Using The Custom-Builder (adrive 101)

The following pertains to Carolina, 'lina-lite, and Saluki. At present these are the only Puppy Linux distros that use an adrive and the custom-builder. Any discussion of the custom-builder/adrive would be incomplete without giving credit to jemimah, both of which were her creation.

The adrive has caused a lot of confusion for both those familiar with Puppy Linux and those that are not. In its simplest form, the adrive is nothing more than an sfs file that (usually) contains multiple applications, xfce settings, and personal settings that is automatically loaded at startup. Any settings that are included on the adrive will be loaded 'over-top' of the settings on the main Puppy sfs file. In this way the default xfce settings may be overridden to personal taste, as can wallpaper, etc. This is a very powerful mechanism. Even though these kinds of things can be accomplished with a save-file, doing so at the adrive level allows these changes to be persistent without eating up precious save-file space. It also allows personal configurations to be persistent in the event that you choose to not use a save-file. And it allows for a more concise way of changing the default applications that are included in the distro. Do you prefer Abiword instead of LibreOffice? Not a problem. You can use the custom-builder to remove LibreOffice and add Abiword in one fell swoop. But this isn't the only thing that the custom-builder is capable of. With the custom-builder you can change out the splash-screen, swap out the kernel, change the name of the distro, or create a personalized ISO. You also are given the option of cutting down the zdrive, which holds the driver files. Cutting the zdrive down will create a zdrive that is specific to your setup; it'll only include driver modules that are needed for your specific hardware, thus significantly shrinking the zdrive sfs file. This will shorten boot-up time and make your customized ISO file that much smaller.

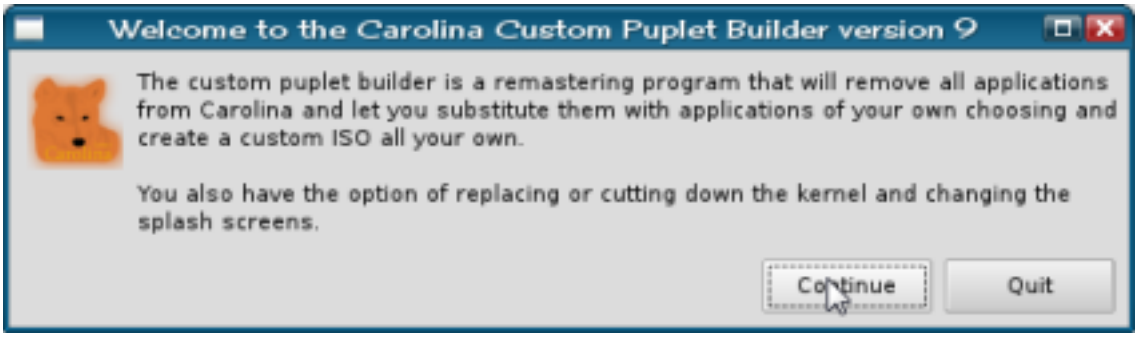
Conceptually the adrive is a revolutionary way of thinking in terms of Puppy Linux. It certainly is not widely accepted, let alone understood, and therefore will more than likely never be included in an "official" Puppy Linux release. However it is quite useful, and those that choose to use Carolina or Saluki while removing the adrive miss the point. Outside of xfce, the adrive is what make these distros what they are. Without it they are nothing more than a traditional Puppy with an xfce window manager. With it they are a powerful piece of software that expands the possibilities for personal customization; the custom-builder also allows for a simplified path to creating a puplet, or even the first steps towards forking the distro.

This guide is not meant to be a thorough, definitive guide to using the custom-builder. It's meant to be more of an introduction. It will walk you through the basics of creating a custom adrive using the custom-builder. To have a thorough understanding of the custom-builder one must use it. Because there is no documentation at present. Experience is currently the best way to learn the custom-builder.

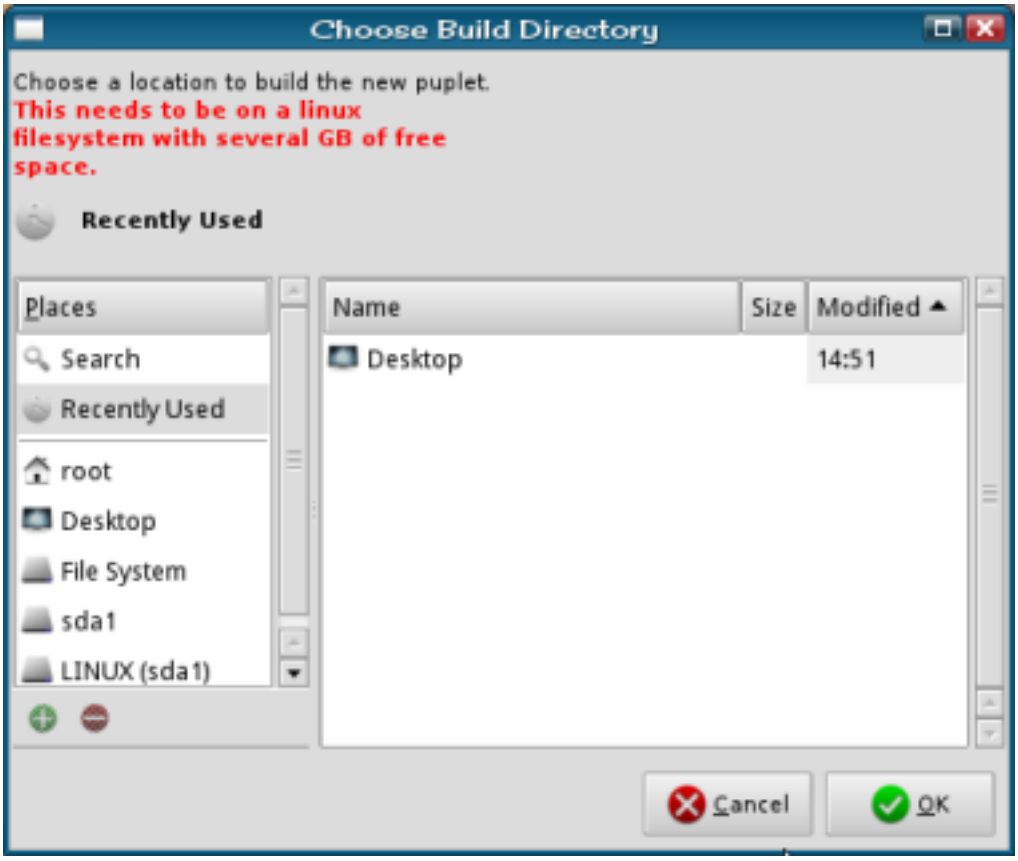
Chapter 1: Getting Started

The first thing you'll need is plenty of space. The build ideally should be done on a hard drive that uses a linux file system, although a large usb should work if it has sufficient free space. How much free space? That depends on how big your adrive will be, and that is dependent on the amount of packages that you'll be installing. But a good rule of thumb would be 1 GB at minimum. You may be able to get by with less. Also, running the custom-builder is best done from a fresh ram-boot (pfix=ram).

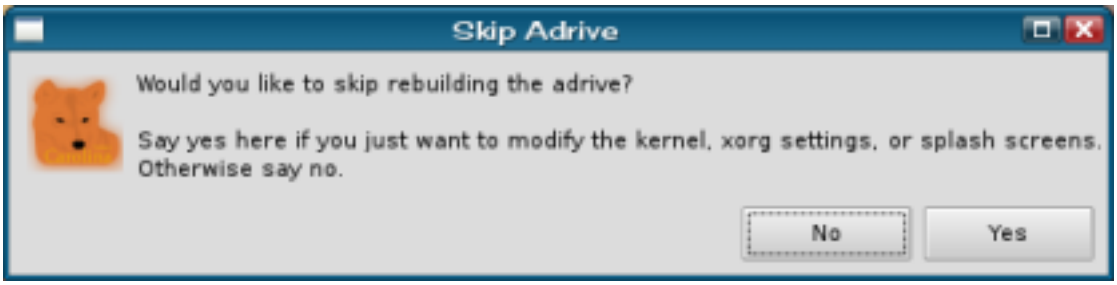
The custom-builder is located in the main menu under System --> Carolina Custom Builder. Go ahead and fire it up. This is what you'll see:



Go ahead an press the 'Continue' button to continue. You'll then see the next screen:

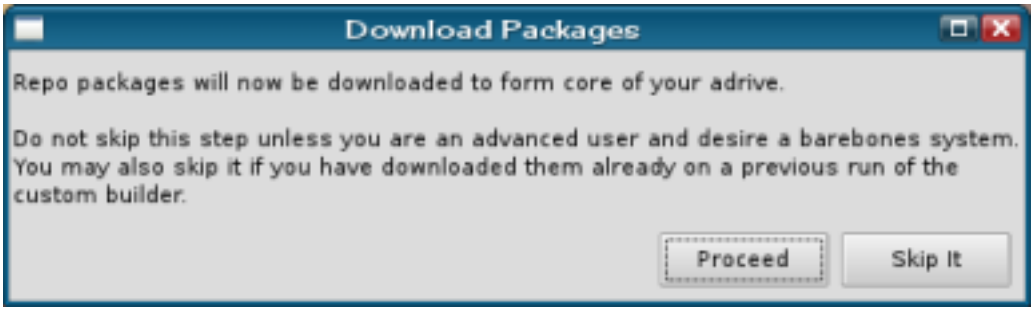


Navigate to a location from where you'll create the build. Create a new directory (preferably) in a location that has a lot of free storage space. I've chosen to do my build on the hard drive (sda1) with a directory named 'new-one'. Click the 'OK' button to continue, and you'll come to the next image:



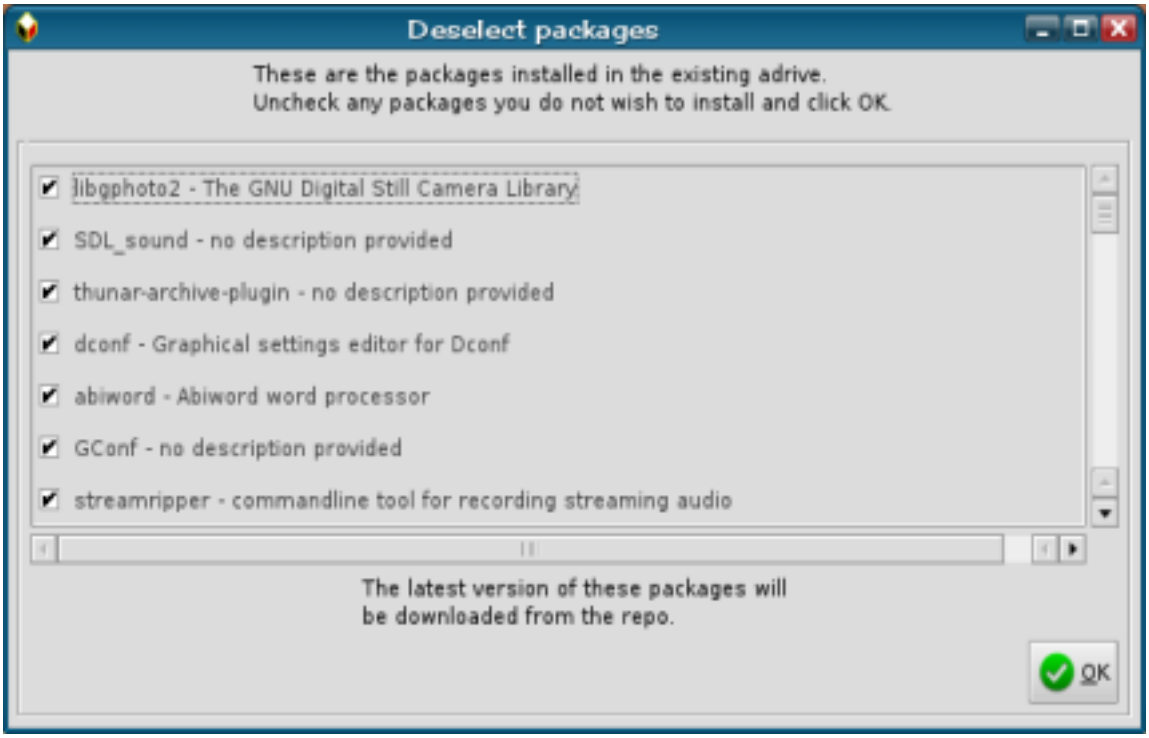
Because the goal of this tutorial is to simply build a customized adrive, we'll choose the 'No' option. But keep in mind that the custom-builder has other capabilities - in reality the custom-builder is very much a 'flow-chart' model, with many branches that allow you to do many different tasks.

The following image is the one that will probably provide the most confusion to those unfamiliar with the custom-builder, as both options can be useful. First, here's the image:



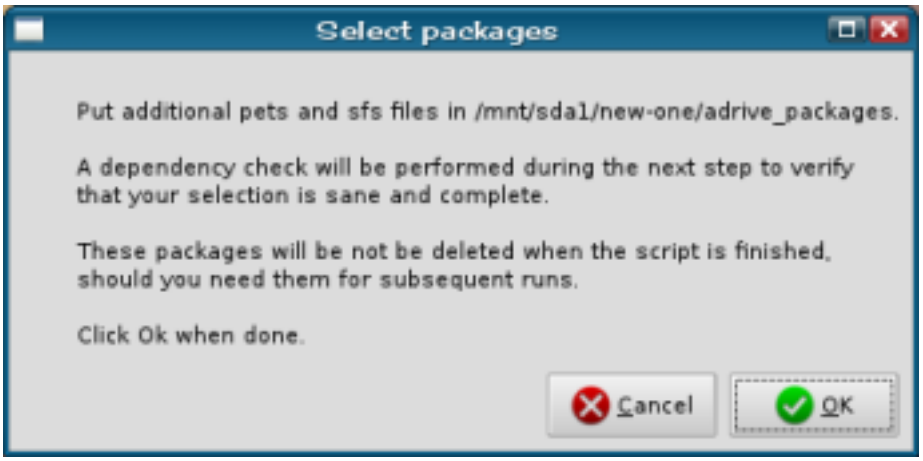
If you choose ‘Proceed’ the custom-builder will download the packages that exist on the ***currently loaded adrive from the default repos*** (please note that some of the packages will not get downloaded due to them being sfs files and/or bugs in the custom-builder. The custom-builder is at present a work in progress. Missing files can be found in the smokey01 repository at http://smokey01.com/carolina/, in either the 'pet_packages-carolina’ or the 'sfs’ folder). This is probably a good option if this is the first time using the custom-builder. Once the packages are downloaded you can then prune the ones you wish to remove, and add other packages (pet and sfs files). The ‘Skip It’ option is best used if you already have a folder already containing the packages you wish to use saved locally. Note that any local pet/sfs files used to create a new adrive will not be altered in anyway, and will therefore be available for future use with the custom-builder. If you have the free space this will save you a lot of download time.

If you choose ‘Proceed’, you’ll be greeted by the following dialog:



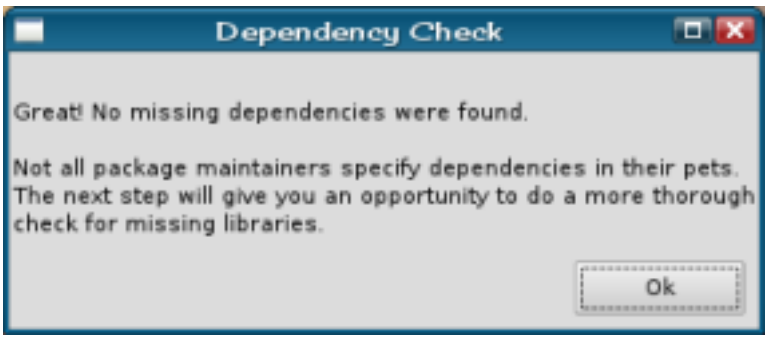
In this dialog, you may ‘un-check’ any undesirable package and click ‘OK’ to continue. The custom-builder will then download the packages that are ‘checked’. If it complains that it can’t download a certain package, you may need to hunt for it in the repository and download it manually.

The next dialog that appears it this:

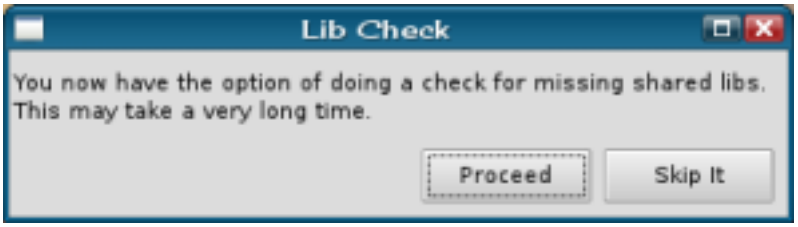


A Thunar window will open with a folder for the adrive packages (in my case / mnt/sda1/new-on3/adrive_packages). It is here that you can prune/add pets and sfs files. Don’t worry about not having the necessary dependency packages; the custom-builder is quite good at finding the necessary dependencies. When you’re finished, click ‘OK’. If you have dependencies that are missing, they will

now be offered to be downloaded and included. Do so if necessary. If not the next dialog to appear will be:



Click ‘Ok’. Files will be copied and eventually the next dialog will appear:

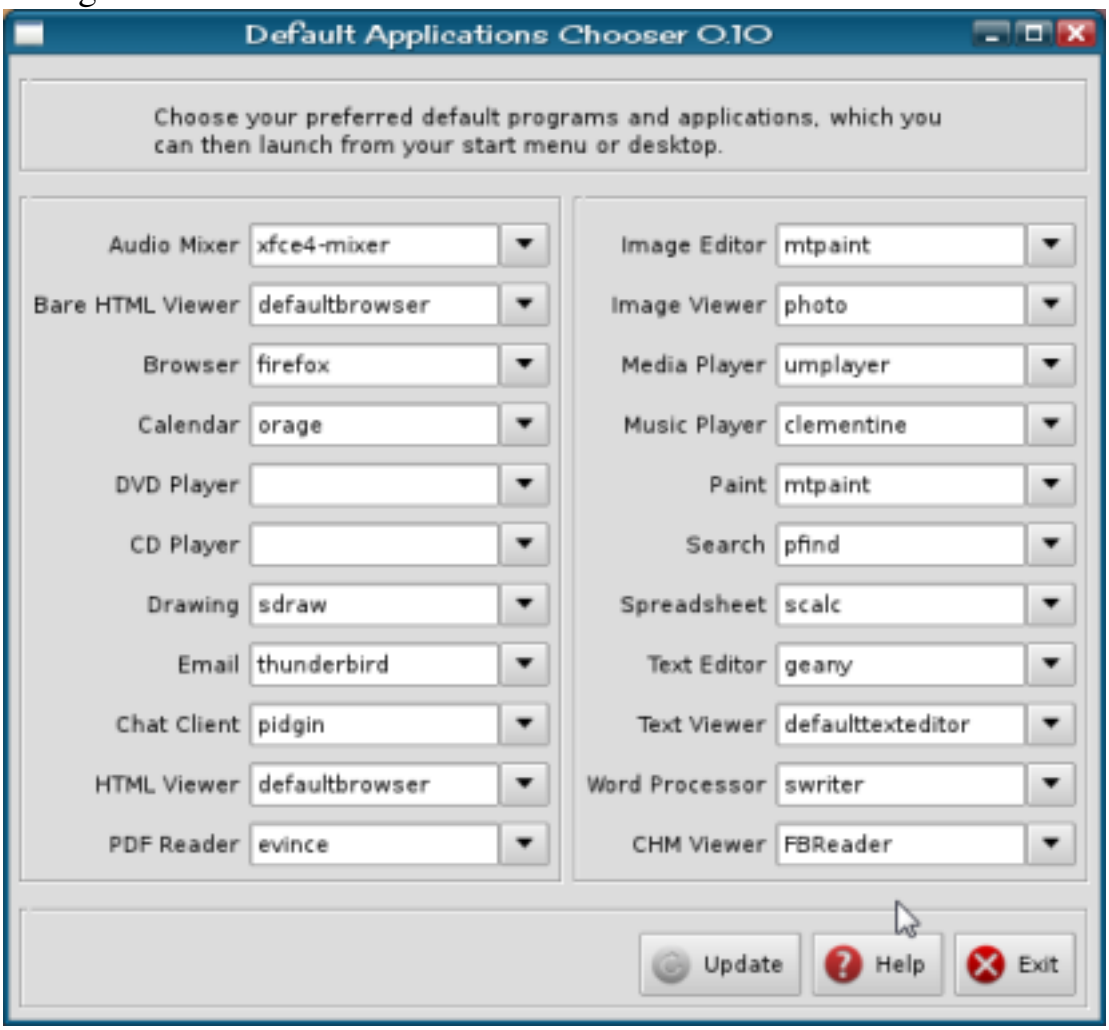


The missing shared library check. This is another area that causes consternation. I’ve built (literally) over 100 adrives. I’ve used this option once, and that was out of curiosity. Puppy packages are designed to be small, and many developers remove libraries that are not required in order to make the packages smaller. If you choose ‘Proceed’ be aware that you will find many missing shared libraries. And for a few packages some of these libraries will be ‘found’ upon running. Here’s the rule of thumb: so long as you are using packages from the standard repos in the carolina package manager, or packages from other sources that you have thoroughly tested and know to work, you will not need to choose ‘Proceed’. Save yourself the trouble; click ‘Skip It’.

This is the next dialog that you’re presented with:

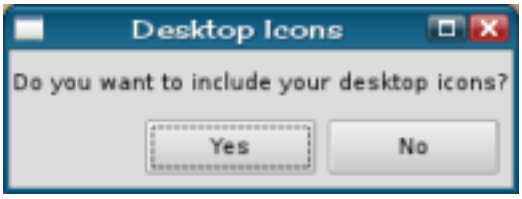
Before clicking ‘OK’ take the opportunity to set up your Xfce environment to your liking. Single click/double clicking in Thunor or the desktop, panel applets, wallpaper, themes, etc. Anything that has to do with Xfce, now is the time to set it up exactly how you like it. After making the desired changes, click ‘OK’ and the next dialog appears:

Click ‘OK’ to continue, and you’re presented with the Default Apps Chooser dialog:



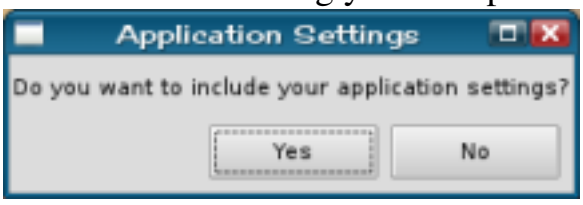
It is here that you connect the applications to their particular file types (mime handling). If you cannot find your particular application in the drop-down

boxes, you may type them in. After making the desired changes, press ‘Update’, ‘OK’ on the confirmation box, and the ‘Exit’ to continue. Here’s the next dialog box that you’ll be presented with:



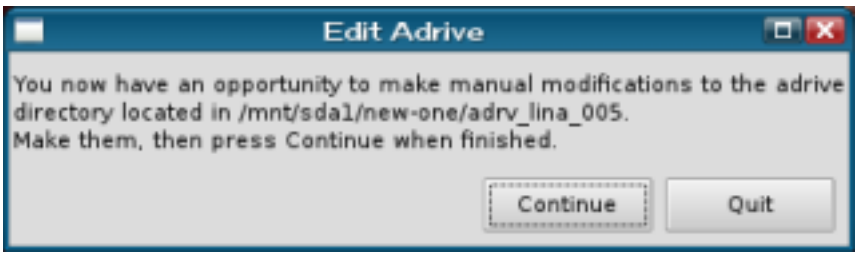
This one is self-explanatory, however there is one thing to note - if you’ve included the pup-volume-manager, or if you have desktop icons enabled in the Xfce desktop settings, then those icons will show regardless. These desktop icons refer to your custom icons that you may have on your desktop at present.

Here’s the next dialog you’ll be presented with:



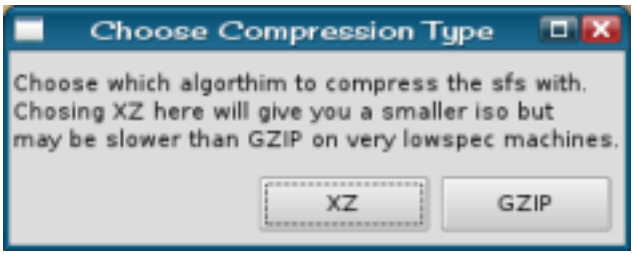
And it’s here that there is a definite bug. Regardless of whether you choose ‘Yes’ or ‘No’, you’ll get the following dialog:

If you want to save any or all of your personal settings, select the ones you want and press ‘OK’. You will then come to the following dialog, which is perhaps the most useful step of the process:

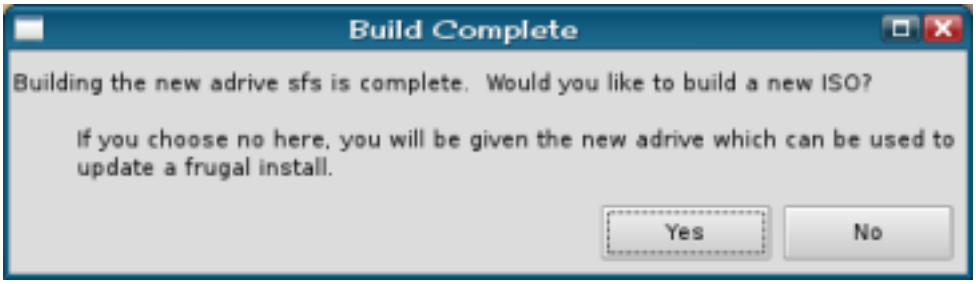


It’s here that you’ll be presented with this dialog box along with a Thunar window. In the Thunar window exists a mock-up of the adrive directory structure. It’s here that you can add/remove files, change settings, etc. The sky’s the limit. For example, you can copy your /usr/local/Frisbee to the (my example path) /mnt/sda1/new-one/adrv_lina_005/usr/local/Frisbee to include your personal wireless settings. Or you can add wallpaper images to the /mnt/sda1/new-one/adrv_lina_005/usr/share/backgrounds folder. There are quite a few other things that you can add/remove. You’ll have to experiment. Or ask questions on the forum. I’d be happy to answer questions that I can. But for the sake of keeping this tutorial short and sweet, we’ll not delve into it here.

Once finished making your changes to the adrive directory structure, click ‘Continue’ and you’ll come to the next dialog box:



Here you have your choice of compression. Pretty self-explanatory. I use XZ, but to each their own. Click the one you want and the SFS compression will begin. Depending on how many packages you included on your adrive (and the speed of your system) this could take anywhere between seconds and 20 minutes. Once finished you’ll be presented with the following dialog box:



You could choose ‘Yes’ to create a new ISO, and there’ll be a few more options/steps to go through, but for the sake of the tutorial (creating a custom adrive) choose ‘No’. You’ll then be presented with a confirmation box that tells you where your new adrive.sfs is located. You can then use this new adrive.sfs as a drop-in replacement for a frugal install. Please note that when doing so it is desirable to not use an older save-file with the new adrive. You’ll have better results by creating a new save-file from a fresh boot (think of it this way...if you’ve created new settings/apps on the adrive, and your save-file consists of conflicting settings/apps, you’ll end up with instability...better to use a new save-file with a new adrive...).

Hopefully this helps some of you that are new to the concept of the adrive and the basic usage of the custom-builder. As time allows, down the road I intend to provide more comprehensive documentation, such as how to swap out a kernel, etc. I hope you find this useful, and if so, please share your results and/or questions on the forum.