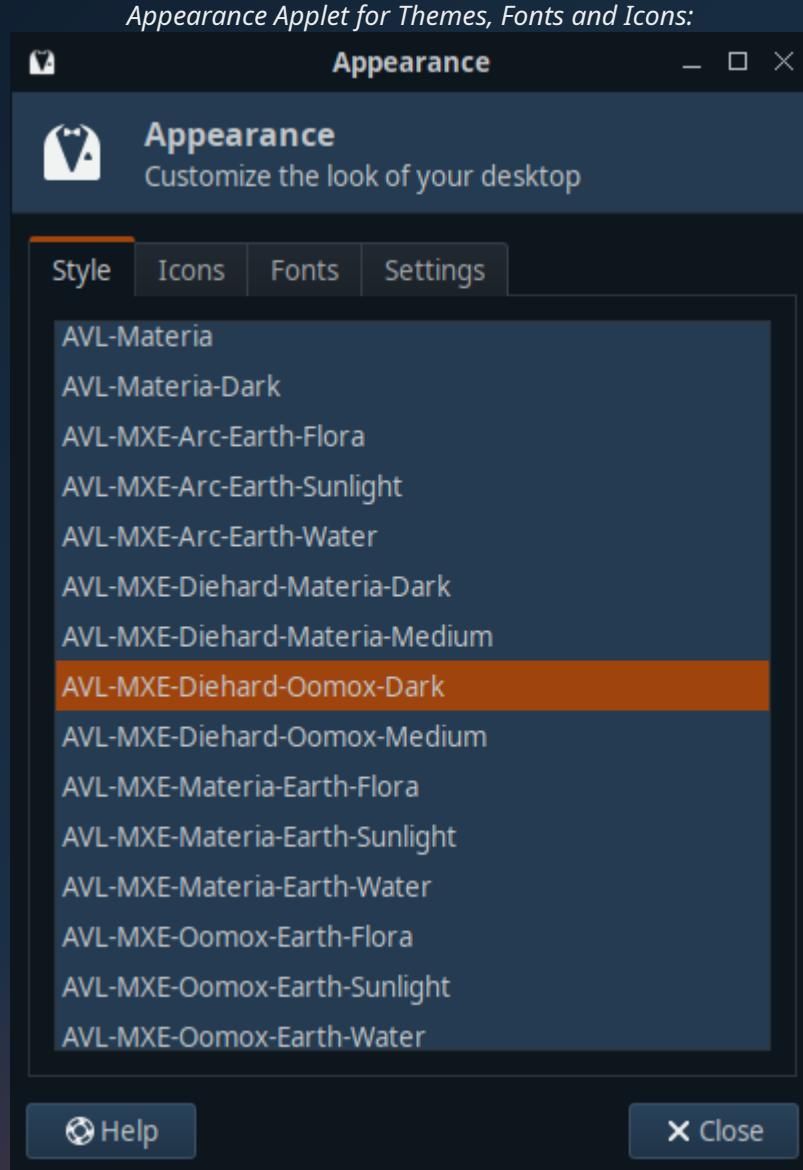


The settings for XFCE4 are accessed by using the 'Settings Manager'. Find it by either typing 'Settings' in the Whisker Menu search bar or navigate to 'Settings' → 'Settings Manager'.

The XFCE4 Settings Manager:

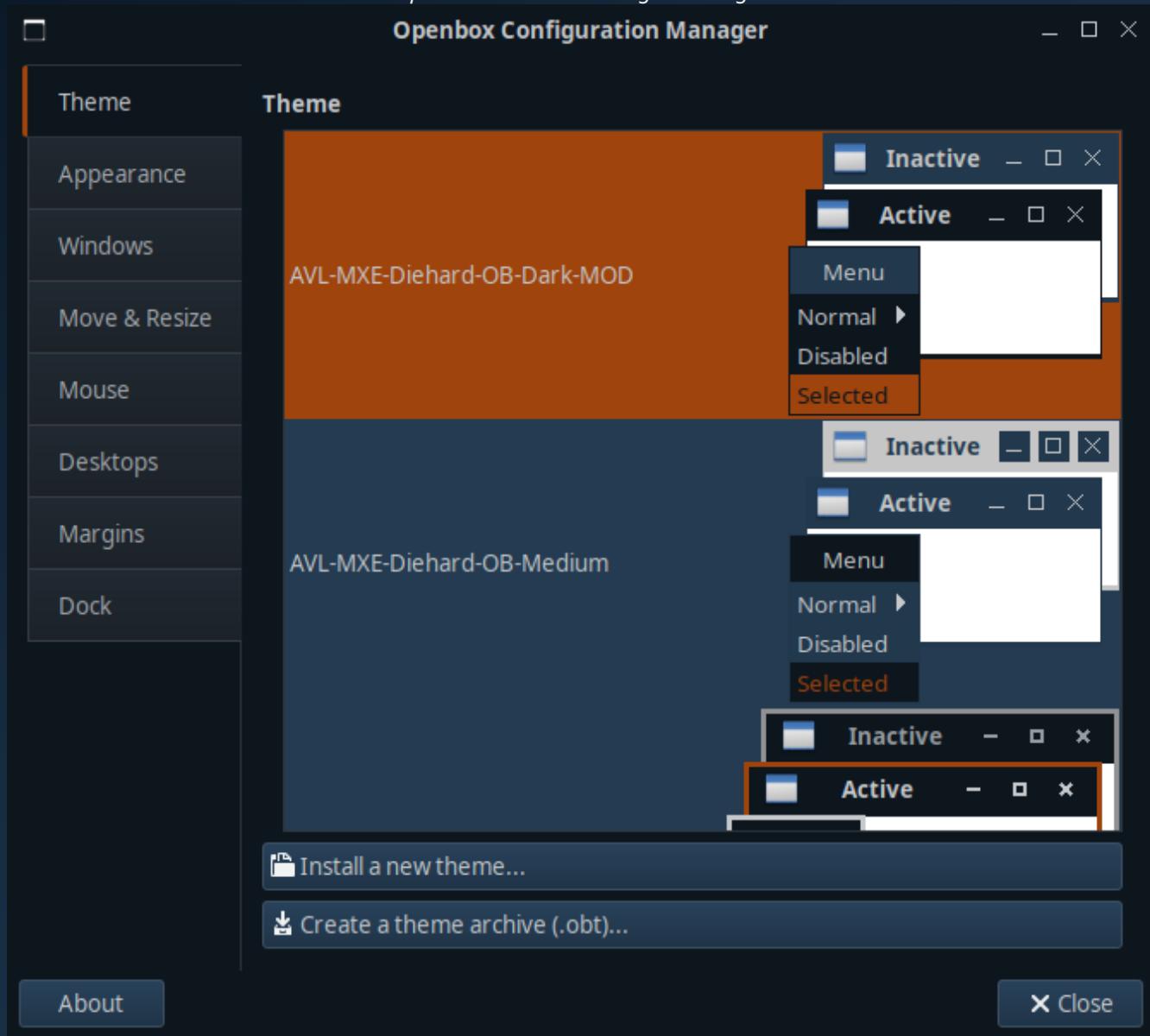


You can change how the system looks with the 'Appearance' settings. This will allow you to set the GTK theme, system icons, system fonts and some menu related items. AVL-MXE comes pre-loaded with a large amount of themes and colors to choose from.

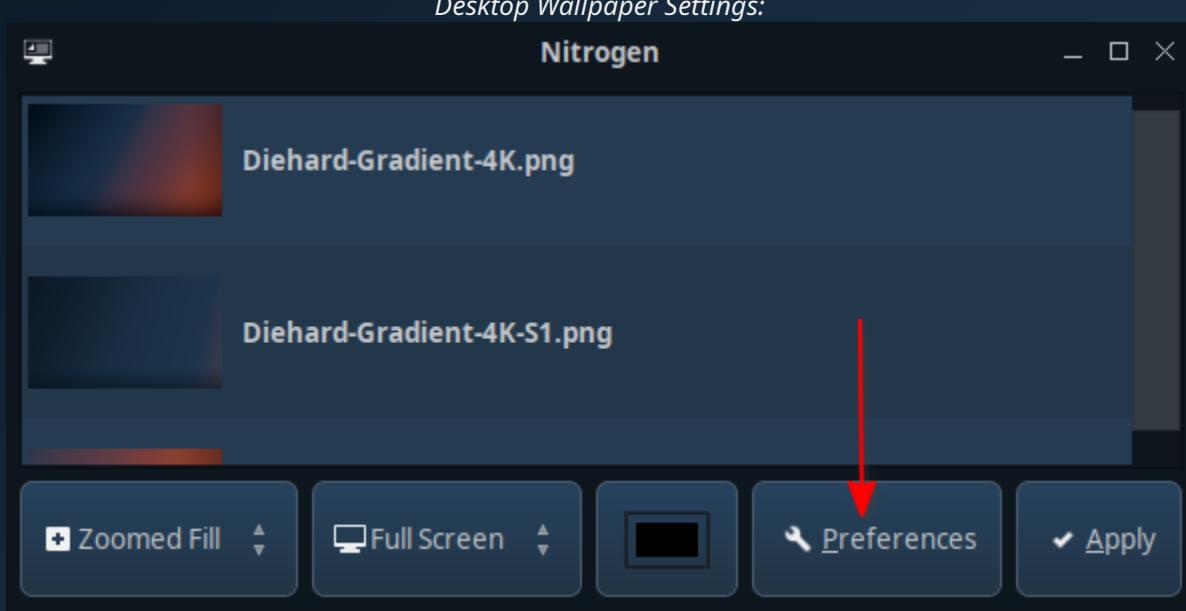


The Openbox Window Manager themes are set in the 'Openbox Configuration Manager' .

Openbox Window Manager Settings:



Change your Desktop wallpaper and other settings with the 'Nitrogen' wallpaper application. It allows you to set wallpaper images for single screen or individual wallpapers for dual-head setups.



The 'Preferences' button will allow you to select how the listed wallpapers appear and add the directories containing the images you want to use for Desktop backgrounds.

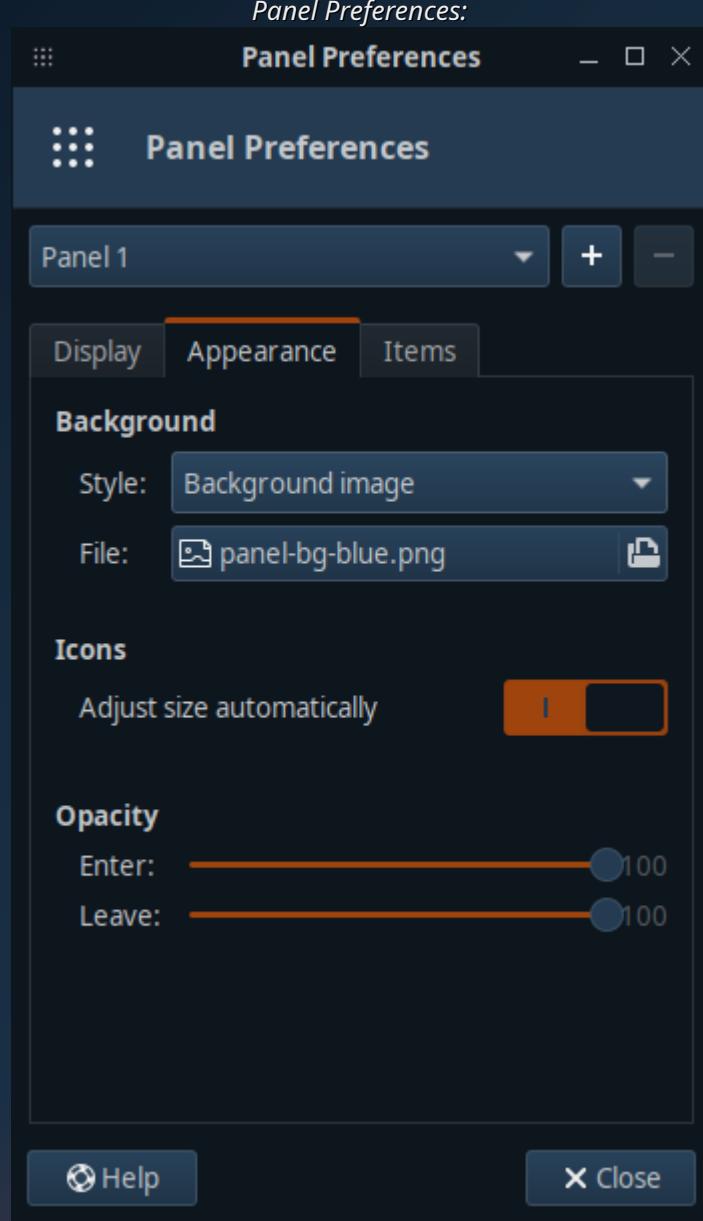
The XFCE4 Panel at the bottom of the Desktop screen combines access to the menu with program launchers, a window menu, PulseAudio sound mixer, notification tray, system clock and session logout buttons. By default the menu is set up to look like a “Taskbar” but right-clicking on any item on the Panel and selecting ‘Panel’ → ‘Panel Preferences’ will open a window to configure the Panel however you want.

Panel Left Side: Menu and Launchers



Panel Right Side: Window Menu, PulseAudio, Systray and Logout

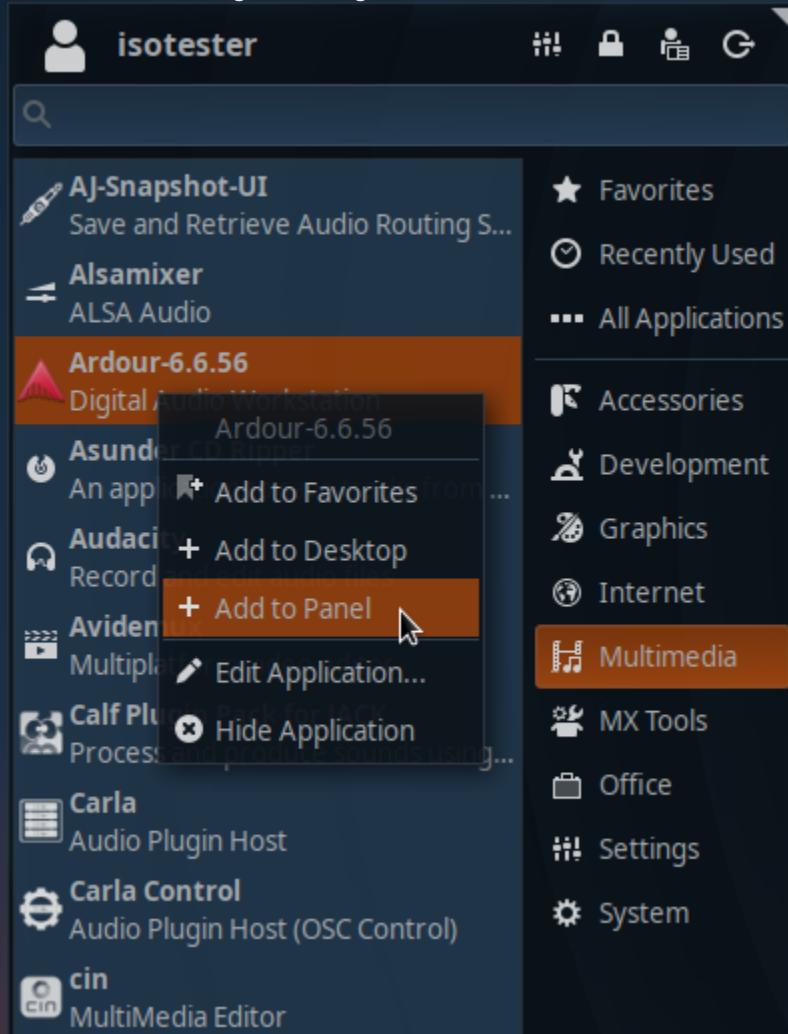




It is common for users to want to take application launchers from the Whisker menu and add them to the Panel. This is actually routinely necessary for the Ardour and Mixbus32C DAWs. Most applications retain their launcher properties and names when updated and once added to the Panel they don't need changing. Ardour and Mixbus32C generate launchers containing their version numbers as part of the Application name so when you update Ardour or Mixbus32C it will unfortunately update the launcher as well so your existing Panel launcher will no longer work after the update. We will use Ardour as an example of adding a Menu launcher to the Panel.

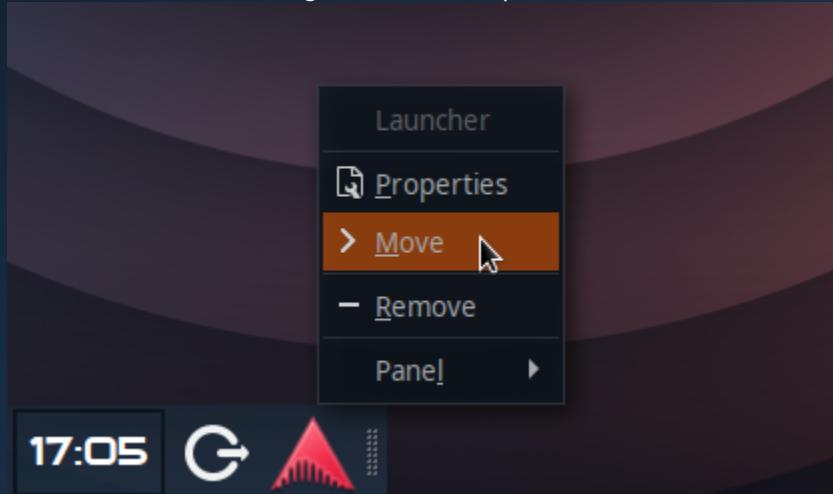
First find the launcher you want to add in the Whisker Menu and then right-click on it which will present ' + Add to Panel' as one of the options..

Right-clicking on Whisker Menu:



When you add the launcher to the panel it will appear to the far right at the very end of the Panel which of course is most likely not where you want it to be. In order to place the launcher in a different position within the Panel you will need to enable it to 'Move' so it can be dragged to the desired position. To do this right-click on the new launcher and a '→ Move' option will be available. When you select the launcher with the mouse cursor and start to drag it a vertical red line will appear that will help you line it up to its new position the Panel.

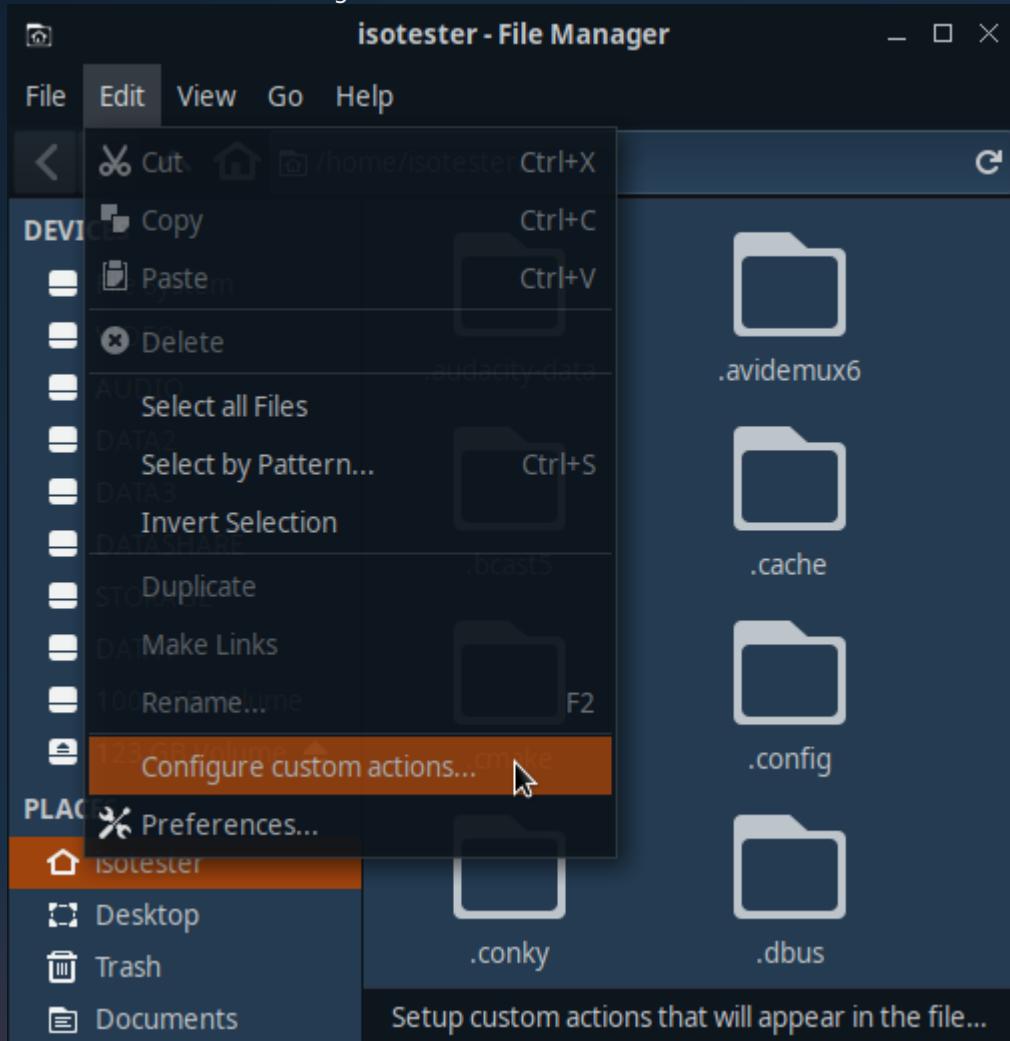
Right-click 'Move' option:

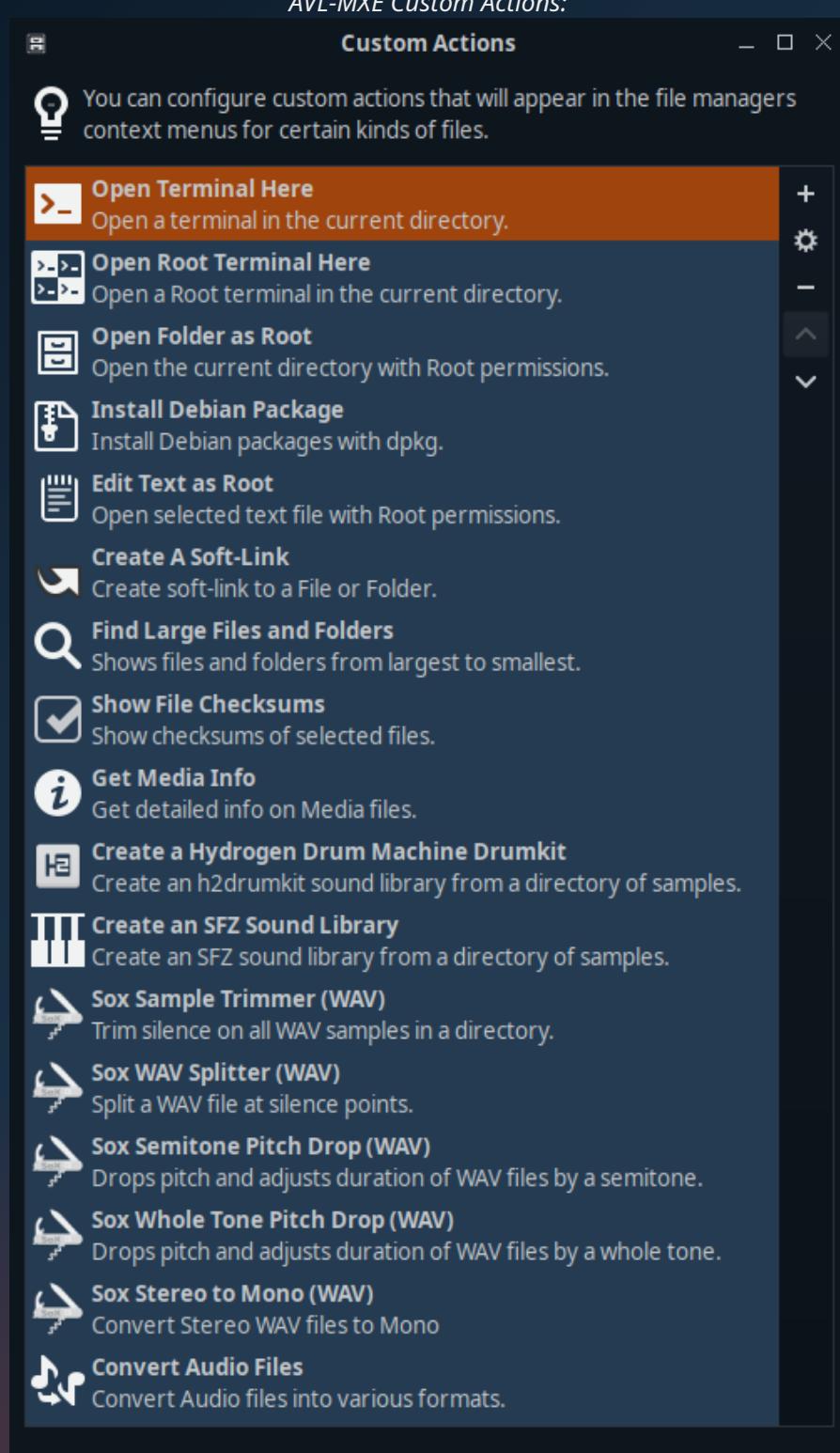


Thunar File Manager

The Thunar File Manager included in AVL-MXE has been customized and extended with Custom Actions to do MUCH more than simply manage files. As you navigate the System you will see there are numerous options when you right-click on the Desktop and in folders , more different options appear when you right-click on certain file types. Custom Actions enable many tasks including system administration, media playing, media conversions, networking and much more! A summary of the Custom Actions is shown below, most options are self-explanatory. Some will be explained further on following pages.

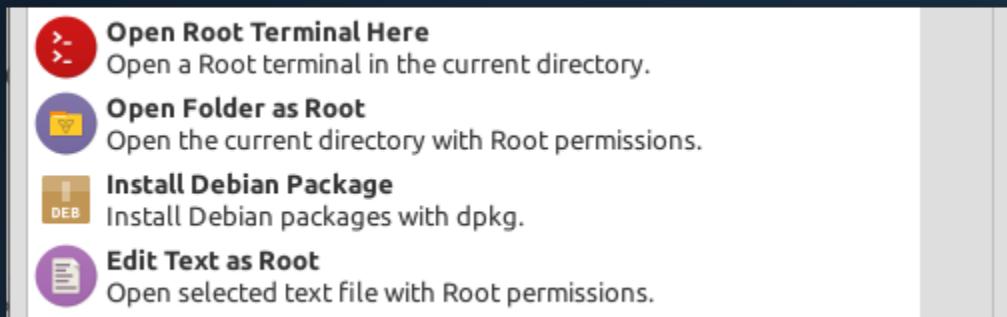
Accessing the Custom Actions Menu in Thunar:



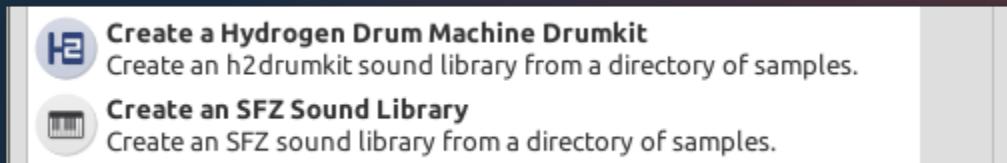


Root Custom Actions (USE CAUTION!) :

For those who need to administrate their system and make changes outside of their home folder in the file system some handy Root Custom Actions are provided. These allow to open a Root Terminal within the folder you're viewing, open the current folder with Root privileges, Install a Debian package from Thunar (simple install, does not resolve external dependencies) and to Edit configuration text files as Root.

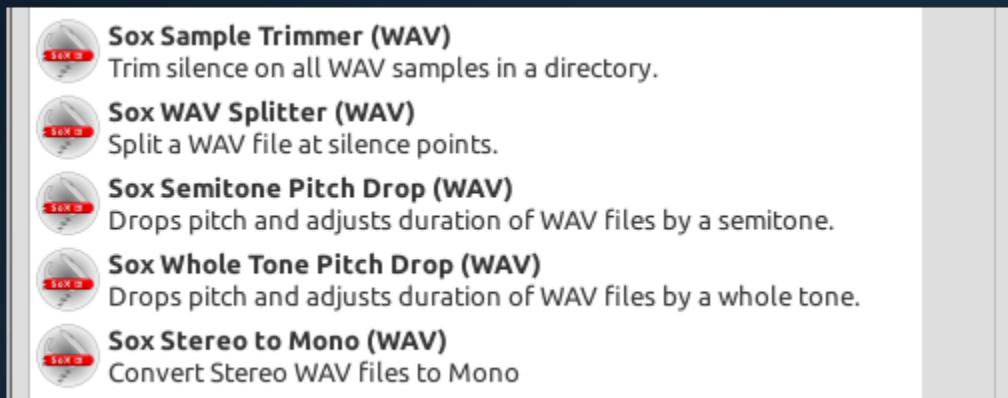
**Create SFZ Sound Libraries and Hydrogen Drumkits :**

From within Thunar you can launch interactive scripts to create SFZ and Hydrogen sound libraries. It is essential that you have your sound samples organized and properly formatted for the target application in a folder first.

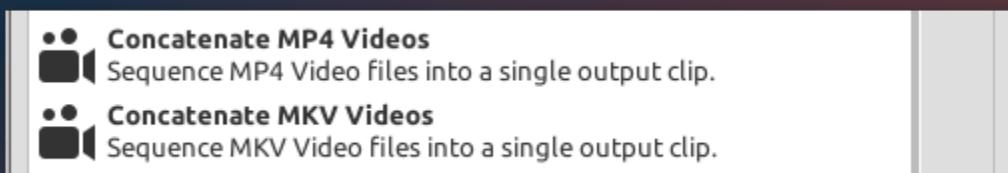


Split, Trim, Transpose and Convert WAV Samples with SoX Utilities :

This is a brand new feature in AVL-MXE 2020 and will probably require further future polishing but it currently allows for quick n' dirty manipulation of WAV files to prepare them for use in Sound libraries all from the comfort of Thunar! The SoX scripts themselves can be found in `usr/local/bin` and can be modified there for those who are savvy with SoX command line.

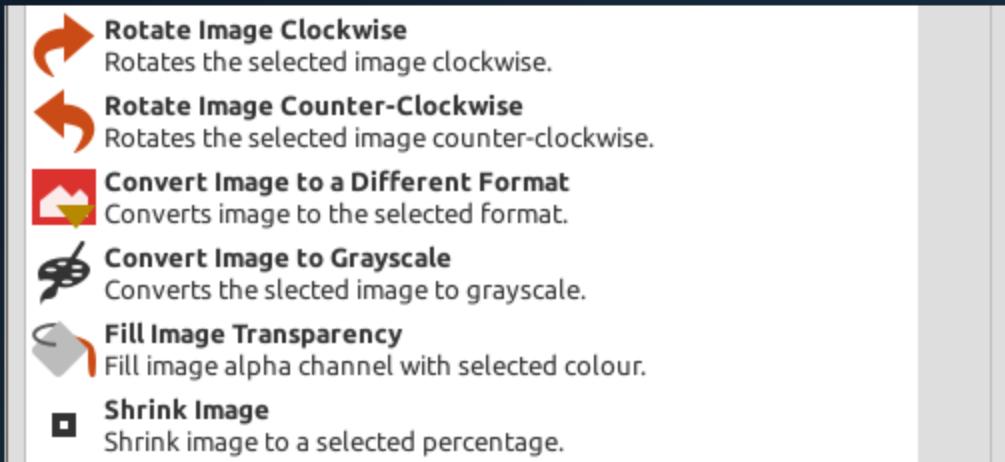
**Concatenate MP4 and MKV Video files :**

You can join MP4 and MKV Video files together, for example you have finished clips you want to combine without having to reopen them in a Video Editor and re-encode them with the resulting quality loss. If the clips are the same resolution, have been encoded with the same encoder and settings and if they are the same Audio sample rate you can use these Concatenate Custom Actions to join them without re-encoding.

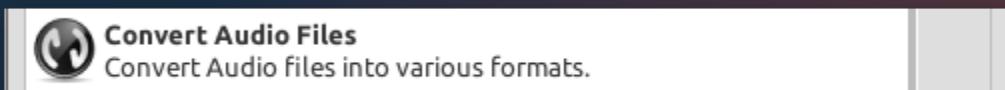


Manipulate, Convert and Work with Image files :

There are some newly added Custom Actions to quickly and easily work with pictures and image files.

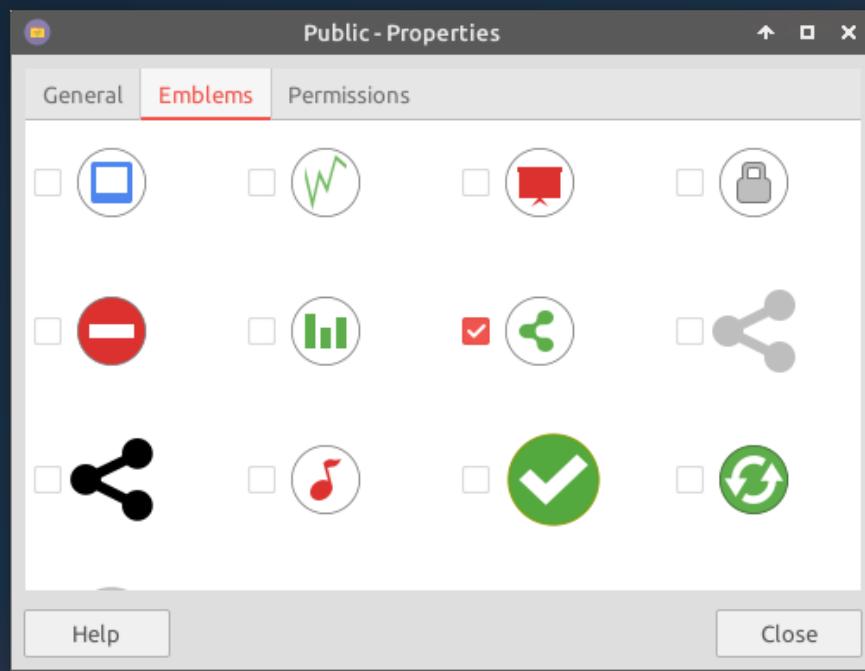
**Convert Audio Files With PACPL :**

You can quickly and easily convert Audio files from one format to another with the integrated PACPL Custom Action. When you right click on an Audio file in a folder you will see the option to "Convert Audio Files" as one of the menu options. Pretty much all common options for conversion are supported.



One-Click Samba Sharing on Your Network :

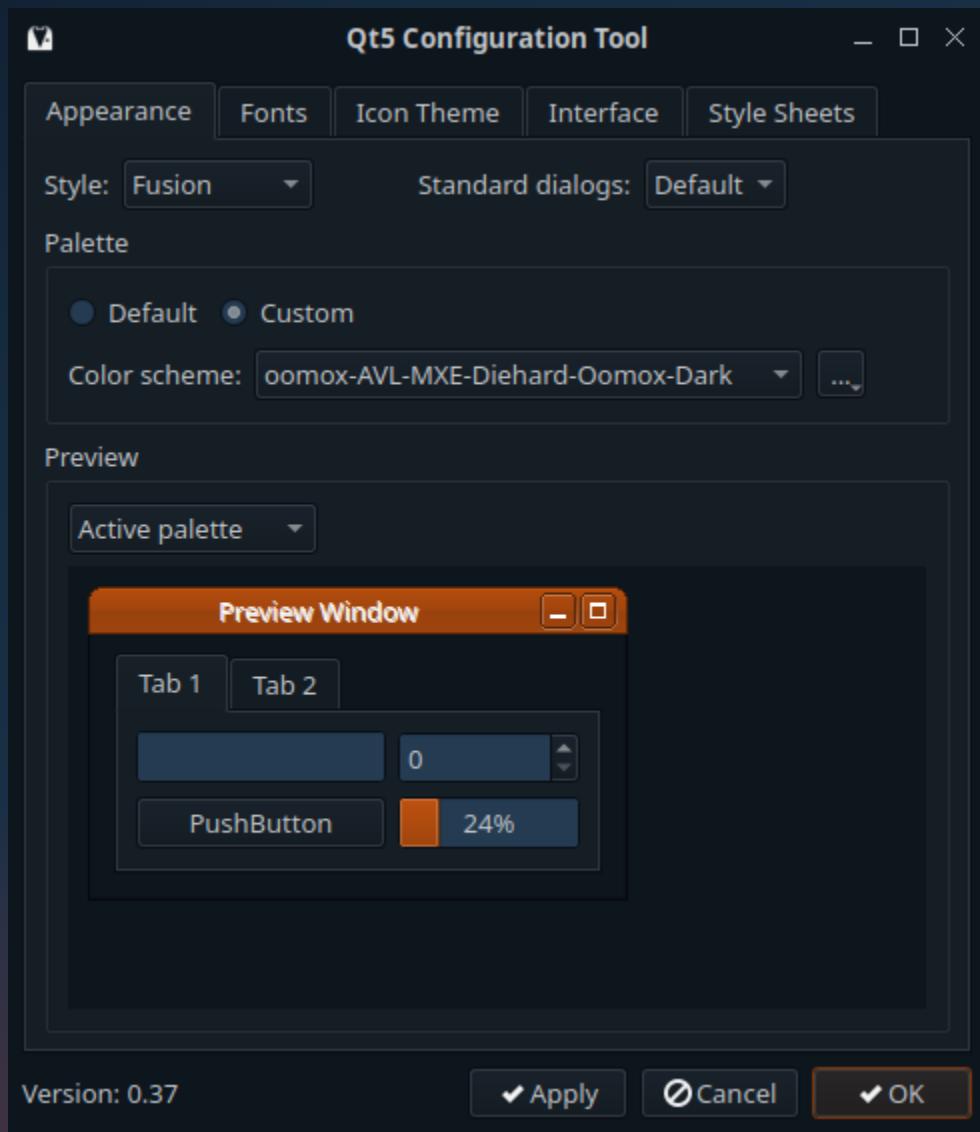
Right-clicking on a folder in Thunar will allow you to 'one-click' share it on your network. If you do share a folder on your network you may want to right-click on it and select a sharing or network 'Emblem' to display on the folder to visually remind you it is shared.



***PLEASE NOTE!!** - "Read and Write" will expose any folders you enable for sharing on your Network with permissions for ALL Users on your Network, if you are concerned about the security of your network, or are not sure which files you are making available DO NOT enable Samba sharing! It is also not recommended to share the root directory of a drive and all its contents on a Network.

QT5 Configuration Tool

When setting up the theme and appearance of your system it is important to note that the XFCE4 Settings for Themes and Icons only affect applications that use GTK2 or GTK3 UI toolkits for their graphical user interfaces. Some popular applications use QT for their GUI's and QT5 requires its settings to be done by a separate 'QT5 Configuration Tool' utility. You can open QT5 Settings either from the 'Settings' Menu or the 'Quick Settings' Dialog of the AVL-MXE Assistant. In some cases it may take a few seconds for it to check your current QT5 configuration and launch. You can set your theme style, fonts, icons and other settings to make QT5 applications match your GTK system themes as closely as possible.

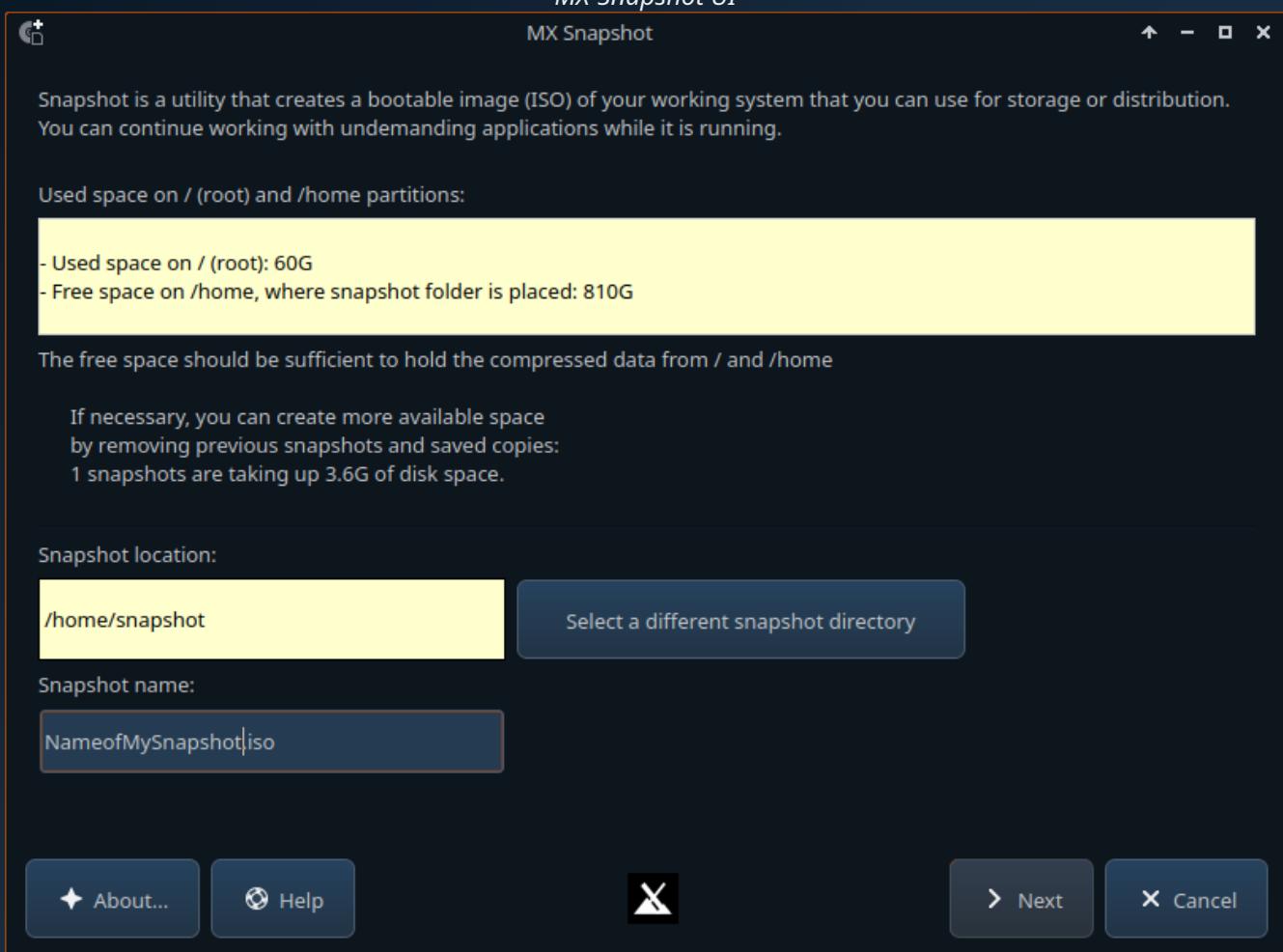


MX-Snapshot in AVL-MXE

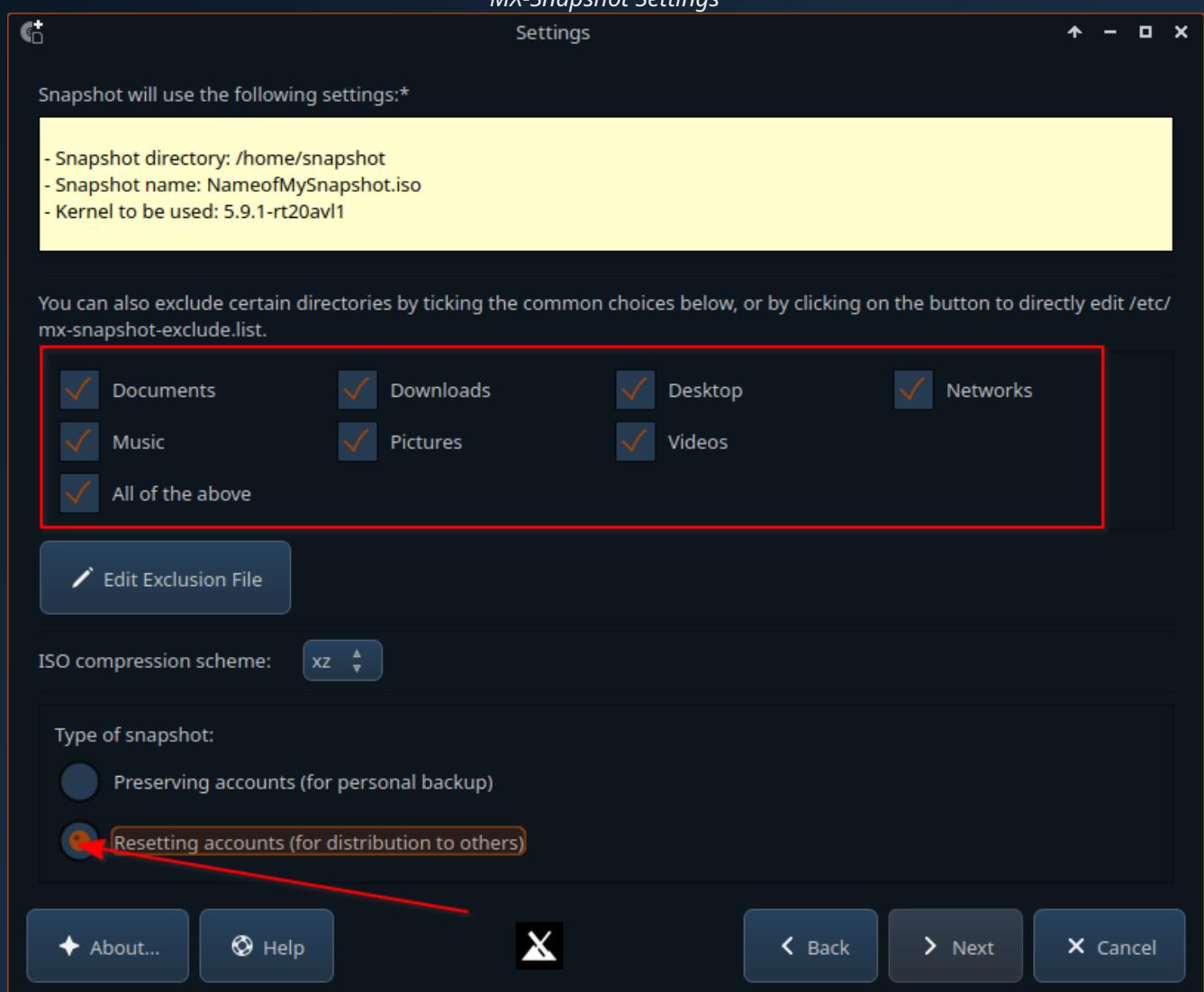
With the 'MX-Snapshot' program found in the 'MX-Tools' of AVL-MXE you can create an ISO image of your system that you can boot with a USB key and run as a LiveISO or install to another computer. This powerful part of MX Linux is what makes AVL-MXE possible and in turn you can add or remove programs and make changes to your installed system and make a bootable ISO of your customized version of AVL-MXE. There are a few important items to be aware of before using MX-Snapshot in AVL-MXE:

- If you are going to make a customized version of AVL-MXE to share with other people to download and use you must first remove the licensed Demo-versions of software using the AVL-MXE Assistant. These demos are specially licensed to be distributed in AVL-MXE only and you will be violating these licenses if you re-distribute them.
- If you are going to create and publicly share your own version of AVL-MXE with other people you must give it a different name and indicate that it is a different project derived from AVL-MXE to prevent confusion for potential Users.

At this point you will be prompted for your Root Password and once entered it will open MX-Snapshot which will give you an option to choose where the ISO and related md5 and SHA512 files will be placed. The default is /home/snapshot/. You can also choose what you want the name for your custom ISO to be.



From here the MX-Snapshot options are as normal, you can select to create an ISO for your own use with your Data and Network configuration preserved or you can choose to make a more generic ISO without your Settings and Data that you can safely share with other people. By selecting the checkboxes for various items you can blend the choices for what you want to preserve from your setup and pass along what you want to exclude from your ISO. Advanced users can also exclude further items in the Filesystem but you should be cautious and familiar with what you are selecting for exclusion. You can also select the type of compression to use which will determine the size of your final ISO, 'xz' gives the most compression and therefore the smallest potential ISO size at the cost of slower performance on the Live booted system. Once you have made your choices click 'Next' to continue, confirm and create your ISO.

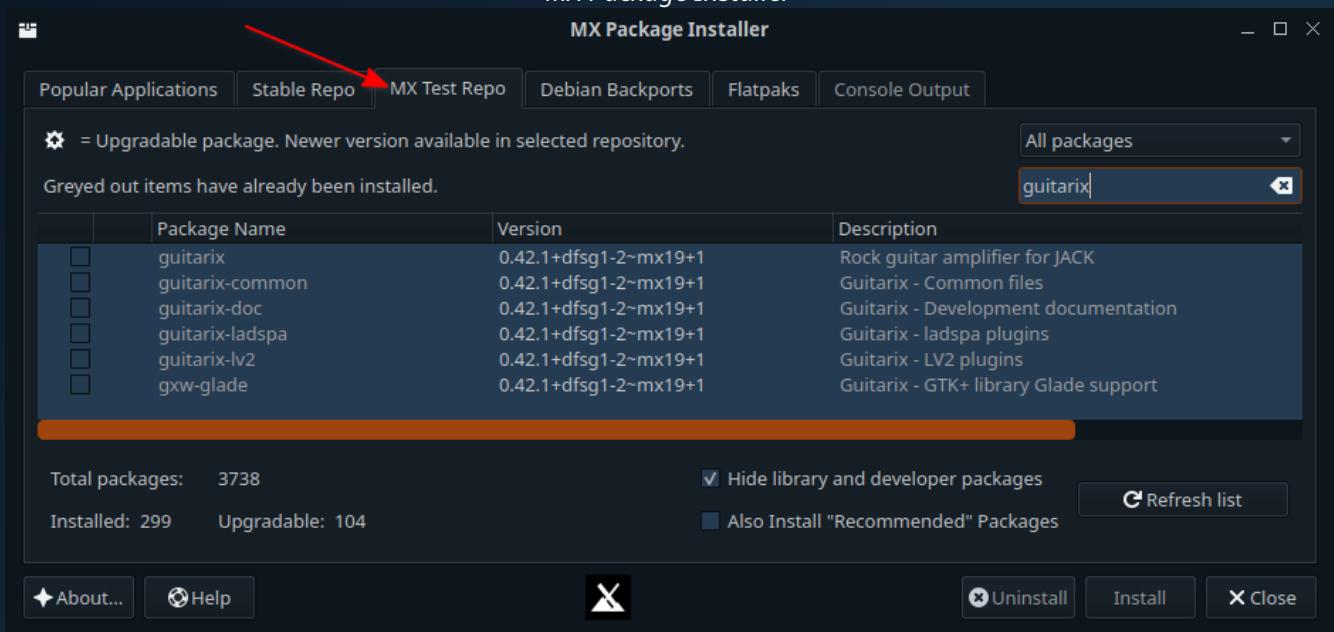


Software Installation Notes

MX Linux (and therefore AVL-MXE) is based on Debian Linux and makes use of its well-known APT/DPKG Package Management. There are some key software installation differences that should be noted between MX Linux and systems using Debian Linux directly. Most Debian Users are familiar with the graphical utility known as the 'Synaptic Package Manager' and AVL-MXE comes with Synaptic installed but it is not necessarily the first recommended method of adding and removing software from your system. The first native MX tool to be aware of is the 'MX Updater' that runs in the systray section of the XFCE4-Panel and is both an indicator of when updates are available and also launches a terminal-based utility to perform the actual updating. When updates are available the MX Updater icon will appear as green and usually a tooltip message will pop up and tell you that updates are available.

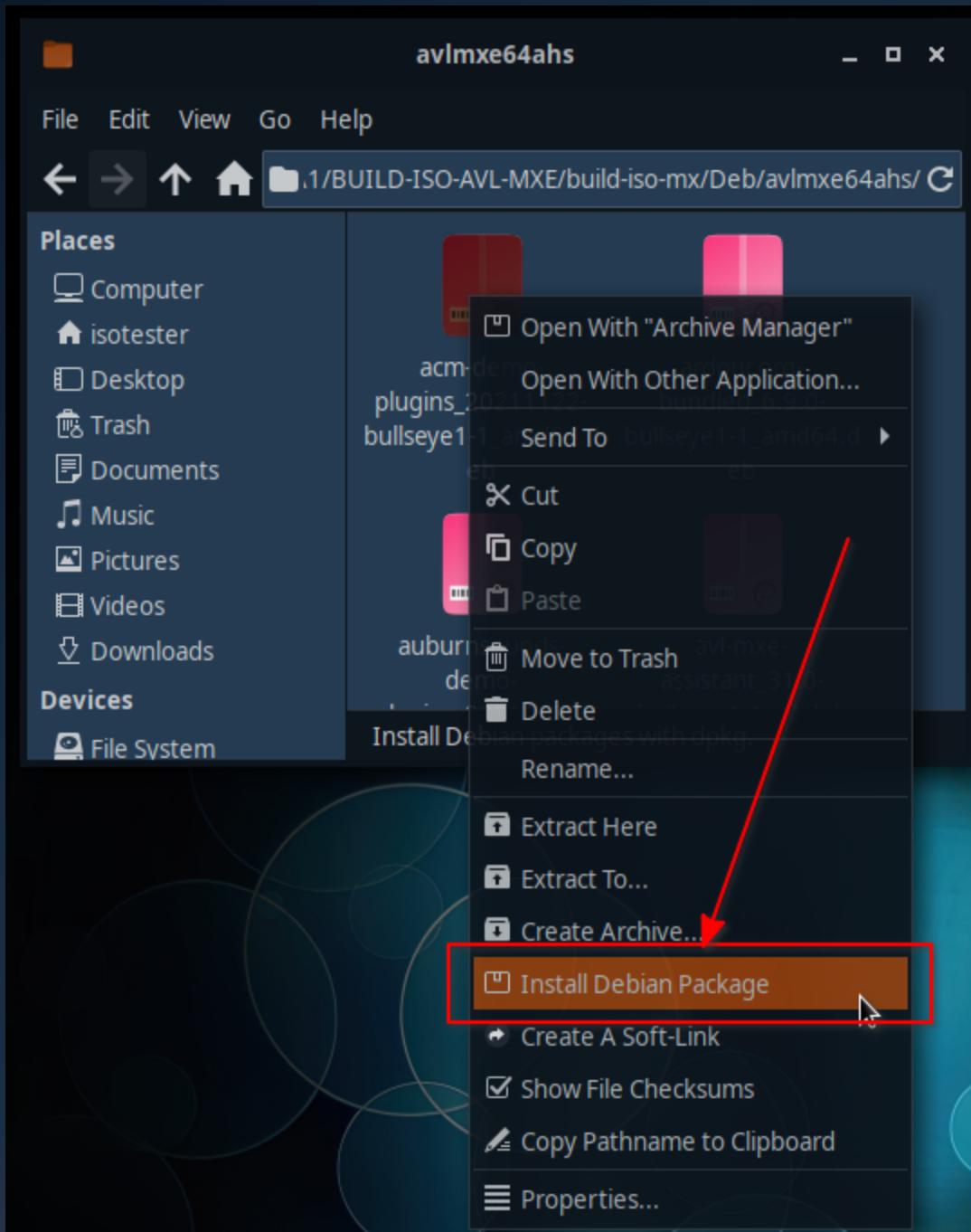


MX Linux has its own powerful Package Manager known as the MX Package Installer (or 'MXPI'). It features a multi-tabbed interface that allows you to install applications from the default software Repositories (aka 'Repos') as you would on a normal Debian system but it has a few interesting extra optional features. The MX developers have an incredibly dedicated team of their own 'Packagers' that fulfill a very important purpose of providing upstream software version upgrades that would not normally be available through the default Debian Linux Repositories. Because of the large amount of work that they do in response to User requests MX Linux has an extra 'Testing' Repo that provides an optional staging area for Users to try out upgraded applications before they enter the default MX Repo. It is NOT advised to have this Repo enabled for use at all times so the 'Testing' tab of MXPI allows you to temporarily enable the Testing Repo and install software from it. For this reason it is discouraged to install Testing packages with Synaptic. If you don't see an updates for a Package you may be interested in upgrading withing the default MX Repos, you may find that it is available in the MX Testing section. In addition MXPI provides the option to temporarily enable 'backported' Package Repos from Debian Linux and a great extra feature is the ability to install self-contained 'Flatpak' applications on the system as well. If you want to play it safe you can simply choose not use these extended features and only install Software Packages from the default Repos provided in AVL-MXE.



*New - Installing non-Repo or Outside Debian Packages

If you download a Debian Package (aka 'Deb') from a Website or location outside of the MX Linux Repository system then you can use the 'Debian Package Installer' Custom Action within the Thunar File Manager. To do this navigate to the location you have downloaded the Package to in the Thunar File Manager and right-click on the Package file. Thunar will recognize the '.deb' file extension and show an 'Install Debian Package' Custom Action to do the installation. This Thunar Custom Action replaces the graphical 'Gdebi' Package Installer application that was in previous versions of AVL-MXE. The Thunar Installer will ask you twice to confirm the installation of outside Debian Packages, never install a Package if you aren't sure if the provider is trustworthy!!



Audio and MIDI in AVL-MXE

Recommended Reading!

PulseAudio with JACK :

It cannot be overstated that AVL-MXE has the potential to differ *significantly* from your usual Linux Desktop in its handling of Audio routing and subsystems. AVL-MXE relies on the low-latency Jack Audio Connection Kit (a.k.a JACK) for its powerful interconnectivity. JACK is a low-latency sound server which runs on top of ALSA (Advanced Linux Sound Architecture). ALSA is the basic 'driver' level of Audio hardware support and is both integrated in the Kernel itself as well as being a core component of the Operating System. By default many distributions and Desktop Environments install the 'PulseAudio' Audio server, in recent years PulseAudio can be made to peacefully coexist with JACK and AVL-MXE makes this easier by pre-configuring JACK and PulseAudio to work together with a custom 'pajackconnect' script .

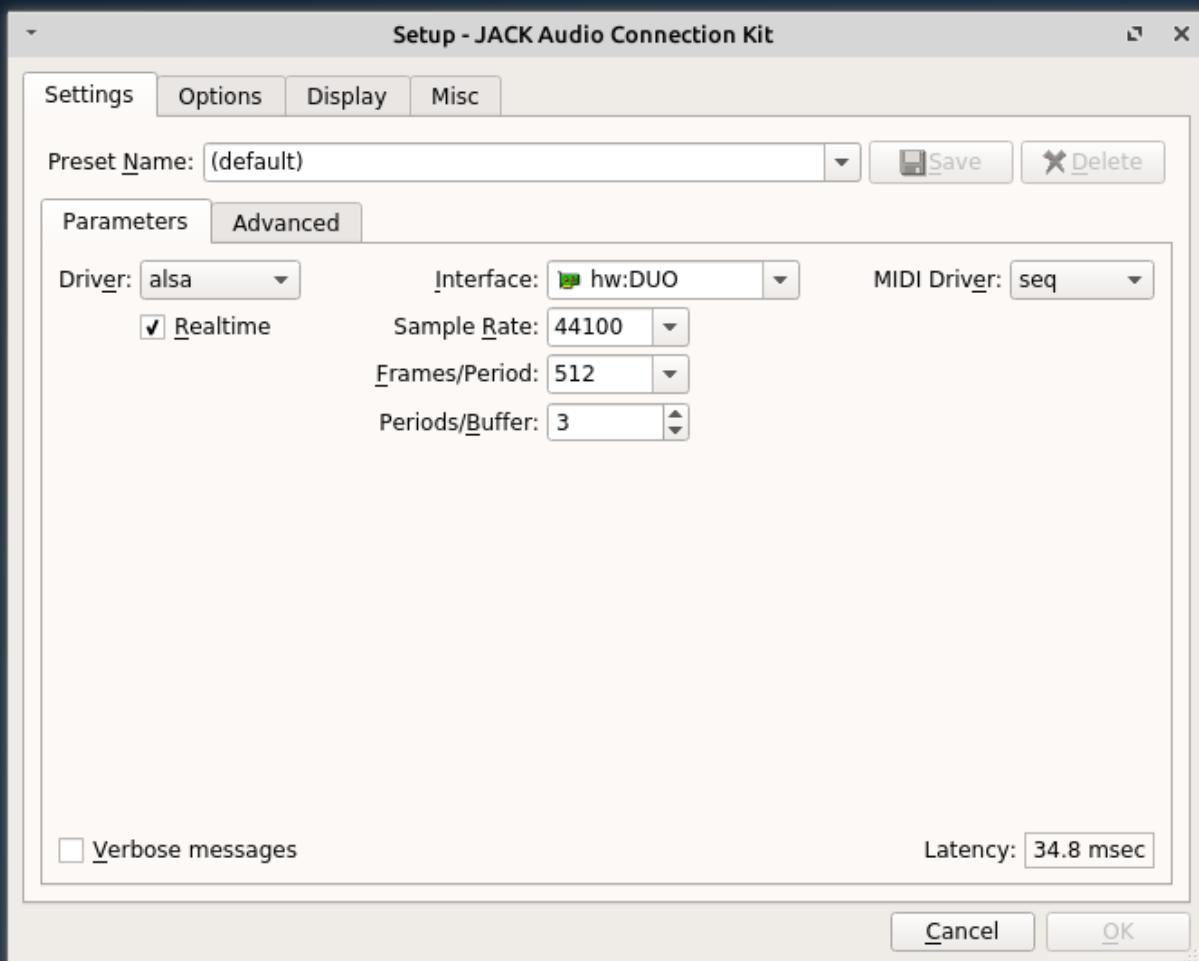
Setting things up with Qjackctl :

The most initially confusing thing for many people getting started with Linux Audio is how to get the applications they want to use to 'see' (or perhaps hear) their Audio device. Reading this section will hopefully facilitate getting everything to 'just work'. The biggest step required of the User is to *identify* and *select* the Audio device they want to use, after that most of the complexity is looked after for you.

The easiest way to setup the JACK Audio server is to use the JACK Control app (a.k.a. Qjackctl). It can easily be launched from the XFCE4 Panel or Whisker Menu. To setup your Audio device click the 'Setup' button.



The 'Setup' Window has multiple tabs which we will go over one at a time, the 'Settings' tab is where we initially select and set up the device we want to use with JACK in the 'Parameters' tab. To the left is the 'Driver' select dropdown box. Internal, PCI, and USB 1.1 and 2.0 Audio devices require the 'alsa' driver and FireWire devices require you to select the 'alsa' or 'firewire' driver (depending on the firewire driver being used). It is essential that the 'Realtime' checkbox is selected. Next we need to select our desired Audio device from the 'Interface' dropdown box.

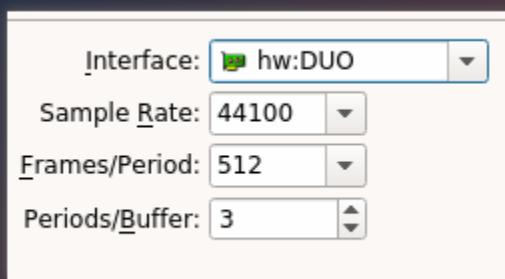


In the 'Interface' dropdown box we will see a list of available Audio devices on the system, generally speaking whatever appears in the dropdown box should be supported to use with JACK, below is an example with multiple ALSA devices on a system to illustrate what to look for.



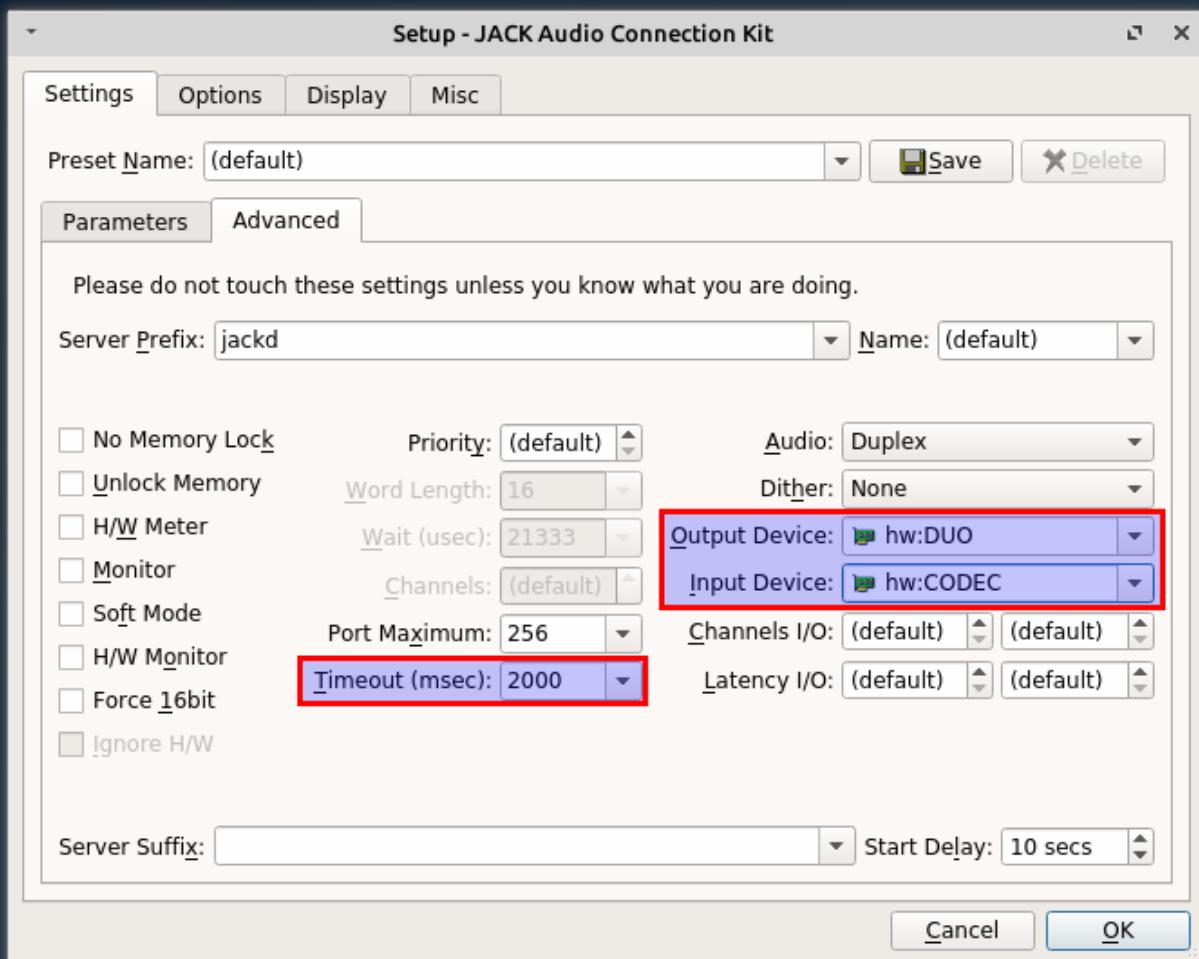
In the example picture above we can see that the system has a 'USB Audio CODEC' device at 'hw:0' and an iRig DUO USB Audio device at 'hw:1'

Underneath the 'Interface' dropdown box we can set the 'Sample Rate' , 'Frames/Period' and 'Periods/Buffer'. By default AVL-MXE is pre-configured for a Sample Rate of 44100, 1024 Frames/Period and 3 Periods/Buffer. Potentially these default settings will work OOTB for most Users but may need to be changed depending on the performance of your hardware. The most influential number is 'Frames/Period'. If you are experiencing Audio dropouts or 'Xruns' you may need to increase from '1024', conversely if you want to have lower 'latency' you can decrease from '1024' until Xruns reappear. The 'Periods/Buffer' figure is dependent on the Audio device type, USB and Internal Audio devices generally perform best at a setting of '3' Buffers and PCI(e) Audio devices require a setting of '2'. Once you are happy with your settings you can start JACK by hitting the 'Start' button on Qjackctl.



***PLEASE NOTE!** - 'Latency' is the delay incurred by your Audio/MIDI signals going into your Audio/MIDI device Inputs and following the signal path of both the hardware and software Applications being used and coming back out of the Audio/MIDI device's Outputs.

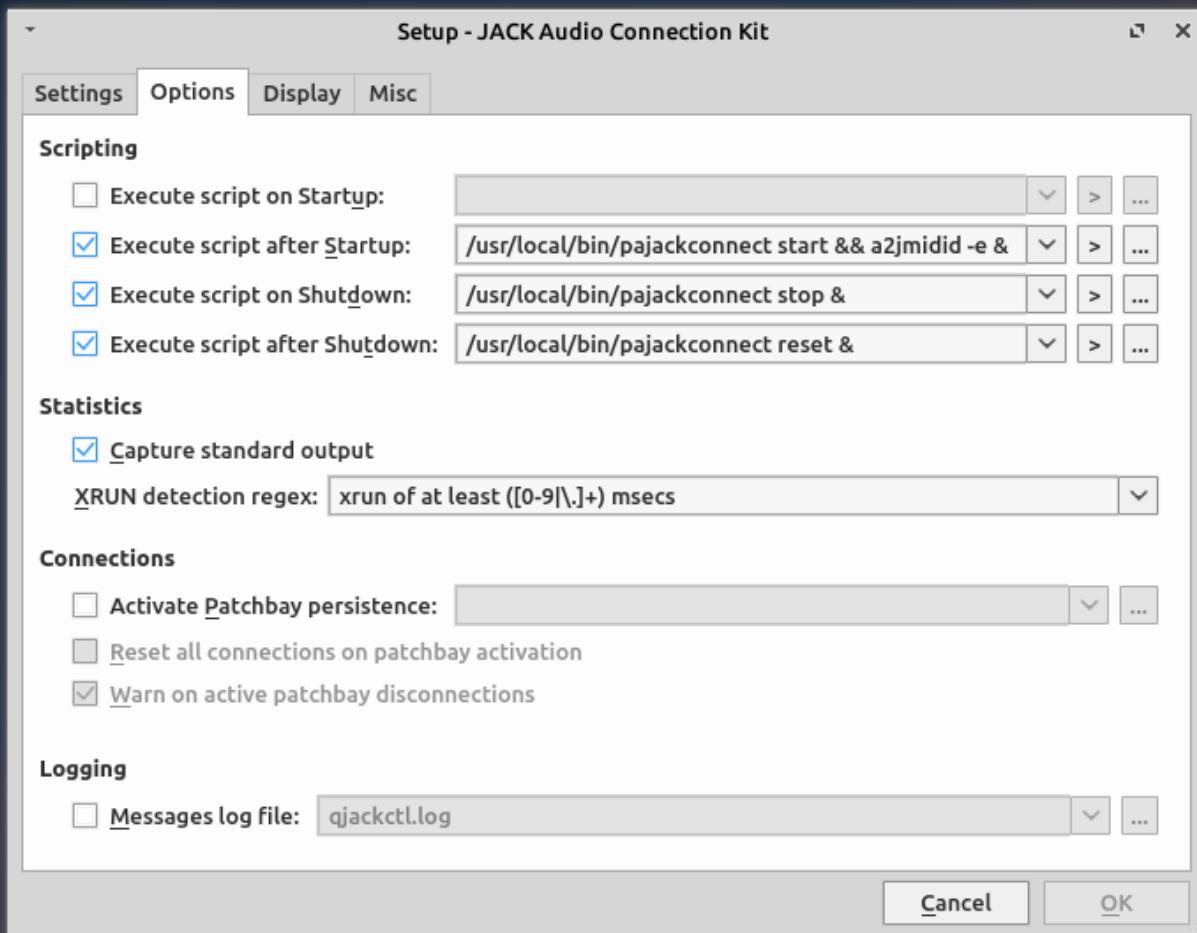
The 'Advanced' tab of the Setup window contains extra settings for experienced Users and in most cases these settings do not need to be changed. The ability to choose separate Output and Input devices may be of interest to people who want to use multiple Audio devices, a common scenario is using USB microphones to record and another System Audio device to play back. Another important JACK setting is the 'Timeout' value for running MIDI Plugins and editing a setting of 2000ms is recommended and set by default.



JACK's L'il Helpers: pajackconnect and a2jmidid :

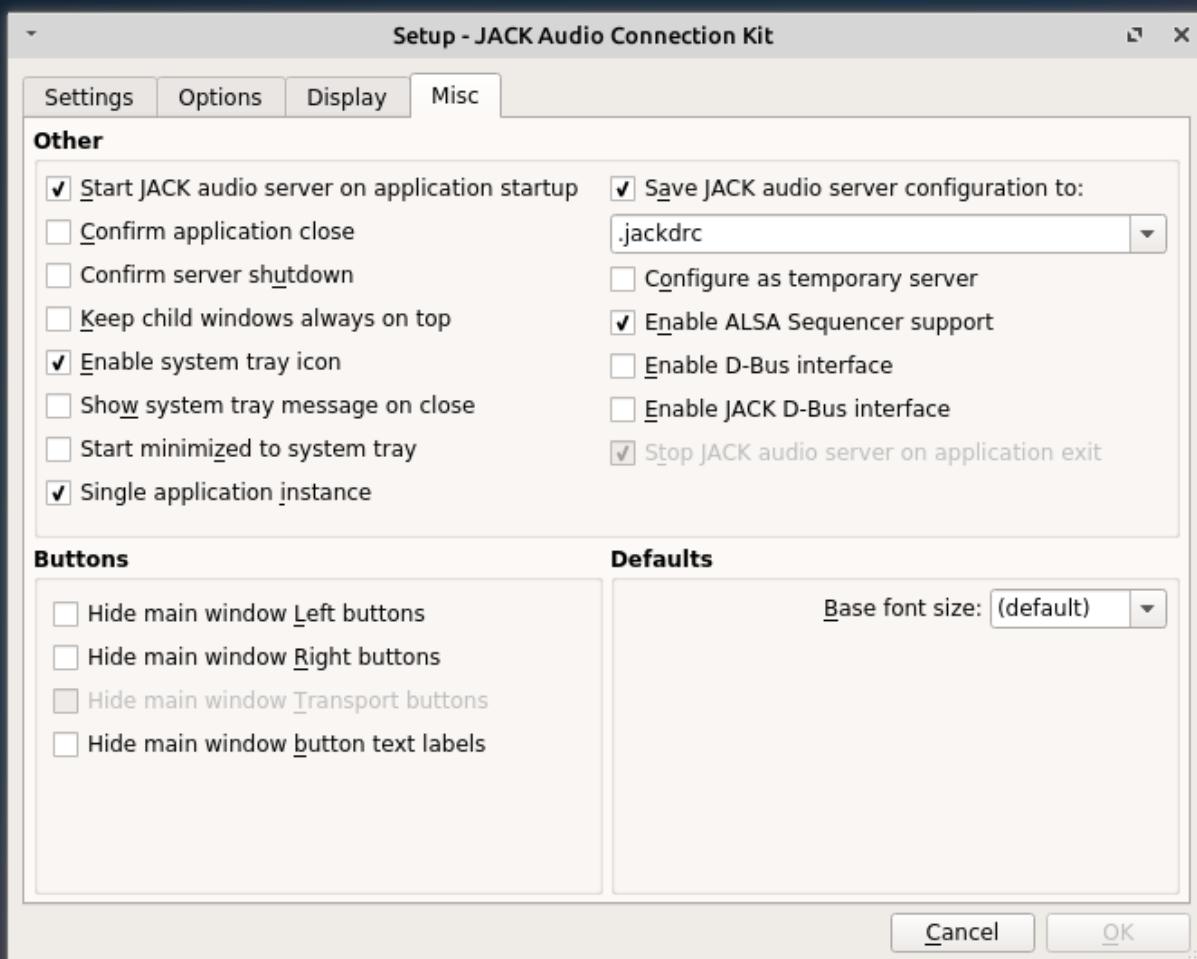
The next 'Options' tab is where we can have Qjackctl execute any optional scripts when it is started, by default AVL-MXE provides a '*pajackconnect*' script as one of two very important helper applications when Qjackctl starts JACK. One helper application is '*a2jmidid*' (ALSA to JACK MIDI Daemon). In Linux there are 2 methods of handling MIDI I/O, one is at the ALSA driver level and the other is with JACK. This can be a problem if you are using an application with JACK and trying to connect a MIDI device with ALSA, the two methods cannot be used at the same time so the solution is to bridge any ALSA MIDI devices so they appear available as JACK MIDI and this is exactly what '*a2jmidid*' is for, now any ALSA MIDI devices seamlessly appear to be selected as '*a2j*' in Applications using JACK.

The second helper application is a script known as '*pajackconnect*' and its purpose is to integrate PulseAudio as a *source* and *sink* server and client routed through JACK. This ensures that applications using PulseAudio will route their Audio to the selected running Audio device you've selected in Qjackctl.



JACK Autostart and Other Settings :

The 'Misc' tab of the Qjackctl 'Setup' window gives some other convenient options for getting JACK integrated smoothly. The '*Start JACK Audio server on application startup*' checkbox means as soon as you launch Qjackctl it will start JACK without you needing to hit the 'Start' button. You can also have Qjackctl run in the system tray on the XFCE4 panel when you close the GUI. If you have a setup where you are using one Audio device and it doesn't change you can have Qjackctl automatically start when you log in to your computer, for this to work you need to select the option to '*Start JACK Audio server on application startup*'. To have Qjackctl automatically start you need to add it to the XFCE4 'Session and Startup' list. Go to the Whisker Menu and type '*Session*' into the search bar and the top choice should be '*Session and Startup*', go to the 'Application Autostart' tab and click the 'Add' button and create an entry for Qjackctl like the one shown on the next page.

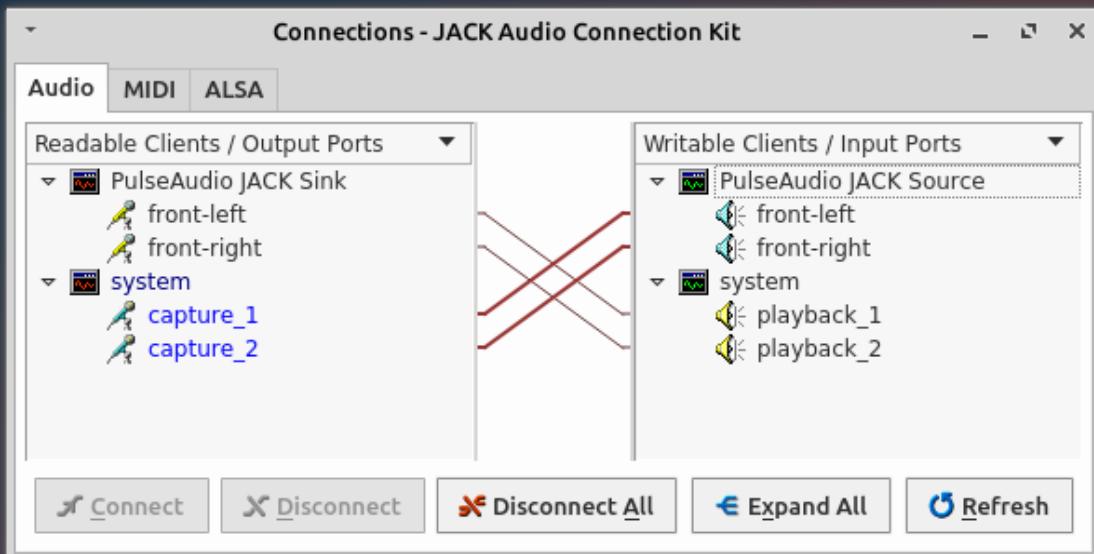


Getting Connected :

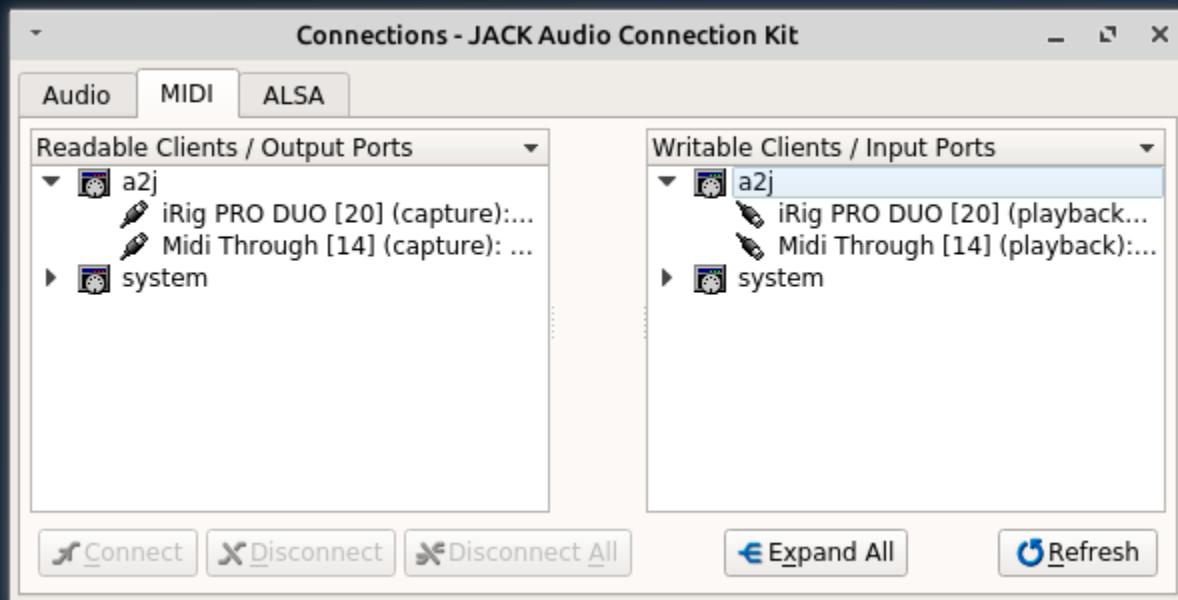
If you are using a monolithic DAW like Ardour or Mixbus with JACK you will find that most of the connectivity you need is managed within the program itself so making manual connections is not necessary, in these cases you simply start JACK and then launch your program. However if you want to use and connect individual JACK-aware programs you may want to use the connection features of Qjackctl. To access the connections window click the 'Connect' button.



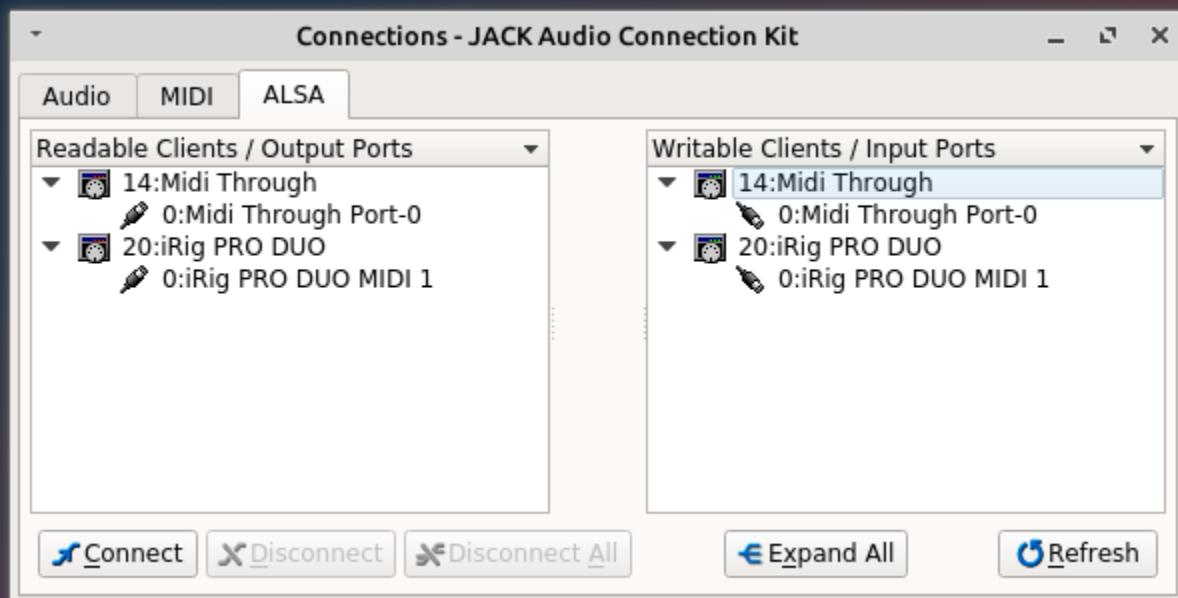
There are three tabs in the 'Connections' window which pertain to three specific connection points. 'Audio' is for both hardware and software Audio (not MIDI) connections. When you are using the default AVL-MXE settings it is normal and expected for the 'PulseAudio JACK Sink' and 'PulseAudio JACK Source' clients to appear because PulseAudio is still active on the system. If you have a JACK-aware application running it should appear in the 'Readable or Writable Clients' panes and in most cases you will want to connect it to the 'system' playback or 'system' capture ports *not* the PulseAudio ports.



The center 'MIDI' tab is for 'JACK MIDI' connections, because JACK MIDI is being bridged by 'a2jmidid' its ports appear under the 'a2j' Client. You can see the iRig PRO DUO is visible in this tab ready to be connected to JACK-aware MIDI applications.

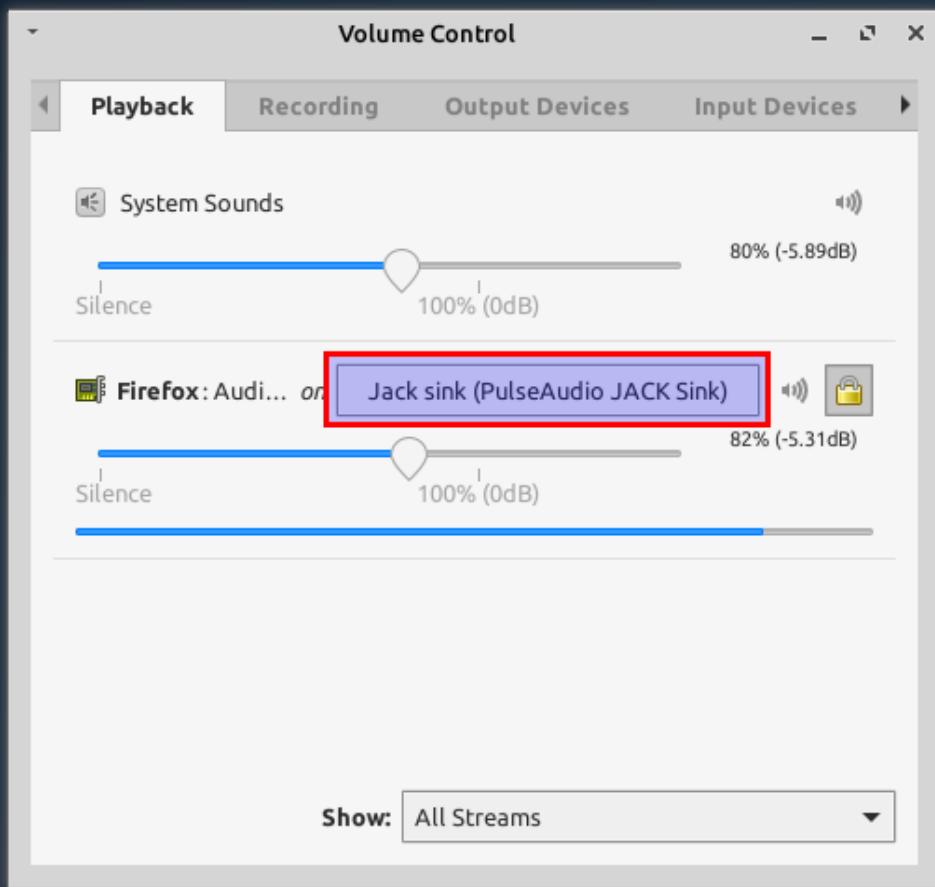


The 'ALSA' tab is for ALSA MIDI hardware and software to be connected, as you can see since the iRig PRO DUO is supported by ALSA it also appears in the ALSA tab.

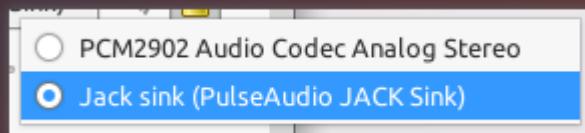


Potential Audio Routing Problems with Mozilla Firefox :

Recently Mozilla's Firefox Web Browser changed its Audio routing from directly connecting to ALSA to utilizing PulseAudio. It has been noted that if PulseAudio and JACK are working together utilizing the 'PulseAudio JACK Source/Sink' Firefox may try to access the default system Audio device directly rather than connecting through the already running PulseAudio JACK Sink. This results in no Browser sound and Audio and Video streams that will not play back. To fix this Firefox needs to be directed to use the JACK Sink by using the PulseAudio Volume Control launched either from Menu→ Multimedia→PulseAudio Volume Comtrol or clicking the PulseAudio applet on the XFCE4 Panel. Look for 'Firefox' in the 'Playback' Tab and click on the highlighted Audio device button.



Make sure you have the 'PulseAudio JACK Sink' selected for Playback rather than the system Audio device.

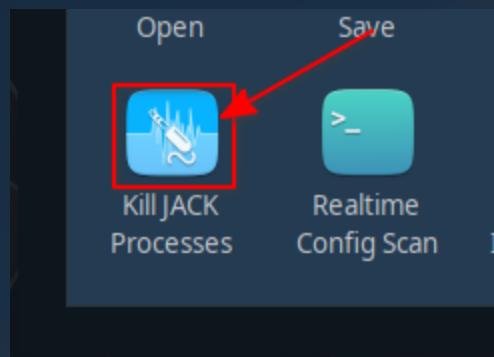


jackd and jackdbus :

There are 2 different ways the system can run JACK behind the scenes, the simplest is by running the 'jackd' executable with 'Qjackctl' and this is what AVL-MXE uses by default, however there is a second way of running JACK and this is with the Desktop interprocess communication mechanism known as 'D-bus'. Running JACK with D-bus is a selectable option in Qjackctl (Setup → Misc. tab). If you start JACK with Qjackctl before doing Audio and MIDI work you will probably not encounter the D-bus JACK executable known as 'jackdbus'.

Occasionally other JACK-aware programs that are launched before Qjackctl may inadvertently start jackdbus and when you later try to launch JACK with Qjackctl you will see JACK is already running and displaying as 'Active' in the Qjackctl GUI and you may even notice that the sample rate may set to a different value than intended. Only one instance of JACK can run at a time so you may not be able to launch other JACK-aware programs or you may find that your Audio setup seems to have a mind of its own if jackdbus is running undetected in the background.

The 'AVL-MXE Assistant' has a utility to help shutdown any instances of jackd and jackdbus running in the background and reset the system to start JACK cleanly.



You can also manually kill all JACK processes by running this command in a regular Terminal:

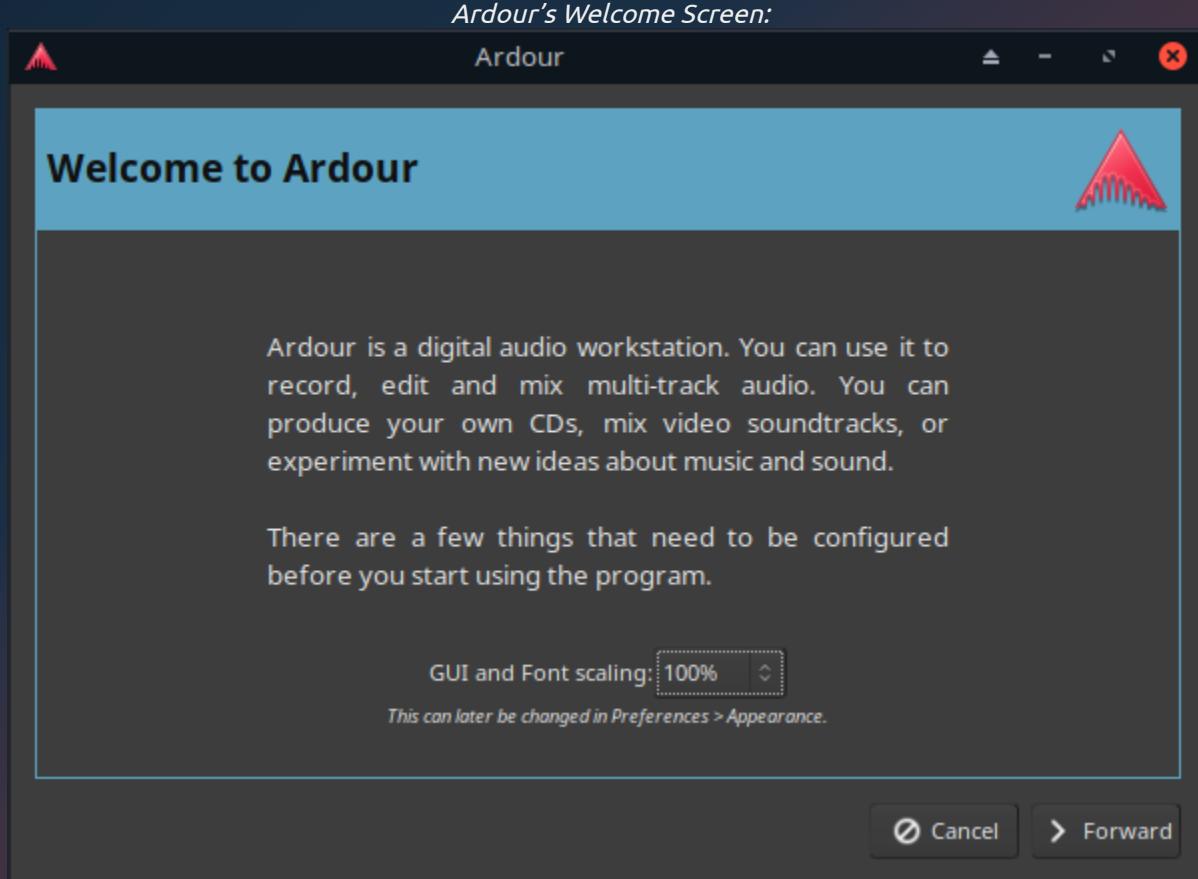
```
killall -9 jackd jackdbus
```

Initial Setup of Ardour and Mixbus 32C

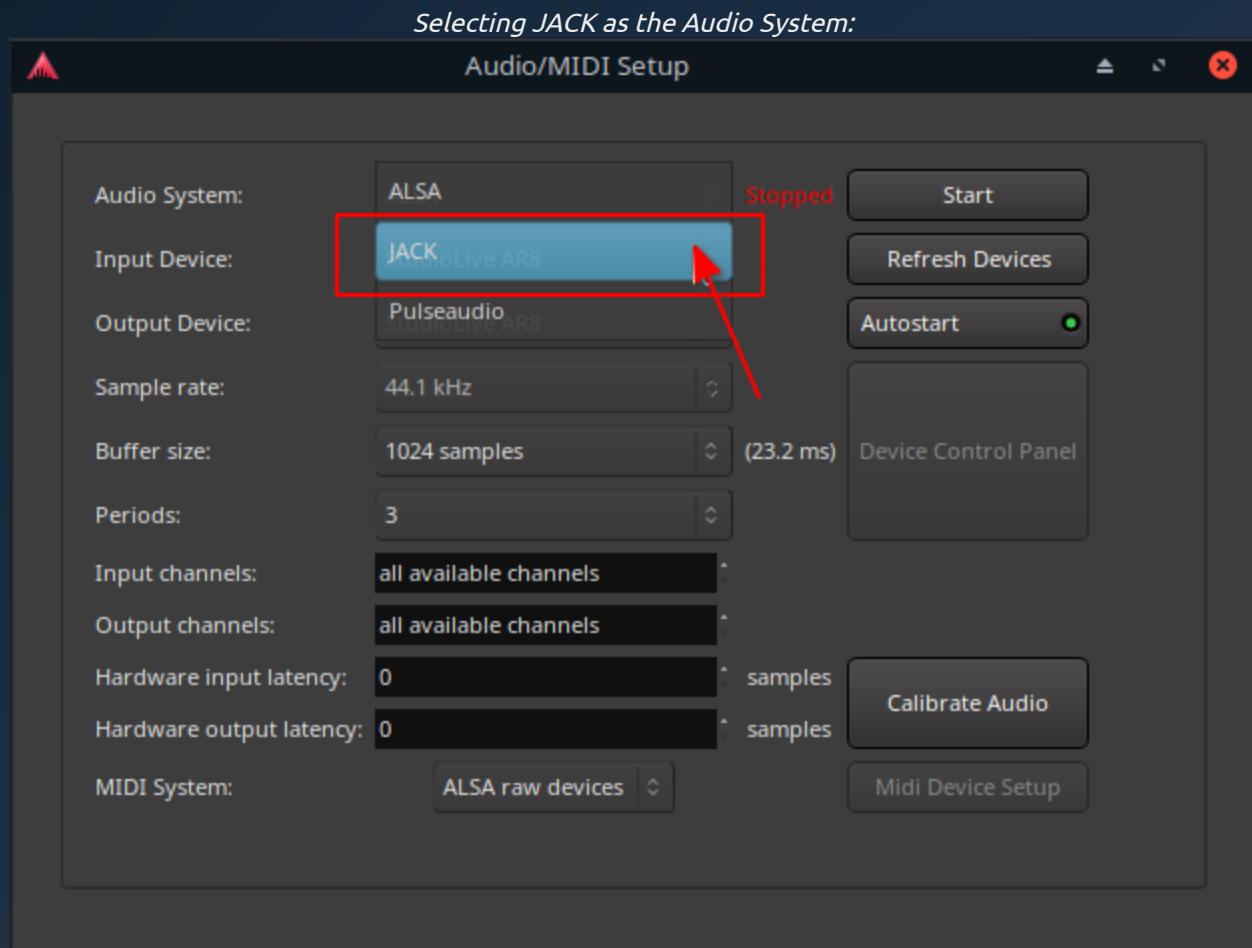
Choosing An Audio Backend :

For many people new to Audio Recording with Linux it can be a bit confusing and daunting to initially set up the full-featured DAWs Ardour and Mixbus. Both powerful programs share the same core so we will use Ardour as an example. Recently Ardour and Mixbus have gained the ability to be used with either JACK, ALSA and most recently PulseAudio 'backends'. If you don't wish to use Ardour or Mixbus with JACK then choose one of the other Audio backends and ignore the JACK specific instructions in this section. As stated in the previous chapter AVL-MXE is centered around using the JACK Audio server, because JACK has interconnectivity with other external JACK-aware programs it is recommended to prefer the JACK backend for using Ardour/Mixbus and to have JACK up and running before Ardour and Mixbus are launched.

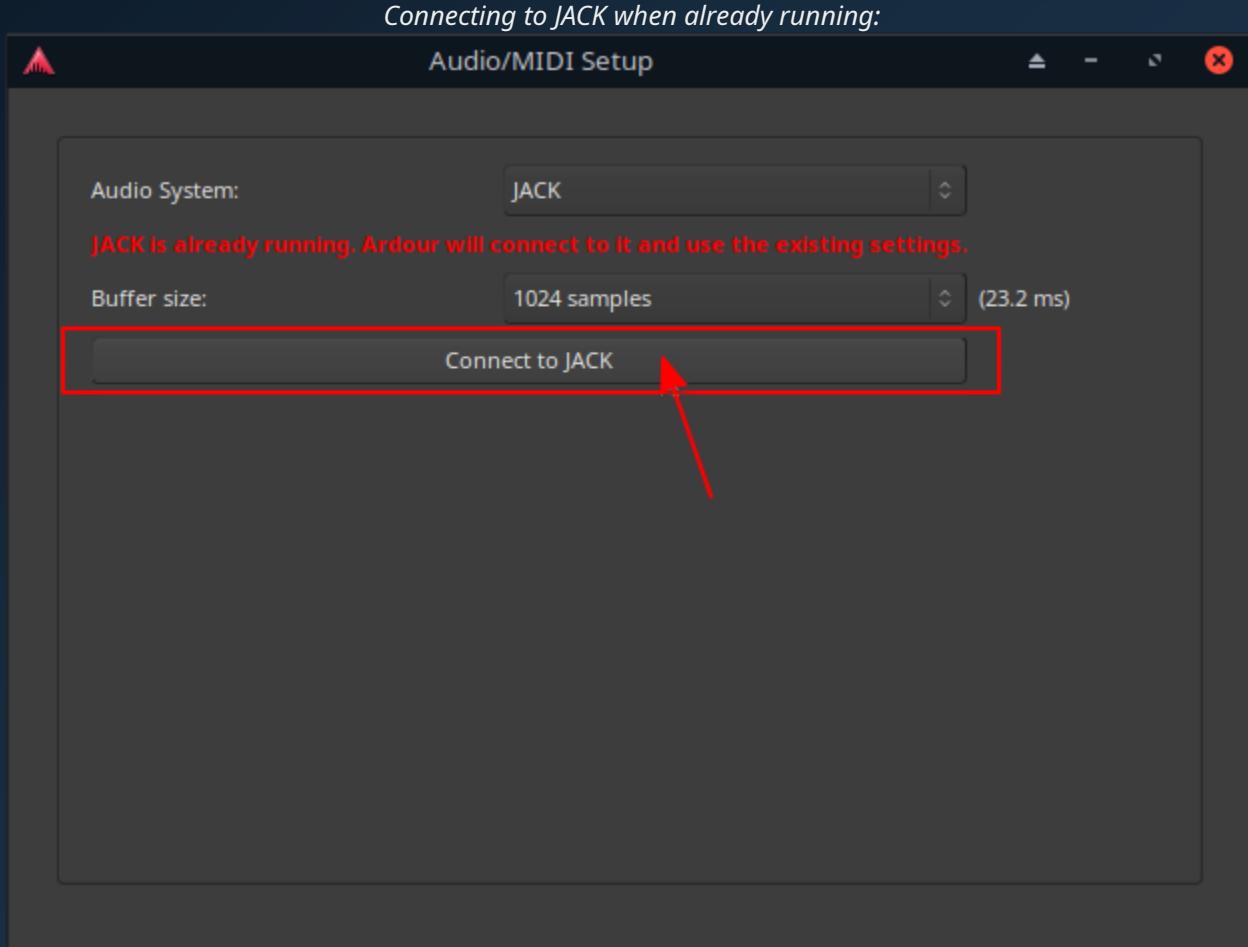
When you initially launch Ardour or Mixbus either in a Live Session or a fresh HDD install you will be greeted by an initial welcome screen and walked through a few setup dialogs, in most cases except for what is detailed here you can accept the default setup choices as offered.



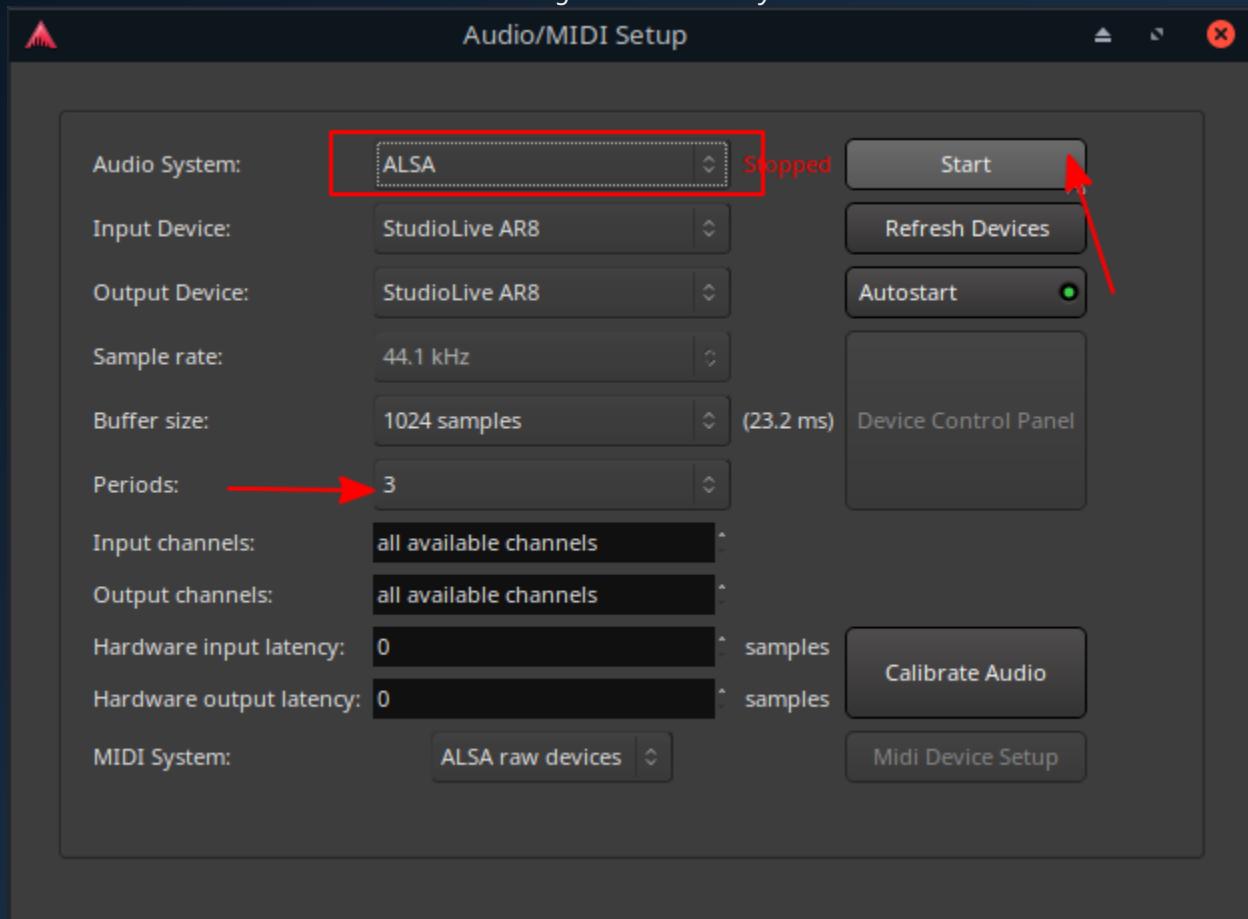
When you set up your initial Ardour/Mixbus session you will be presented with an 'Audio/MIDI Setup' dialog, this is where you will choose whichever Audio backend you will use. As stated JACK is the *preferred* recommendation, ALSA can be selected but has the following caveat: No Audio/MIDI from any other external programs or applications can be routed in or out of Ardour/Mixbus. In the ALSA setup example below we are using the same USB Audio device for both Input and Output which is a suggested practice. It is possible to select a different Input than the Output, people using USB Microphones may want to use the mic with an Input and the System Audio device as an Output. Take note that since this is a USB Audio device we set the 'Periods' dropdown selection to '3', for internal or PCI(e) Audio devices it is recommended to use a setting of '2'. If required it is also possible (but not usually necessary) to accurately calibrate Audio/MIDI latency using a cable looped from your Audio/MIDI outputs to the inputs to calculate the actual latencies. Once you have your settings selected click the 'Start' button to your upper-right.



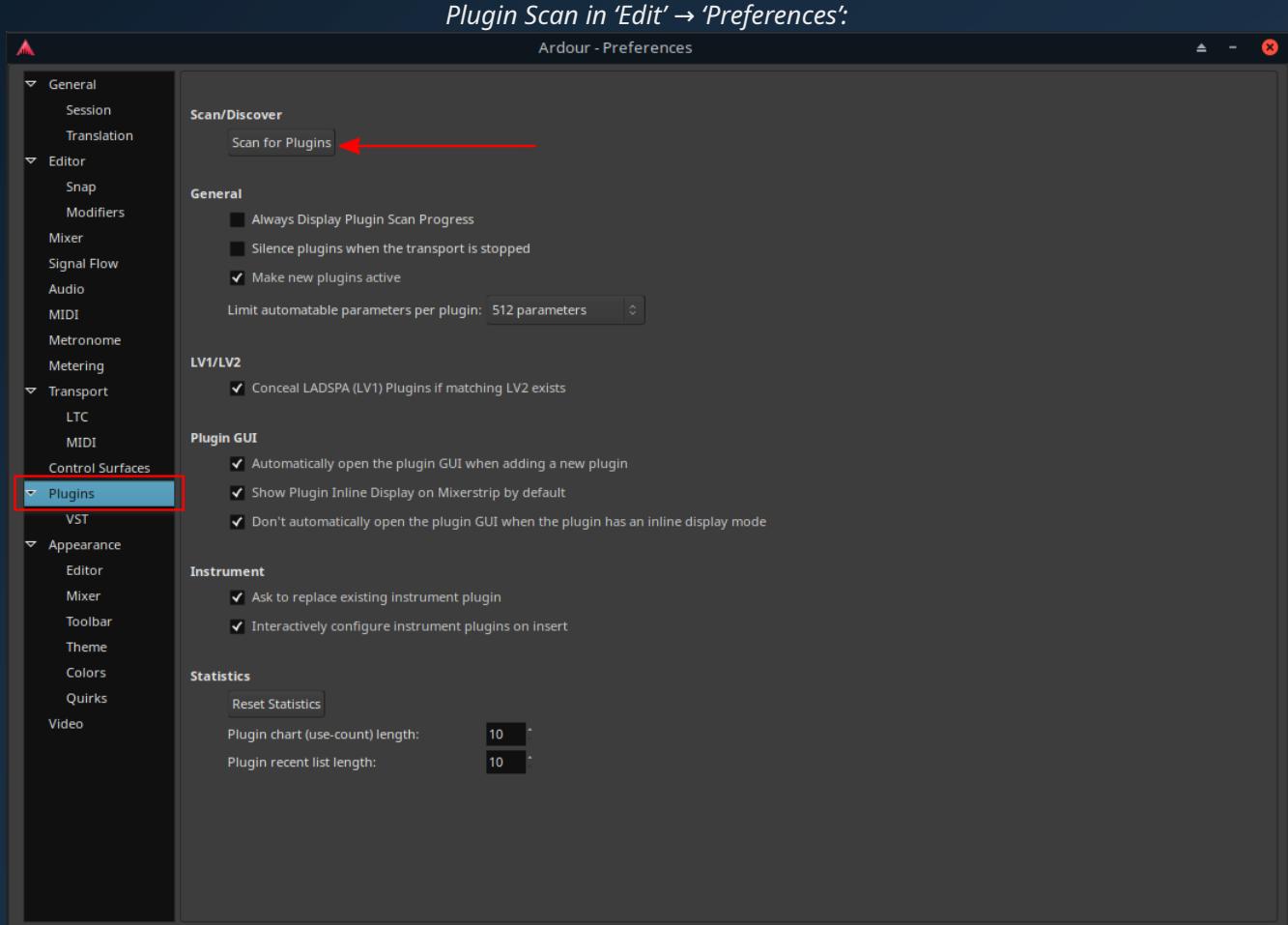
If you already have JACK up and running before setting up your initial Ardour/Mixbus session when you select the 'JACK' backend it will detect that JACK is already running and simply clicking the 'Connect to JACK' button will connect you to whatever settings and Audio device you've chosen in Qjackctl.



Connecting to ALSA Audio System:



A very important part of the initial setup process is to have Ardour scan for Audio plugins, the scan will ensure that the hundreds of pre-installed Audio Plugins included in AVL-MXE will all be present and ready to use in your Ardour sessions. Ardour will auto-detect LADSPA and LV2 Audio plugins, LinuxVST plugins specifically will be found during the scan. It is important to note that any LinuxVST plugins you may have added yourself that aren't located in the standard paths of /usr/local/lib/vst or /usr/lib/vst will require you to point Ardour/Mixbus to their folders by adding a custom path after launching the program, custom paths are set in 'Edit' → 'Preferences' → 'Plugins'.

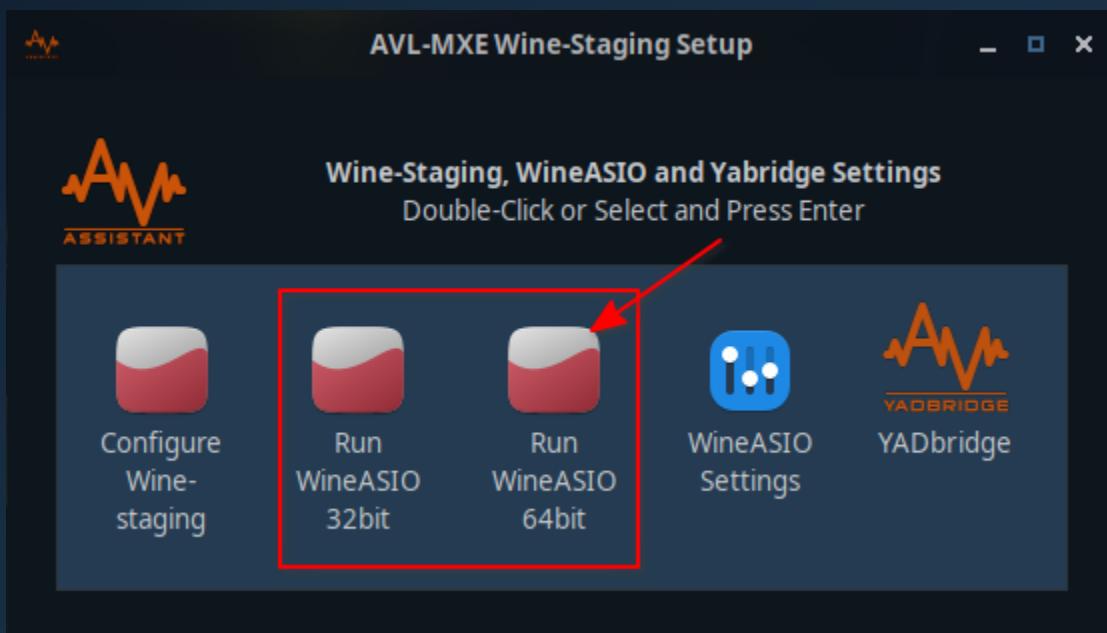


Further Info: [The Ardour Reference Manual](#)

Running Windows Audio Software

Exposing Windows ASIO Audio drivers to JACK with WineASIO :

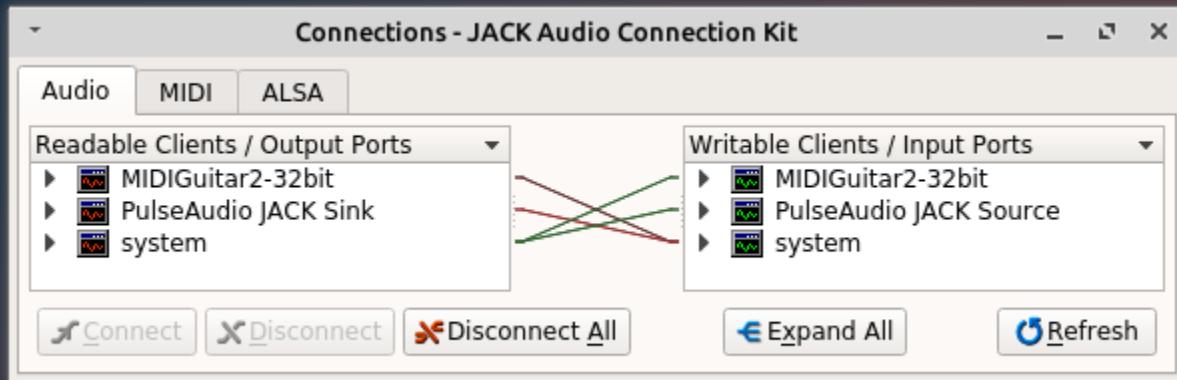
It is possible to run some popular Windows Audio software in AVL-MXE and have it run with Windows native ASIO Audio drivers. This is accomplished using a utility called WineASIO which runs in the Microsoft Windows emulating Linux software known as 'Wine'. AVL-MXE comes with 'Wine-staging' installed and a WineASIO launching utility that can be found in the 'Wine-Staging Setup' Dialog of the AVL-MXE Assistant. To use WineASIO you must first have the JACK Audio server running as detailed in the '[Audio & Midi in AVL-MXE](#)' section. WineASIO can be used with both 32bit and 64bit Windows software applications.



Once you select the appropriate button pertaining to the Windows Audio application you want to run a Terminal window will open confirming that the WineASIO '.dll' has successfully launched. The terminal window output will be very verbose with several warnings specific to Wine-Staging. The main thing to watch for is that the DLL has launched successfully so look for this line or something similar at the bottom of the Terminal:

```
regsvr32: Successfully registered DLL 'wineasio.dll'
```

When WineASIO is up and running you can launch your Windows program and select the WineASIO driver in the Audio setup dialog of the application. As an example we will use the standalone version of the Windows software 'Jam Origin MIDI Guitar 2'. As you can see in the highlighted section the WineASIO driver has been selected which will route the Audio output through JACK.



Running Windows VST Plugins in Ardour/Mixbus :

Steinberg's 'Virtual Studio Technology' or 'VST' Audio plugin format is a cross-platform industry standard and there are a wealth of good quality commercial and free VST plugins available. VST's for Windows are by far the most common and a large number of them will run on Linux with the aid of 'Wine' Windows emulating software and various bridging applications. As you would expect running Audio plugins that are not native to Linux presents challenges and *it can not be overstated that there are no guarantees of functionality and stability, it is always preferable to run Audio plugins that were intended for the Linux platform.*

AVL-MXE comes with a small set of 'oldie-but-goodie' free 32bit Windows VST Audio plugins placed in the '/usr/local/lib/vst' folder of the filesystem. With 'wine-staging' and 'Yabridge' pre-installed these plugins are ready to be hosted and used in Ardour and Mixbus32!

Hosting Windows VST's with 'Yabridge' :

The AVL-MXE method of hosting Windows VST plugins is the use of 'Yabridge' VST bridge, Yabridge works by wrapping the Windows '.dll' filetype and making it appear to the plugin host as a native Linux VST '.so' file. Follow the instructions below for any new Windows VST's you may add.

In AVL-MXE Windows VST's are placed in /usr/local/lib/vst and /usr/lib/vst is for native for Linux VST's only... The rationale for this is that most DAWs will look along both of those paths automatically for VST plugins of any kind and we want VST's to be available system-wide for all Users not buried in the .wine folder in the Users home folder. Yabridge creates the special '.so' file in its 'Sync' function which is used as a reference file to convert all your Windows VST '.dll' files to Linux-friendly '.so' files. It is the '.so' files which Yabridge will use for the Windows VST Plugins to appear in the usual list of plugins in Ardour/Mixbus.

AVL-MXE comes with a little graphical utility called 'YADbridge' to set up and organize all your .dll files and do the conversion. YADbridge also facilitates removing dll directories, checking the status of your converted dlls and manually editing it's configuration file manually in the rare case it becomes necessary.



X Yabridge Status of VST Folder(s)...

```
yabridge path: <auto>
libyabridge-vst2.so: '/home/isotester/.local/share/yabridge/libyabridge-vst2.so'
libyabridge-vst3.so: '/home/isotester/.local/share/yabridge/libyabridge-vst3.so'
(64-bit)

yabridge-host.exe: '/home/isotester/.local/share/yabridge/yabridge-host.exe'
yabridge-host-32.exe: '/home/isotester/.local/share/yabridge/yabridge-host-32.exe'

installation method: copy

/usr/local/lib/vst/
    ANWIDASoft/ANWIDA Soft DX Reverb Light.dll :: VST2, 32-bit, copy
    Camelcrusher/CamelCrusher.dll :: VST2, 32-bit, copy
    Charlatan/Charlatan.dll :: VST2, 32-bit, copy
    DSK/DSK Ethereal PadZ.dll :: VST2, 32-bit, copy
    DSK/DSK Overture.dll :: VST2, 32-bit, copy
    DSK/DSK SF2.dll :: VST2, 32-bit, copy
    ESP/ESP_FreeeeQ.dll :: VST2, 32-bit, copy
    ESP/ESP_UniQ.dll :: VST2, 32-bit, copy
    EasyCrumar/easy-crumar-MS.dll :: VST2, 32-bit, copy
    FabFilter/FabFilter Pro-L 2.dll :: VST2, 32-bit, copy
    FabFilter/FabFilter Pro-Q 3.dll :: VST2, 32-bit, copy
    GSI/WB3.dll :: VST2, 32-bit, copy
    GSI/WB3fx.dll :: VST2, 32-bit, copy
    IgniteAmps/Emissary.dll :: VST2, 32-bit, copy
    IgniteAmps/SHB-1.dll :: VST2, 32-bit, copy
    Kjaerhus/ClassicAuto-Filter.dll :: VST2, 32-bit, copy
    Kjaerhus/ClassicChorus.dll :: VST2, 32-bit, copy
    Kjaerhus/ClassicCompressor.dll :: VST2, 32-bit, copy
    Kjaerhus/ClassicDelay.dll :: VST2, 32-bit, copy
    Kjaerhus/ClassicEQ.dll :: VST2, 32-bit, copy
    Kjaerhus/ClassicFlanger.dll :: VST2, 32-bit, copy
    Kjaerhus/ClassicMasterLimiter.dll :: VST2, 32-bit, copy
    Kjaerhus/ClassicPhaser.dll :: VST2, 32-bit, copy
    Klanghelm/DC1A2.dll :: VST2, 32-bit, copy
    Klanghelm/IVG12.dll :: VST2, 32-bit, copy
    Klanghelm/MJUCjr.dll :: VST2, 32-bit, copy
```

Open Ardour/Mixbus and go to Edit--->Preferences--->Plugins--->VST and make sure the path to your Windows VST folder is there (this is why I use /usr/local/lib/vst because Mixbus looks there automatically). If your particular path isn't there you'll need to add it manually so Ardour/Mixbus knows where to scan.. Once that is done run a plugin scan and if all goes well those converted '.so' files should be picked up and appear as selectable in the Ardour/Mixbus plugin list. The image below shows the 'Slick EQ' Windows VST Plugin running with Yabridge in Ardour.



PLEASE NOTE The first time Wine-Staging runs on your computer it needs to configure itself and set up its directories in a hidden '.wine' directory in your User home folder. Wine-Staging does not run its setup until the first time it is called to run a Windows program by the host system. If the first time Wine is initiated is during a plugin scan within a DAW it is possible the initial Wine configuration setup will occur as you are running the VST plugin scanner which ensures the renamed '.so' files of the Windows VST Plugins show up in your Linux DAW. On some systems this may cause the scan to time out, fail and not completely finish. Once Wine has set up its initial config you should be able to run the plugin scan again in your DAW and it should be able to successfully complete.

Saving and Restoring JACK Connections

AJ-Snapshot for JACK-Aware External Application Connections :

If you are using several JACK-Aware applications and connecting them together those unique individual JACK connections will be lost when you shut down the individual applications. This is a much different scenario than when you are working within a DAW like Ardour, in that case the DAW remembers the connections when you save your session. What if you want to use external JACK programs in various combinations with a complicated routing? You certainly don't want to have to remember and manually re-connect every time you use those uniquely arranged programs.

A very simple and effective tool to save and restore ALSA and JACK Routing is a little CLI utility called aj-snapshot. In AVL-MXE I have added a very simple UI to make using aj-snapshot a little easier. With JACK running and your applications connected simply launch 'AJ-Snapshot-UI' from the Whisker Menu, it can be found in Multimedia → AJ-Snapshot-UI. Select the option to 'Save an aj-snapshot' and it will prompt you to enter the name of the snapshot and present a File Manager to save it where you want. Later, with JACK running and the applications you wish to re-connect launched and running simply open your saved Snapshot and the routing and connections will be restored.



Commercial Software Demos in AVL-MXE

An Introduction to Some New Options :

AVL-MXE is somewhat unique compared to its contemporaries in that it offers the choice of trying Commercial software demos within the base Operating System. Over the years good relationships with Vendors have been formed and have provided the opportunity to showcase both Open-Source excellence and innovative Commercial Audio choices. Currently AVL-MXE provides the Demo version of **Harrison Mixbus 32C** which brings the authentic sound of the famous Harrison hardware consoles into a powerful DAW based on Ardour. Harrison Mixbus 32C also comes with all of Harrison's Audio Plugin Demos (both XT and AVA) for you to evaluate as well. Brand new for the first release of AVL-MXE is the addition of the very popular **Reaper** DAW. Reaper for Windows has been known to Linux users for years for its functionality with Wine but in the past couple of years the project has been producing 'native' Linux builds. On top of these DAWs (and also new to AVL-MXE 2020) are additional versatile Audio plugin choices with Demos provided by **Applied Computer Music Technologies**, **Auburn Sounds**, **Cut Through Recordings** and **OvertoneDSP** (64bit AVL-MXE only).

Commercial Demo Plugins running in Harrison Mixbus32C:



Purchasing and Registering the Demos :

All of the Commercial Demos are limited in some way and are not fully functional until they are purchased and registered. The limitation methods vary from not all options being available, dropouts in sound, generated noises at certain intervals etc. Each Vendor has their own method of handling purchasing and registration as well. Harrison, Applied Music Technologies and OvertoneDSP all provide a license text file upon purchase that can be copied and pasted in the Users home folder. Once this text file is present the software will find it the next time it is launched and will appear as registered and be fully functional. With this method there is no need to remove the demos and replace them you can leave them installed as they are and merely add the text key.

In the case of Reaper you can evaluate it unlicensed for 60 days after which time you are expected to purchase a license and register it, the version that comes on the AVL-MXE ISO has its binary folder wrapped in a Debian Package so it can be uninstalled with usual Debian Package tools, if you wish to use a newer version of Reaper simply uninstall the 'reaper6-linux' package with Synaptic and download and extract the usual Reaper Linux bundle wherever you like and use it from there.

Audio Assault, Auburn Sounds and Cut Through Recordings employ a different system where the User is given a special private download URL in an email once purchase and payment are completed and the User then downloads replacement Plugins which are fully functional. For this method it is necessary for the User to manually remove the installed demos using either Synaptic Package Manager or 'dpkg' command (shown below) and then install the replacement plugins as instructed by the individual Vendor.

To remove Audio Assault, Auburn Sounds or Cut Through Recordings Demo versions from your system you can either search for 'audioassault-demo-plugins', 'auburnsounds-demo-plugins' or 'ctrecordings-demo-plugins' in Synaptic, then remove them or alternately open a Root Terminal and issue this command:

```
dpkg -r auburnsounds-demo-plugins
```

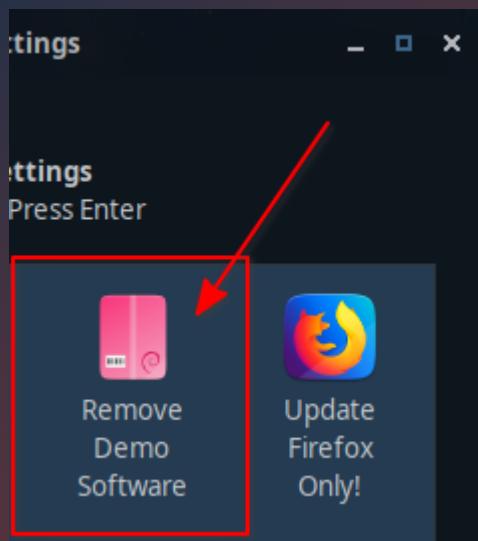
```
dpkg -r ctrecordings-demo-plugins
```

Detailed information on installing the replacement Plugins can be found at the Vendors respective websites linked at the beginning of this section. In most cases it is suggested to create a '.vst' folder in the User's home folder for Linux VST2 Plugins and a '.lv2' folder for LV2 Plugins and copy your Registered versions there.

**NOTE – Both Ardour and Harrison Mixbus (the default DAW Applications included in AVL-MXE) do not automatically search for Plugins in the User's home so these folders must be added to the Plugin search paths manually in Ardour and Mixbus. Plugin search paths can be edited in the 'Edit'→'Preferences'→'Plugins'→'VST' menus of both Ardour and Mixbus.*

Hey Wait! Not For Me! :

With the knowledge that not everyone necessarily wants to have or purchase Commercial Software and that some may prefer for ideological reasons to use only Open-Source applications you can of course choose remove the Commercial Demos from AVL-MXE. For this reason removing all of the Audio Demoware from AVL-MXE is as easy as a one-button click in the AVL-MXE Assistant!



Thanks and Acknowledgements

The Debian/GNU Linux Distribution

MX Linux Developers – dolphin_oracle, fehlix, Adrian, Stevo, SwampRabbit

antiX ISO Build Script – anticapitalista, BitJam

Trulan Martin – RT Kernels

Damentz – Liquorix Kernels

Anahata, Autostatic, linuxaudio.org – Torrents and Mirrors

Korakios – CPU governing utility

Tony Brijeski – Yad/Zenity Scripts

RockHopper – H2 and SFZ Thunar Custom Actions

Yann Buchau – Thunar Image Custom Actions

XFCE4 Team

Filipe Coelho – KX Studio Repositories

Paul Davis, Robin Gareus, Ben Loftis – Ardour/Mixbus

osxmidi – LinVST, SFZero

Robert Vander Helm - Yabridge

sysrqr – BPM Converter

aciddose – Xhip synth and plugins

User Manual Proofreaders

Special Thanks:

All financial supporters of AVL-MXE, your kind generosity and encouragement is very much appreciated !