The Technician

Technician's Guide

This Handbook area is where the ultra-savy Wasta technician can help his peers. This person will be setting up "special" requests for others, preparing wasta-offline software update media, or perhaps "creating a new distro of Wasta" for a particular region. This person is familiar with setting up a normal Linux Mint or Ubuntu system "basics" and knows their way around the OS. This Guide also attempts to help a former Windows OS specialist, since many of the file management concepts are similar; just the syntax and the applications involved are different.

It is expected that command line examples under a terminal session will not confuse the user. It is assumed that the problems associated with Admin rights will also be easily solved by the reader.

News Report `(June 17, 2014)`

Notes on the new releases of wasta Linux:

wasta Linux 14.04

- **NOTICE:** as of 2014-06-15, key SIL / UBS Applications that use "mono" such as Paratext and Fieldworks will not be able to be run in wasta Linux 14.04. The development teams are working hard to solve these problems. Please use wasta Linux 12.04 if these applications are required. Note that other key SIL Applications such as Adapt-It and kmfl (for keyboarding) are available for installation and are working well in wasta Linux 14.04.
- Based on Ubuntu 14.04 "Trusty" / Linux Mint 17 "Qiana"
- Security updates provided through April 2019

wasta Linux 12.04.4

- Ability to use stable releases of key SIL / UBS apps that use "mono" such as Paratext and Fieldworks (WeSay is "very very close" as I understand).
- Based on Ubuntu 12.04.4 "Precise" / Linux Mint 13 "Maya"
- Security updates provided through April 2017

Is this Guide Clear?

The Wasta Manual has three main parts, designed to be separately printed as a "Handbook" for the given type of user.

It is the hope of the authors here that the explanations, particularly for the *Consultant* and the *Language User* be written for the "common man". Such a person may, or may not have prior computer experience, so explanations must be made clear. For comments and suggested changes please email: **brian_chapaitis@sil.org**[1]

Making a Bootable Memory Stick

The Internet source for Wasta Linux image downloads is at: **http://itconnect.info/group/linux-desktop/page/wasta-linux**[2]. There are several different Wasta Linux images available. A particular Wasta image is contained in a single large .iso file - which is approximately 1.5GB in size, so it is preferable if possible to download a Wasta image from a local server - espcially if your Internet connection costs would be high.

Which Wasta Linux Version should you use?

The first decision to be made is whether to install a 32 bit version of the Wasta system or a 64 bit version. Most computers today have 64 bit capable processors and hence it is possible to install the 64 bit version of Wasta Linux on them. Some people prefer, however, to stay with a 32 bit version of the operating system, since some older computers might require the 32 bit version of Wasta Linux. The main limitation of using a 32 bit operating system is that a 32 bit system cannot handle files that are larger than about 3.5GB at one time.

In addition to the architecture of the computer, there are versions of Wasta Linux which are based on different versions of Linux Mint. There are two main versions, the older version based on Linux Mint 13 (which is in turn based on Ubuntu Precise), and the newer version based on Linux Mint 17 (which is based on Ubuntu Trusty). The different versions of Linux Mint and Wasta use different versions of the Linux Kernel. Because of that difference, some software (such as FieldWorks and WeSay) cannot currently run under Linux Mint 17. Also in some cases, computer hardware works better on one version of the Wasta system than on the other versions. For example, the Lenovo Thinkpad laptops may have some problems with freezing up on the older Wasta based on Linux Mint 13 (Ubuntu Precise).

To download a copy of the desired Wasta **iso** image file while at Ukarumpa in PNG, you should contact Bruce Schwager at: **cts-netengr4@sil.org.pg**[3].

Once downloaded you will need a USB mem stick that is formatted with a FAT or FAT32 format. There are two main utilities that can be used to "burn" the iso image onto the mem stick and make the iso image a bootable image. You can use: "Startup Disk Creator" under Mint desktop, or you can use Unetbootin. The Startup Disk Creator is readily available only under Linux. The Unetbootin program is available for use either under Linux or Windows, but it would need to be installed in either case. For the purposes of this manual we will describe how to use the **Startup Disk Creator** which comes pre-installed on a Linux Mint system.

Note that using either utility is dangerous if you are not pointing the target to the mem stick and somewhere else by mistake, although the defaults are usually correct.

As mentioned above, there is a choice of versions that install for 64 bit machines and 32 bit machines. The PNG branch is experimenting exclusively with the 32 bit version, because for demo purposes and during classroom situations, most likely, Wasta Linux will be booting up on a very old machine. And yet, it appears that newer, more modern machines will still work with the 32 bit version even though one might want the 64 bit version for some speed improvements. The average user probably cannot actually perceive a difference in run times between a 32 bit, and 64 bit installation. Therefore, instead of supporting two separate versions of any given Wasta, the PNG branch is trying to support only one version at any one level of customization. There are going to be plenty of "versions" out there to manage as it is.

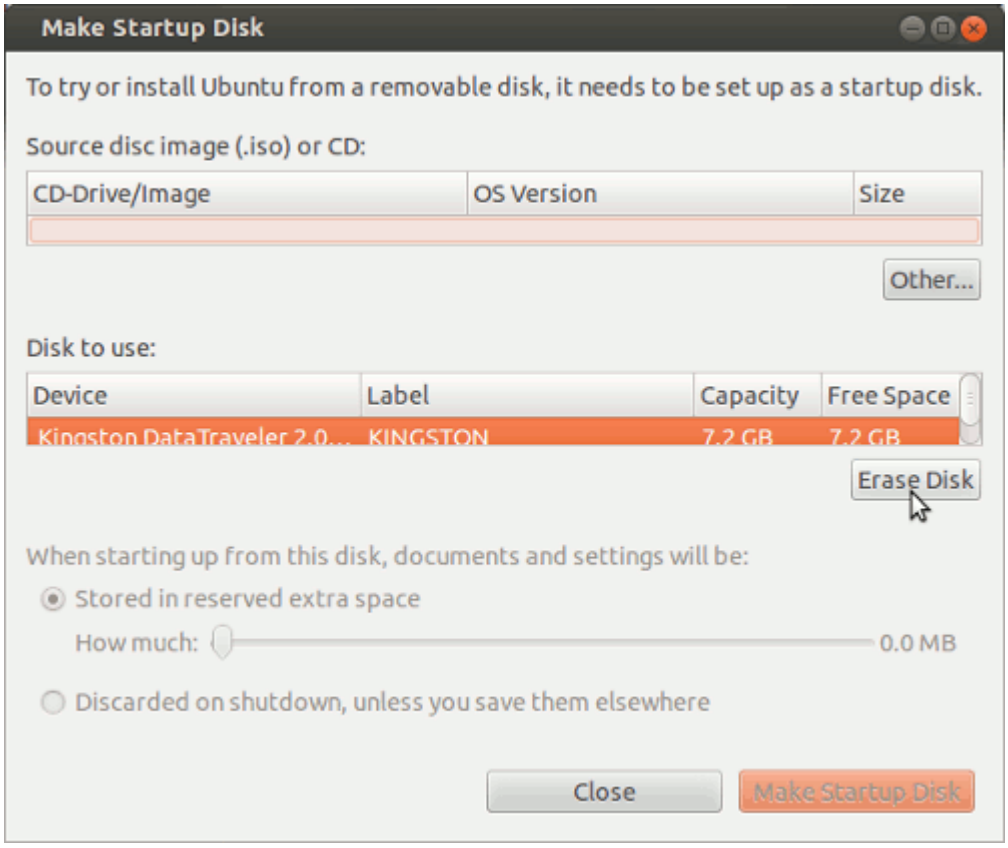
Creating a Bootable Memory Stick[4]

The steps below assume that you have the following:

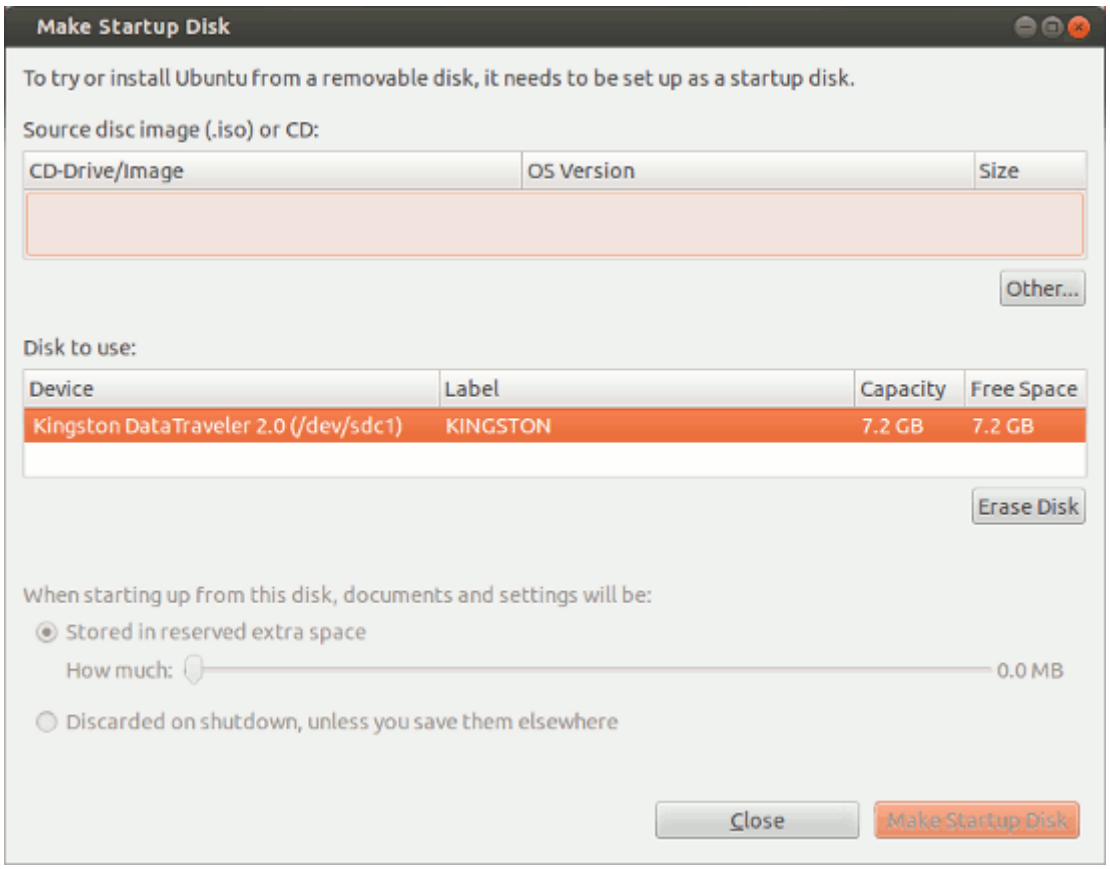
- a memory stick that is at least 2GB in size and that is formatted with a FAT or FAT32 file system
- the desired Wasta Linux iso image file - copied to a local folder such as the Downloads folder on the computer you are using to create the bootable memory stick
- you will use the Startup Disk Creator to make the memory stick into a bootable Wasta Linux "live" system

First you need to **insert the memory stick in a USB slot on your computer**:

Next, run the **Startup Disk Creator** program. It should be located on the **Preferences** menu. The Startup Disk Creator will initially look something like this:



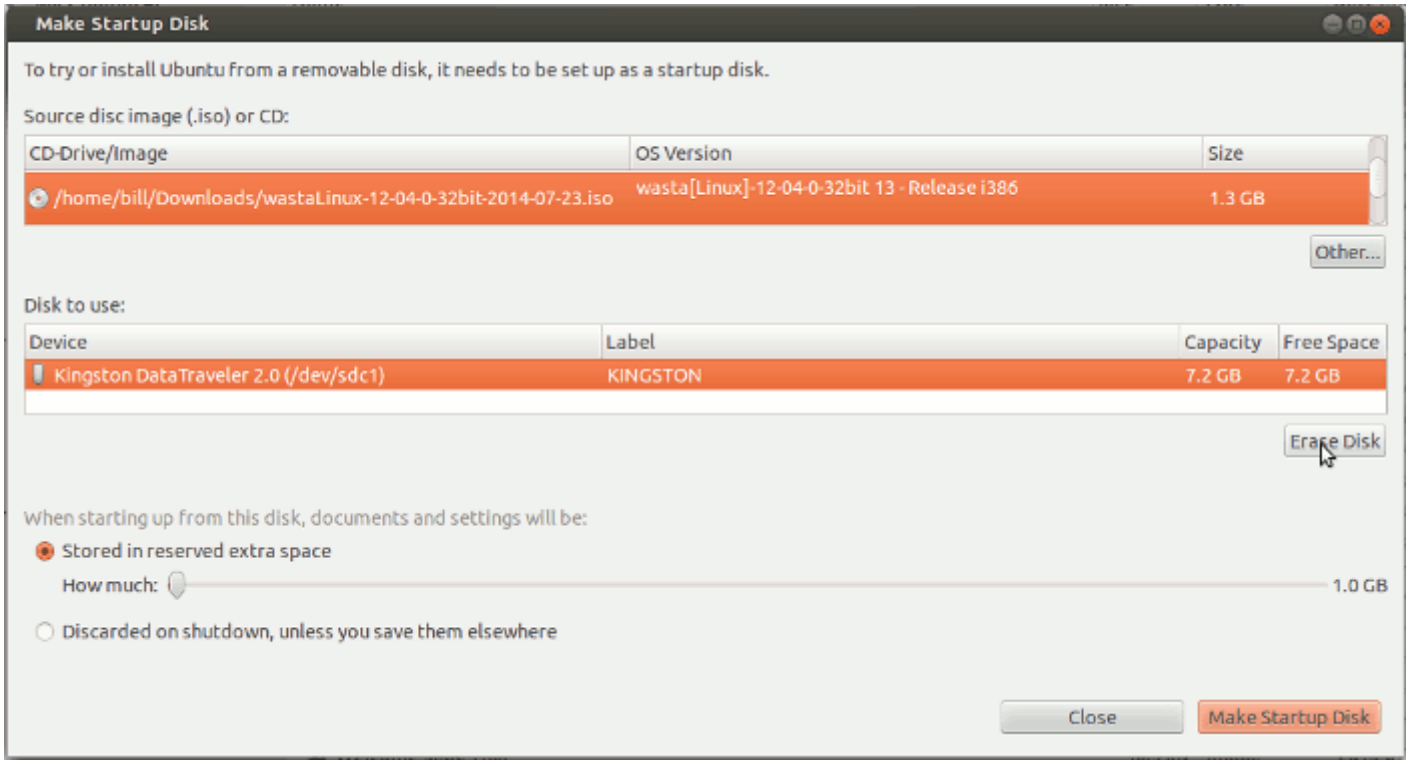
Grab the lower right corner of the program's window frame and drag it to expand the window frame to make it taller and wider. Doing a resize of the window frame will make the program's content easier to read. After expanding the window it should look something like this:



The first time you use Startup Disk Creator the "Source disc image (.iso) or CD" list will be empty. In subsequent uses of the program any previously used iso image files will also appear in the list. The "Disk to Use" part of the window should list the memory stick that you have inserted into the USB slot.

Click on the "**Other**" button at right and below the top list. A file dialog will appear. Use the file dialog to navigate to the Downloads folder - or whatever folder your iso file is located in, and highlight the desired **iso** file to be used for the bootable system:

With the desired iso file selected, click "**Open**". The Startup Disk Creator should now appear something like this:



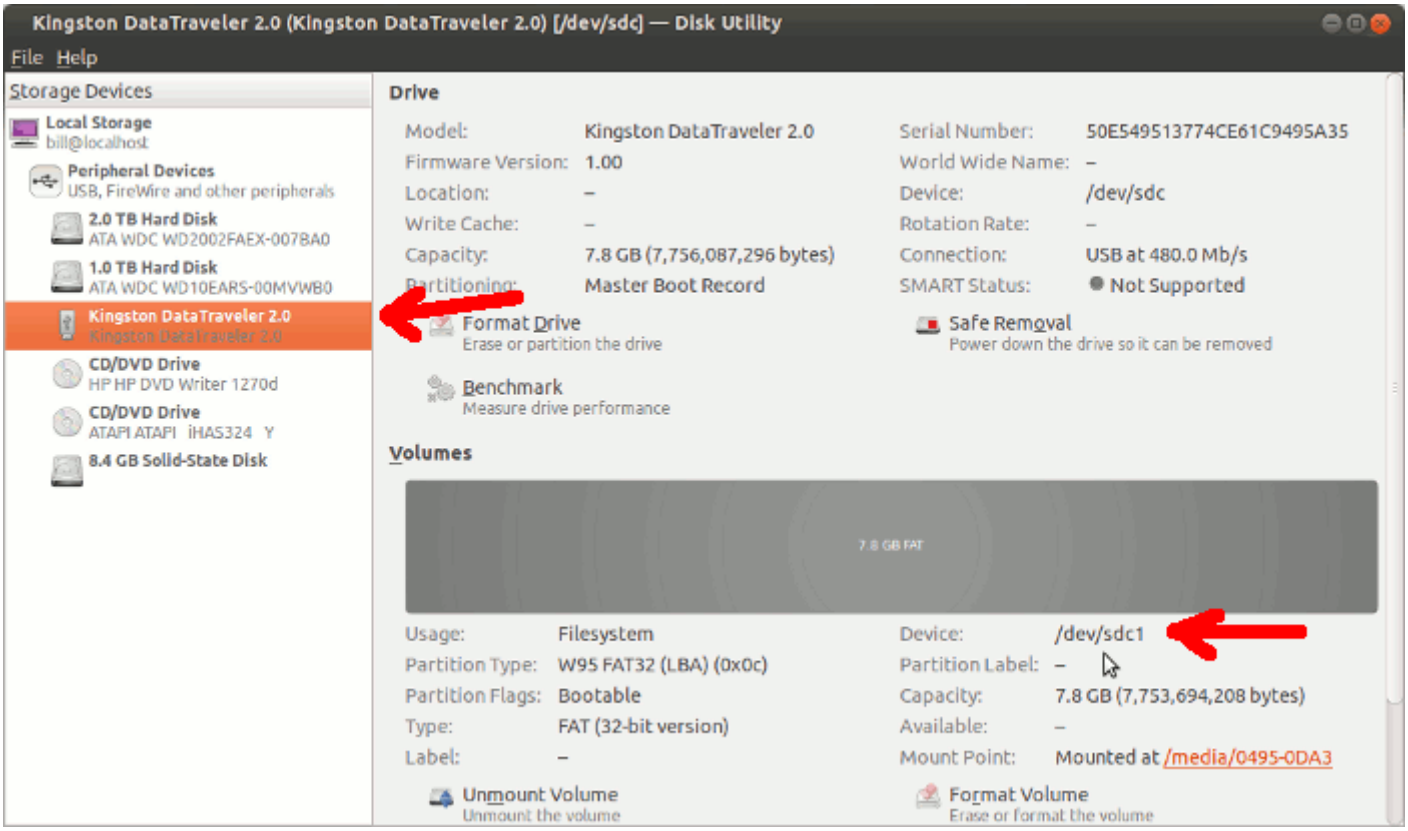
Warning: Any data that exists on the USB memory stick will be erased when the Wasta bootable drive is created.

Now, before you continue it is important to verify that the device under the "Disk to use" selection is indeed the device that represents the memory stick that you inserted into the computer and that will be dedicated to becoming the bootable Wasta system. Usually the Startup Disk Creator accurately detects the existence of your memory stick, but we will double check just to make sure. You don't want it to erase your main hard drive or some other disk that is connected to your computer.

Look carefully at the information in parentheses for the device under the "Disk to use" selection. Write down the information in parentheses. In the above example it is:

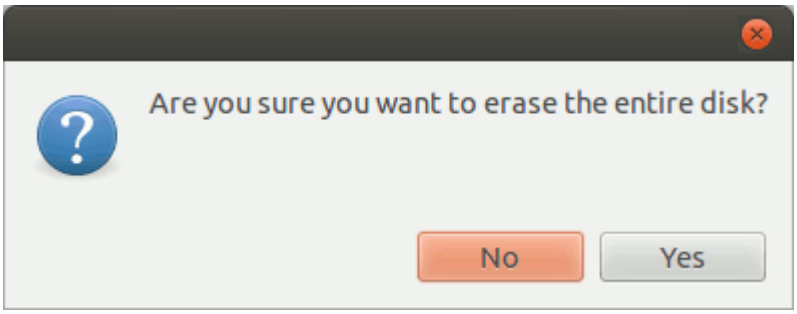
/dev/sdc1

OK, before proceeding and while the Startup Disk Creator program is waiting, go to the **Preferences** menu and select "**Disk Utility**" (older Wasta version name) or "**Disks**" (the newer Wasta name). The Disk Utility (Disks) program will appear:



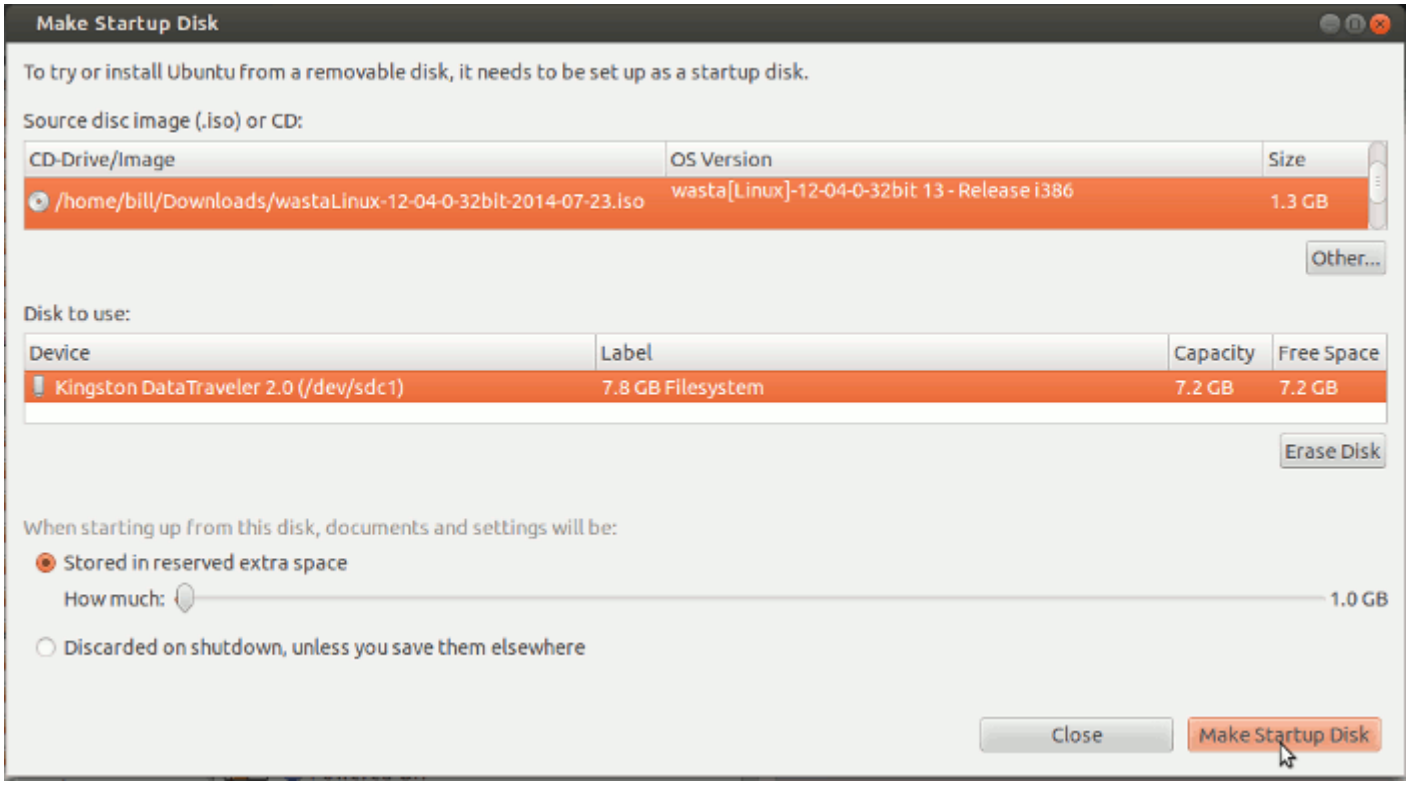
Select and highlight the item under the "Peripheral Devices" column at left that you know to be the memory stick that is to be used for your bootable Wasta memory stick (the red arrow at left points to my Kingston Data Traveler 2.0 memory stick - yours will have a name to match your memory stick). Then with the memory stick selected at left, look at the Device in the right hand part of the screen. **Make sure that the device listed here is the SAME as the device you wrote down above (here it is /dev/sdc1)**. If the device names match then you can continue with the steps below. If the device names don't match then you should abort this process until you can determine for certain that the device you will use is accurately selected in the Startup Disk Creator program.

If you are confident that the device listed in the "Disk to use" section of the Startup Disk Creator is the correct device, then you should click on the "**Erase**" button at the right and below the "Disk to use" area of the screen. This will erase the contents of the memory stick in preparation for making it bootable. The following message will appear:



Click "**Yes**" to confirm that the memory disk can be erased. The system will ask you for the computer's password:

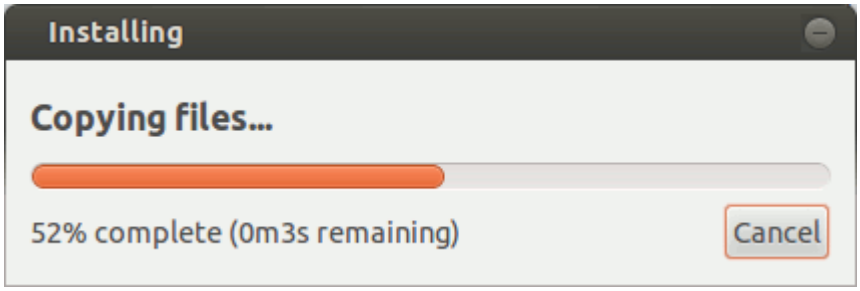
Enter your password and click on the "**Authenticate**" button. In a few seconds the Create Startup Disk program screen will be updated and should then look something like this:



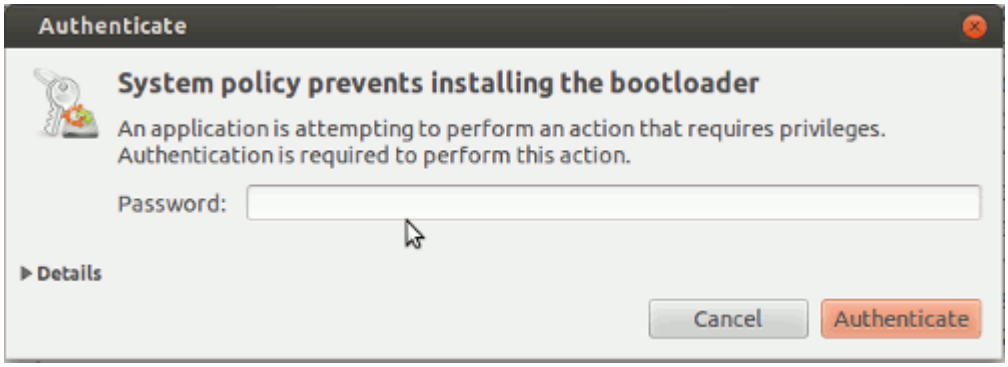
Double check that the iso image in the top window is the Wasta iso you want to use to make the bootable Wasta memory stick, and that the Device selected in the lower window is the device that represents the memory stick that is plugged in to your computer.

There is an option change the amount of "reserved extra space" on the bootable disk. This setting is optional and usually you won't want to change it. It is also known as "persistence" which means that when the USB mem stick boot goes "live" that data and changes will be preserved for later boots off mem stick. Note that if you change the reserved extra space setting, it may take longer to create the bootable mem stick from the .iso file.

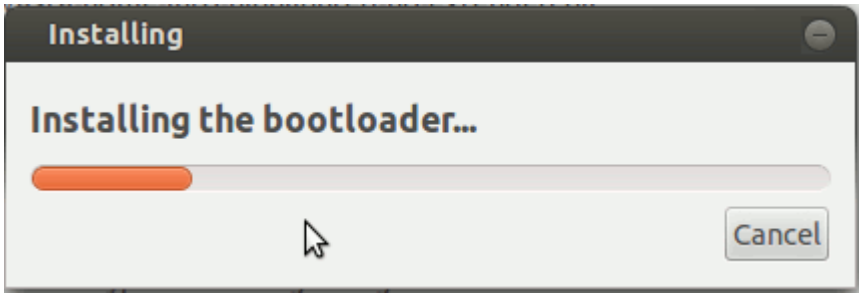
Click on the "**Make Startup Disk**" button at the lower right to start the process of creating the bootable disk. Normally, the process takes approximately 10 minutes to complete. You should monitor its progress as you will have to supply a password when prompted. You will see the following progress dialogs:



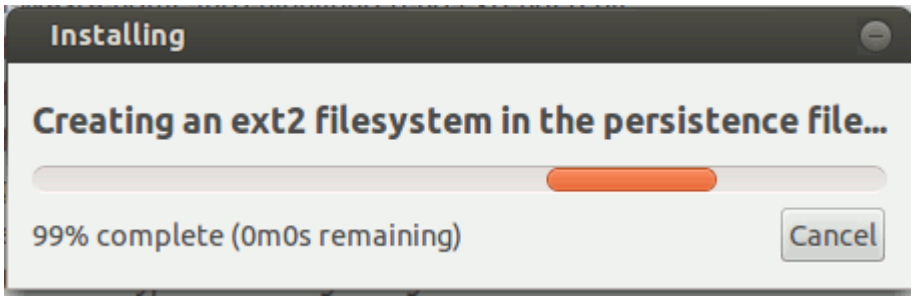
and after a short time you will be asked to authenticate again with the following dialog:



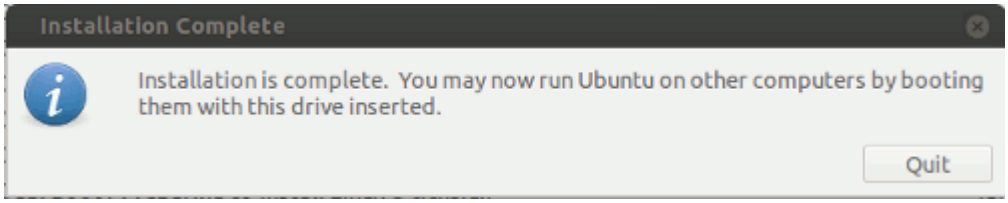
Enter the password for the computer and click on the "**Authenticate**" button. Then during the remaining time you will see other progress dialogs as follows:



then:



and finally:



When the "**Installation Complete**" message appears, click the "**Quit**" button. it is now safe to remove the bootable memory stick.

You should test that this newly created memory stick works by using it to boot a computer to a live session of Wasta Linux. Once you have verified that it works as a bootable live session, you can shut down the live session. Make a small label and attach it to the memory stick (or write directly on the memory stick with a marker) indicating which version of Wasta Linux it was made from and its date as given in the iso file name it was created from, something like this:

Wasta-Linux-12-04-0-32bit
2014-07-23

and store away the memory stick for later use.

Booting into a Live Session of Wasta

If you need to make a bootable memory stick of a Wasta Linux image (.iso file) see the above section on Creating a Bootable Memory Stick[5].

Booting to a live session of Wasta[6] is described in detail in the Consultant Guide.

After booting to the live session, you should carefully test and ensure that the computer's hardware works under Wasta Linux[7]. This topic is also covered in some detail in the Consultant Guide. It is NOT necessary to do a full install with the Internet present to test hardware at this point. If Wasta Linux has any problems with your computer's hardware, you want to know about it and possibly find an acceptable solution before proceeding with an actual installation, otherwise you will run into the same problem (there are exceptions) after you do the full install. "Try before you buy into" the Wasta concept here.

Checking the Hardware

Wasta Linux is capable of running on "high powered" computers, and it is also designed to allow "low power" computers to run well. Within PNG, however, in the wild and during open workshops, users may come from other agencies and bring any kind of possible hardware, including Apple OS hardware. Wasta will boot up, but with so many different hardware platforms out there, one should test as much as possible before committing to a full hard-disk install.

Here are the general areas that you should test and check that they are working properly before committing to a permanent installation of Wasta Linux:

- **WiFi**: Most Intel chipsets should work out of the box, but some need a little help. Broadcom chipsets just need the proprietary driver that can be installed in the Additional Software dialog of Ubuntu/Mint. I suggest you have an Internet connection, however, up and running.
- **Ethernet**: This should simply work.
- **Speakers**: Work almost always. If you don't hear anything, make sure that the speakers aren't set to mute.
- **Microphone**: Works often, but not as often as the speakers. Be sure to check this.
- **Webcam**: Works most of the time, although some Apple products have lower rates of success.
- **Function Keys** on your keyboard such as display brightness, keyboard backlight brightness, media controls, and more. If they don't work, see whether you really need to use them or not. There may be a work-around via the Main Menu.
- **Ports, including USB, HDMI, etc**: These should work, but it's good to check. USB really shouldn't be an issue, but video ports like HDMI are more important to test.
- **Graphics card**: Does it perform well? If you are running a newer AMD or NVIDIA graphics card, you'll probably want to use the proprietary drivers. Warning: You **can't** really test these in a live environment, because they require that you restart, but when you restart a live environment you lose everything. So for a true graphics card software change and test, you must go ahead and install Linux onto the hard-drive.
- **External devices such as your printer**: Printers are important to test. Some work flawlessly (this is more the case for

HP printers), but others are complete duds. If you have a choice of printer manufacturers, HP printers are BY FAR the easiest to set up.... for some reason the folks at HP really want the Linux community to join them.
- **Important Software:** Yes, we know, so this isn't about a hardware test at all! But go ahead and make sure that your most important apps, like Paratext are working well. For example, what is the screen like when scrolling multiple windows under Paratext? Are there any major software bugs/ features that you simply cannot do without? If the software is not running well... then do not proceed with the install.

Installing Wasta onto the System Hard Drive

There are a number of different ways that Wasta Linux can be installed on a given computer. The Consultant Guide has a discussion about how to decide to install Wasta Linux[8]. That information is repeated here for convenience:

If you have decided to install Wasta Linux on your computer, you will usually have two main options:

- Installing Wasta Linux to Dual Boot with Windows[9]
- Installing Wasta Linux as the Only Operating System[10]

Deciding which type of installation is an important decision. Today, Wasta Linux has nearly all the software programs that might be needed in a typical language and translation program. We recommend that you back up all your documents and other important files to an external hard drive or to some other secure place, and then install Wasta Linux as the only operating system on your computer. In this case the Windows operating system will be completely deleted from your computer, and Wasta Linux will take its place. After installation, the documents and other important files that you backed up can be copied back over to the new Wasta Linux installation on your computer.

How do you decide if you should install Wasta Linux as the only operating system on your computer? A Wasta-Only installation might be the better option if any of the following are true of your situation:

- If your computer has Windows XP on it. Windows XP is no longer supported by Microsoft with updates and security fixes so it will rapidly become a significant security problem.
- If your computer does not have at least 15GB of unused space on its hard drive and you want to try Linux.
- If you are tired of having your computer threatened by viruses, trojans, and other security issues, and tired of having its performance degraded by anti-virus programs running in the background, and tired of being distracted by fragile, slugish Windows Updates.
- If you don't want to be locked in to one proprietary vendor - paying a high price to Microsoft for licenses to use their operating systems and software, and find it difficult in your region to even find non-priated copies of Windows software.
- If you want to identify with those who treasure the freedom we have while living and working in the world of Linux and open source software, and enjoy being able to give valuable software tools away to your friends without being hindered by restrictive licensing agreements.

If you want to make Wasta Linux your only operating system on this computer you can go ahead and follow the instructions in the Consultant Guide for Installing Wasta Linux as the Only Operating System[11] [12].

How do you decide if a dual boot installation is a better option for your situation? A dual boot installation might be better if **all** of the following are true of your situation:

- If there are certain must-have Windows programs that you are certain have no equivalent on Linux.
- If your must-have Windows program cannot be run within Linux using the Wine emulator.
- if you have at least 15GB of free space on your hard drive (preferably more).

If all of the above are true of your situation, then you may decide to install Wasta Linux as a "Dual Boot" option. In order to install Linux alongside Windows you must shrink down the size of the disk partition(s) devoted to Windows to make space to install Wasta Linux and to also have space to create new data in the space reserved for Linux. If you want to preserve your Windows operating system, and have the option to boot into Wasta Linux, you can go ahead and follow the instructions in the Consultant Guide for Installing Wasta Linux to Dual Boot with Windows[13].

Regardless of which installation method you have decided on, in the detailed instructions in the Consultant Guide we will tell when you get to the "Point Of No Return", that is, the point at which proceeding further will actually make permanent changes to your system. Before you get to the point of no return, you will always be able to cancel or quit the installation process without making any changes to your existing computer's system and its files.

IMPORTANT: You should always back up your important files before continuing with either type of installation.

The following two topics are discussed in detail with screen shots in the Consultant Manual. As a Wasta Technician you should be familiar with the following ways to install Wasta Linux on a computer (all are detailed in the Consultant Guide):

Installing Wasta as a "Dual Boot" machine leaving MS Windows in Parallel[14]

Installing Wasta as the Primary and Only Operating System[15]

Re-Installing Wasta over an Existing Wasta System[16]

Creating the Wasta-custom-png Version of Wasta

The primary developer of Wasta, "Rik Shaw," does not know what the various SIL Branches will want in their unique "standard" regional creation of Wasta. This means that if PNG Branch as a "base case" wants Paratext, Adapt It, and Xiphos (to name just a few programs) already installed in their "base case" before further modifications, then a PPA gets specified where various installation scripts can be run later by the Technician. This is just a handy way to automate the process. The source of the PPA lives in a launchpad repository for Wasta development, and yes, we (PNG-Uka) are registered (it is very complex registration) and yes, we can control and alter these later. But for now, we are using the "base case" that Rik Shaw has given us to get started. The very cool part of this approach is that when a new version of Ubuntu or Mint comes out every 6 months or so, then we can get the latest and then automate the primary software programs and system settings that we want for a new version of the "base case" for PNG Branch.

The basic steps for creating a custom version of Wasta Linux for any region are found on the Internet under the topic wasta-Linux: re-customizing for your needs[17].

The basic steps are:

1. **Create a wasta-custom-xyz Package**: Work with the wasta-Linux team to create a new "package" called 'wasta-custom-xyz' with all your desired customizations included. The wasta-Linux team will create the first version of this package for you. Update Note: The wasta-Linux team has already created a customization of Wasta for PNG called wasta-custom-png. When the 'wasta-custom-png' package is installed on any wasta-Linux computer it will automatically include all of our desired customizations for PNG.

2. **Maintain the wasta-custom-xyz Package**: Ideally any future modifications to this 'wasta-custom-xyz' package will be made directly by you, the regional customizer. The process to maintain 'wasta-custom-xyz' packages includes:

- Setting up your ID in Launchpad to contribute to the wasta-Linux software repository.[18] This has already been done.
- Making modifications to a wasta-custom-xyz package.[19] As of September 2014, this part remains to be done.

(More to be written)

What's found under "Wasta-custom-png-uka" now that you have it

At this point there is a nicely functioning full Wasta environment with important applications already installed. In the list of very important programs would be Paratext, WeSay (when available), Adapt It, and Xiphos. Plus there would be a host of standard applications included such as Firefox, Thunderbird, Audacity and the rest, that could be useful in a language programme, but someday later when the user is ready for them.

The goal now is to create a new localized "standard" called "Wasta-custom-png-uka" which would be a new instance with various settings "standard" and particularly with NTC (National Training Course) students in mind. In your case (the reader) you might want to take a standard version, and do your own modifications for a particular language team and their unique and specific needs.

Once you have your new version set and ready, you may want to "create your own" Wasta distribution using a tool called "Remastersys". This program too, is already loaded in the system and ready for use.

Adding an Icon to the Desktop

This is a Consultant level topic. Please refer to that section[20] if you do not know how to do this task.

This follows normal Mint 13 procedures under Cinnamon.

Discover Versions

You will want to open **Menu > Administration > Synaptic Package Manager** and search for "wasta-base-setup", and / or "wasta-custom-png", where it will then list the version number in the right hand column.

Version 12.04.5.0 and 0.1.12 as of this writing.

The Standard Run-Prompt

In Linux Mint pressing these keys **Alt + F2** will gives the standard **"Run prompt**". For a better description please refer to the Consultant's section on this topic.

Wasta Menus: Hiding Standard Menu Items

There are two hidden commands under Wasta-custom-png that affect the display of menu items and their categories. There is a "standard set" of available programs in the system, the Technician might want to suppress most of them using a pre-made script called "wasta-menus". These commands can be run either from terminal session (simultaneously key-presses **Ctrl + ALT + T** or the run prompt shown above. (Note: Tab auto-complete does work under Wasta)

You have 2 main commands:

wasta-menus-config

wasta-menus-add-app

Of the many Linux programs that are presented as a default under Wasta, the Technician can now suppress a set of these which will no longer display when Wasta-menus is "ON". If you want to revert to the "normal" or the default settings for Wasta-custom-

png, then turn Wasta-menus "OFF".

Notice how many standard Mint 13 applications have been removed from user presentation. But not all. The developers felt that the remaining set of applications might be more common and now the Technician can work from this reduced set of possibilities. This is also a safety net of sorts, keeping more dangerous utilities far away from the Language user and even the Consultant user. The Consultant would normally know the hidden commands for adjusting menu items, but now even those more dangerous applications have been suppressed.

If you are not familiar with this feature of the Wasta system, then refer to the Consultant's section under this topic.

Advanced Note:

The "default apps" are symlinked to /usr/share/wasta-menus/wasta-menus-default-apps.txt. Since you may not want some of those defaults (it does include VLC, for example), then when wasta-custom-png is created, it will OVERRIDE the wasta-menus-default-apps.txt with whatever is defined in wasta-custom-png. For changes to take effect in the "apps.txt" control file, you have to restart the user session (logout/login)

Note that Wasta is using the standard Gnome Categories, which means that after you "add an application" it will already be categorized, instead of needing manual category assignment.

Further Reductions or Simplifications: The Minimalist Presentation

If the Technician desires a further simplification of the Main Menu items, then standard Mint 13 procedures would apply. For a detailed description please see the Consultant's section of the Manual. It is quite possible that for your "standard user" you do not want certain Internet programs to be presented, because you anticipate that the "standard user" will not generally have Internet access in the "base case".

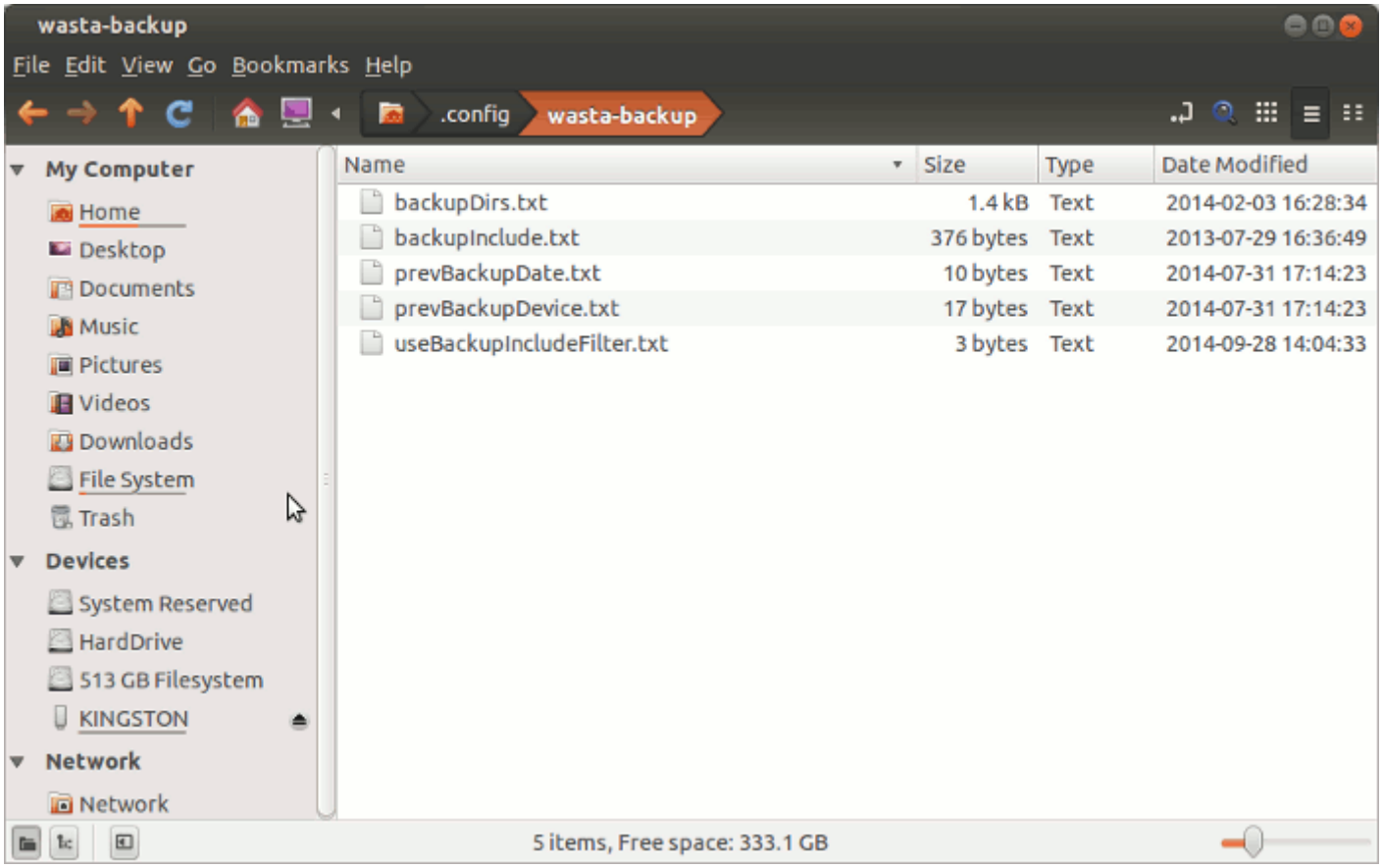
For the National Training Courses (NTC) we plan to have a standard "Wasta-custom-png-uka" live-distro that will be rather minimalist in approach, and the Consultants will be able to add programs back into the GUI presentation as needed with a simple check-box, Alacarte system. This will be explained in the Consultant's Handbook in their section of this Manual and they do not need to know about the complexities of Linux at all.

Wasta-Backup Control Files

Like most programs under Linux, Wasta-Backup has certain control files that can be adjusted to change the behavior of certain programs. These are usually human readable files with comments that can be edited with a normal text editor, such as **gedit**.

Wasta's backup control files can be found together in the hidden folder at this location within the user's Home directory: **~/.config/wasta-backup/**.

Use **Alt + F2** to get the standard run prompt and enter **nemo** at the prompt to summon the Nemo File Manager. Since the files we want to access are in a hidden folder, we need to make the hidden folders visible by pressing **Ctrl + H** (or select "Show Hidden Files" from the View menu). Scroll down through the folders until you can see the **.config** folder. Open the **.config** folder and scroll down to the **wasta-backup** folder and open it. You should see



So, for example, it is possible a user is successfully using wasta-backup, making historical backup data files onto a memory stick inserted into the computer – but they have a special request that photos or other file types be included, that are not normal to wasta-backup. The control file to edit is the one called **backupDirs.txt** shown in Nemo screen shot above. You can edit the file by double clicking on the backupDirs.txt file, and the file will open in the gedit editor. The file will have contents that look similar to this:

```
 1 #Folder Display Name,Folder Location,useInclude,Additional rdiff-backup Parms,Remove Older Than
 2 #  useInclude = YES ==> only include specified filetypes in backupInclude.txt
 3 #  useInclude = NO  ==> don't limit backup to specified filetypes in backupInclude.txt
 4 #  Remove Older Than options:
 5 #          30B means keep versions from the past 30 backup sessions, delete older ones
 6 #          3M to remove versions older than 3 months
 7 #          1Y to remove versions older than 1 year
 8 #          1Y3M to remove versions older than 1 year 3 months
 9 #
10 Paratext,$HOME/ParatextProjects,NO,,1Y
11 Adapt It,$HOME/Adapt It Unicode Work,NO,exclude ignorecase:**/.temp,1Y
12 Thunderbird,$HOME/.thunderbird,NO,exclude ignorecase:**/Cache,1Y
13 Documents,$HOME/Documents,NO,,3Y
14 Desktop,$HOME/Desktop,NO,,3Y
15 Pictures,$HOME/Pictures,NO,,3Y
```

The last line shown above:

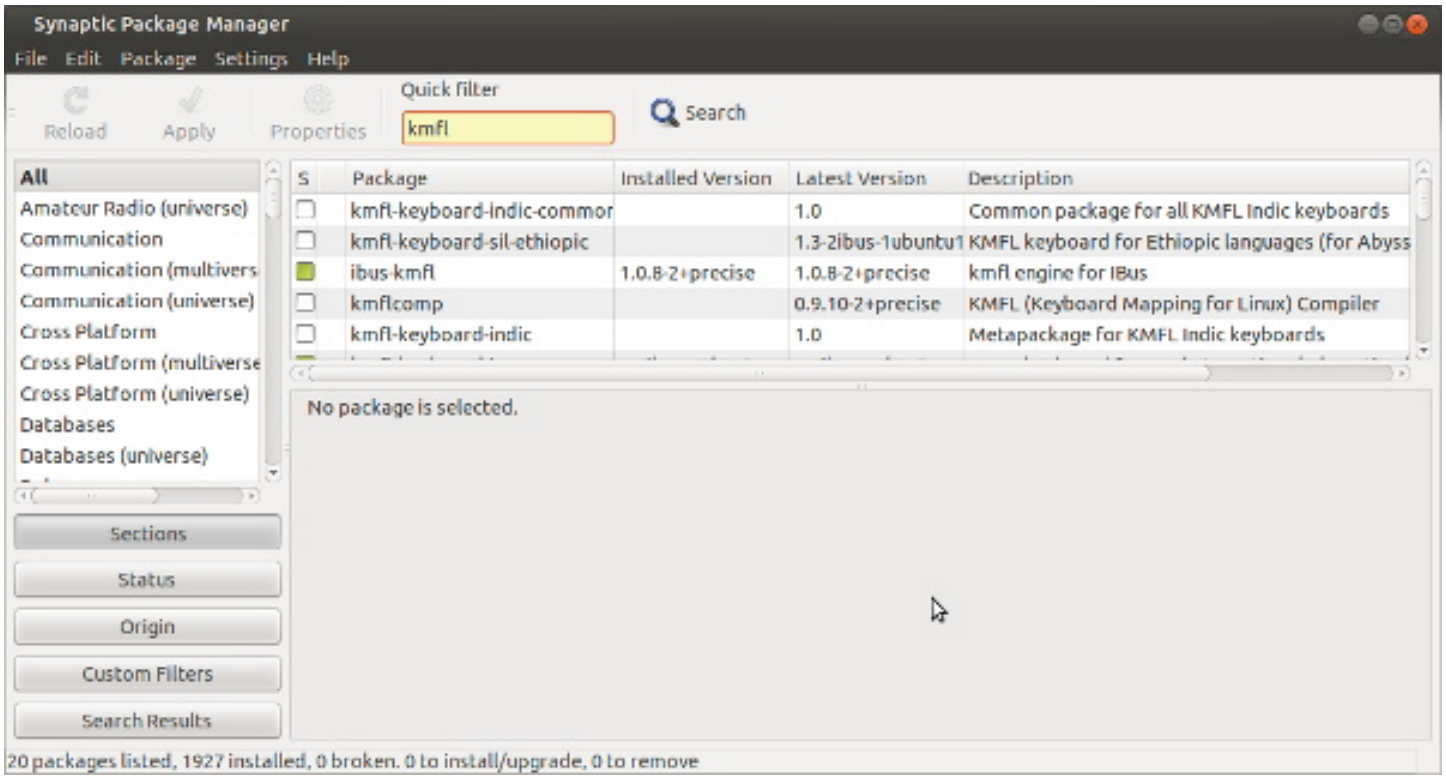
Pictures,$HOME/Pictures,NO,,3Y

is what you would need to add to the end of the file to cause wasta-backup to include the items in the user's Pictures directory in the backups made with wasta-backup. You can copy and paste the line above. If you manually type it, take care to include the proper number of commas to separate the fields as shown above.

Custom Keyboards

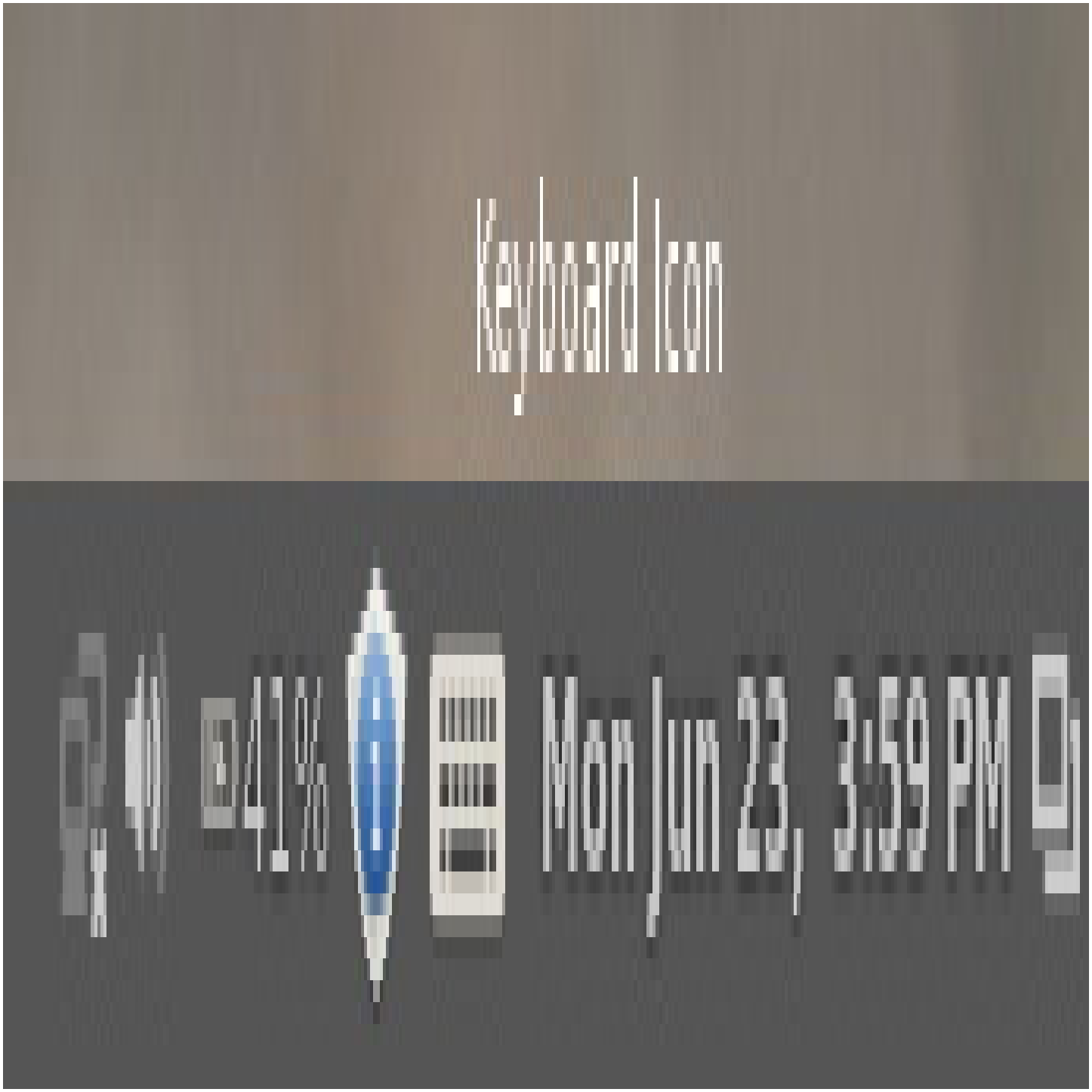
Keyboard Mapper for Linux (KMFL) is the tool of choice under Wasta/ Mint/ Ubuntu. It is based on IBus. For a more detailed technical discussion go to the SIL Linux Development site: http://linux.lsdev.sil.org/wiki/index.php/Installing_KMFL_on_Ubuntu[21]

The KMFL application ibus-kmfl should already be installed in Wasta-base, but in case it is missing you can check under Synaptic Package Manager and then install as normal



Note the green check box here, for ibus-kmfl, indicating that this system is already installed.

Then as further confirmation, there should already be a small keyboard icon in the toolbar, ready to be used.



A simple keyboard indicates a generic English oriented keyboard for the most part, or whatever the main system language is, such as Spanish.

If you still do not see the generic keyboard icon in the toolbar area, then we need to run iBus and we need to have it run/ loaded automatically at startup. So under terminal you can run the command:

ibus-daemon

Just to make sure it can be run. And then you can add this command under the "startup" utility under the Menu | Startup Applications. You will have to "add" a new iBus item in the list presented under Startup. Click the "add" button. To test you will need to restart Wasta and then check that the generic keyboard item is present now in the toolbar.

Of course, if the language orthography is normal English Latin characters, then you don't really need iBus at all.

Adding a new Keyboard

Most likely the Technician will be called upon to create a custom language keyboard, or two, and install for a given team or a set of teams in a multi-language project. The key folders to know about are:

/usr/share/kmfl/icons/vernacular.bmp

/usr/share/kmfl/vernacular.knm

Substitute a good name for "vernacular" above.

You can have more than one vernacular keyboard .KMN definiition file and more than one corresponding .BMP icon. The ANSI text file for the definitions follows the normal "Keyman for Windows" syntax for keyboards, so if the user is coming with a pre-made .KMN file, then try that one first for the project. However, very complex syntax Keyman files, might not work under KMFL. The Icon files are any graphic you want reduced to 24 x 24 pixels. The technician should have a library of useful "Letter" icons already, such that one can easily pick a letter representing the name of the language. The name of this .BMP graphic is specified in the .KMN file. In our examples here, we are working with the Kala language. So an example for Kala keyboard definition as a text-only file looks like this:

```
c Keyboard for Kala (John and Amy Lindstrom)
c Prepared by Brian Chapaitis for Wasta Linux, Unicode syntax
c 31-May-14


NAME "Kala"
BITMAP "kala.bmp"
VERSION 6.0


begin Unicode > use(main)


group(main) using keys


';' + 'a' > U+00E1    c á[22]
';' + 'e' > U+00E9    c é[23]
';' + 'i' > U+00ED    c í[24]
';' + 'o' > U+00F3    c ó[25]
';' + 'u' > U+00FA    c ú[26]


';' + 'A' > U+00C1    c Á[27]
';' + 'E' > U+00C9    c É[28]
';' + 'I' > U+00CD    c Í[29]
';' + 'O' > U+00D3    c Ó[30]
';' + 'U' > U+00DA    c Ú[31]


';' + ';' > U+003B    c ;[32]
';' + 'n' > U+014B    c ŋ[33]
';' + 'N' > U+014A    c Ŋ[34]
```

The Kala.kmn file

Note that the binary files from Keyman for Windows .KMB are unusable in this system. Unicode notation was used here, and appears to be the "standard" for KMFL definition files. If you are unfamiliar with Unicode character possibilities then launch "*Character Map*" under the Wasta Main Menu and look around.

In the Kala keyboard, the semicolon key is called a "dead-key" because it does not appear to do anything at first press, and waits for a second press before action. The semicolon precedes the vowels and then the appropriate accented character is displayed. The N consonant is used for the "eng" character, or velar nasal, popular in many PNGan languages. Also note that by assigning a "double semicolon" press, we can retain the original semicolon for punctuation, if the user wants that. The "c" symbol is for "comments" in the file.

The company Keyman.com has a tutorial for Keyman syntax here: http://help.keyman.com/developer/8.0/docs/tutorial_keyboard.php[35] but beware that not all syntax elements appear to be truly supported by the Open Source, Keyboard Mapper for Linux.

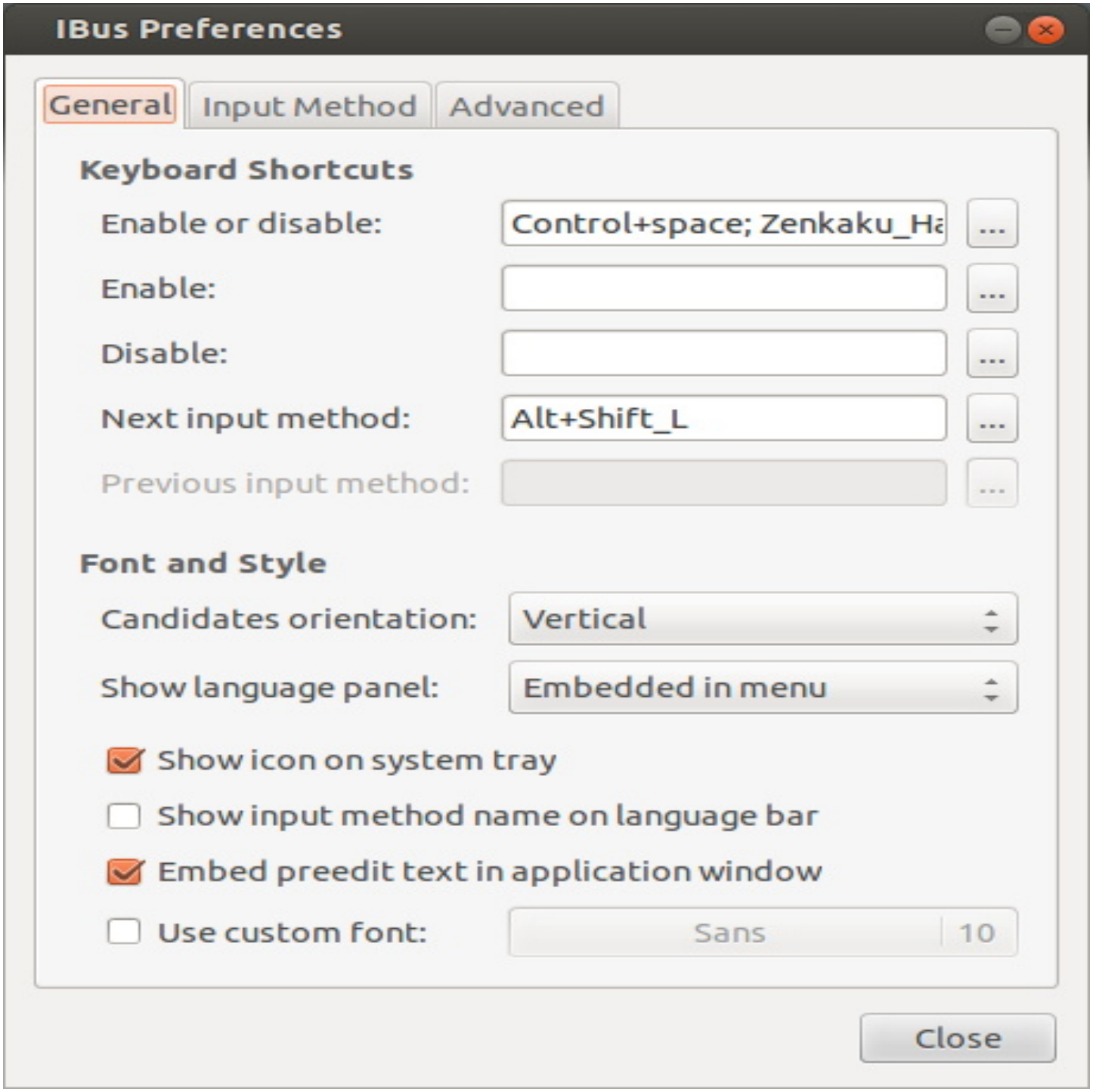
Setting the Input Method for IBus

Once the keyboard definition files are specified and placed in the correct folder, plus the appropriate icon file, then one is ready to specify this keyboard and others from a list of available keyboards. In the Wasta-custom-png distro, there will be a library of pre-made keyboards to choose from in the future, and the library will hopefully grow in time. So, once the technician adds their

examples, then they must be chosen as possible input methods, but not **all** keyboards need be chosen. The Technician only has to do this once for the given language project, as an initialization step.

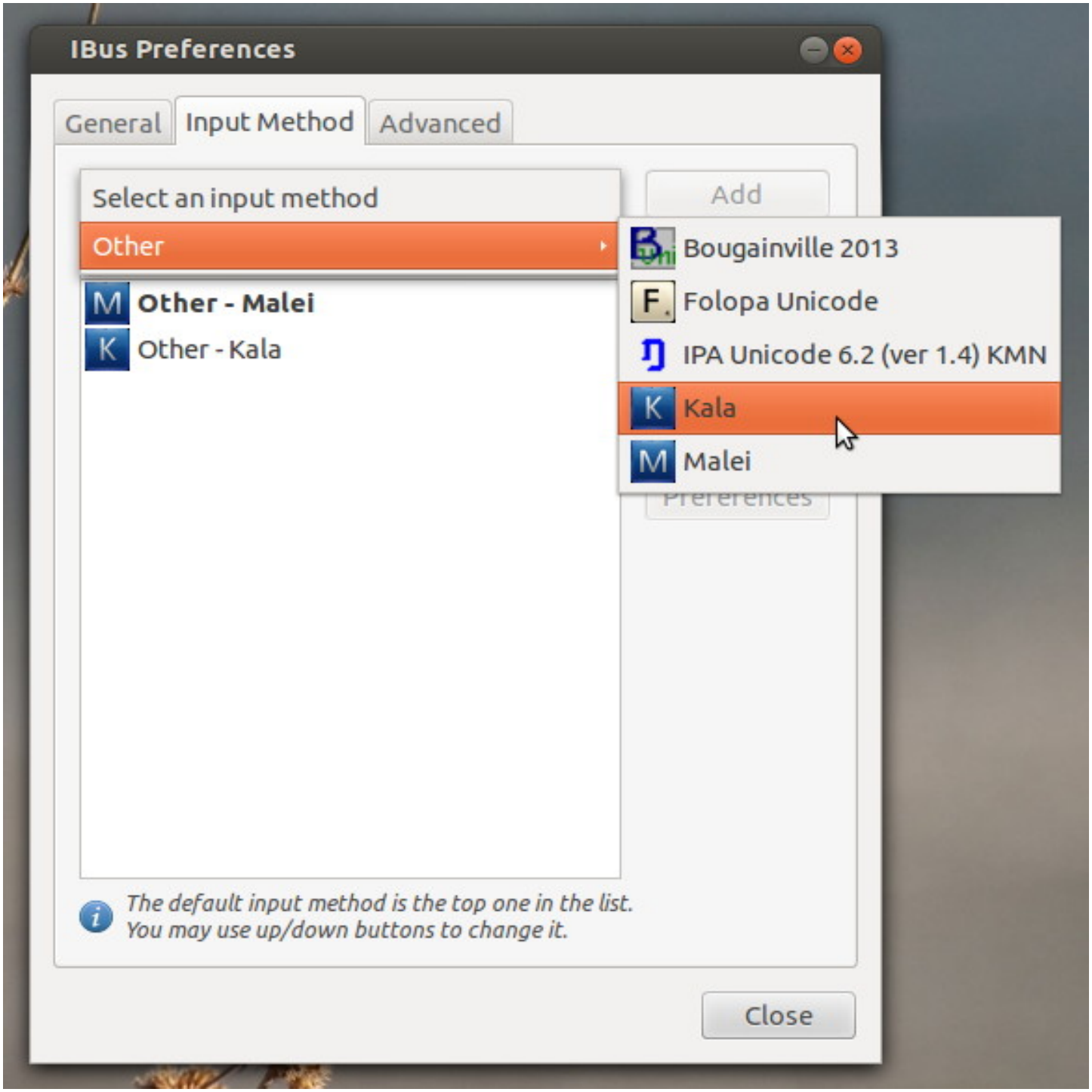
Start by a right click on the "keyboard" icon on the toolbar. Select "Preferences"

Normally a tabbed dialog box will appear under the "General" tab:



Here one can change the default "hot key" sequences, such as keys changing to the next keyboard input method (when one has more than one keyboard) and for simply turning "on" or "off" KMFL. Normally, one can ignore this dialog.

Next the technician will want to specify the input methods and that is on the next tab "input method":



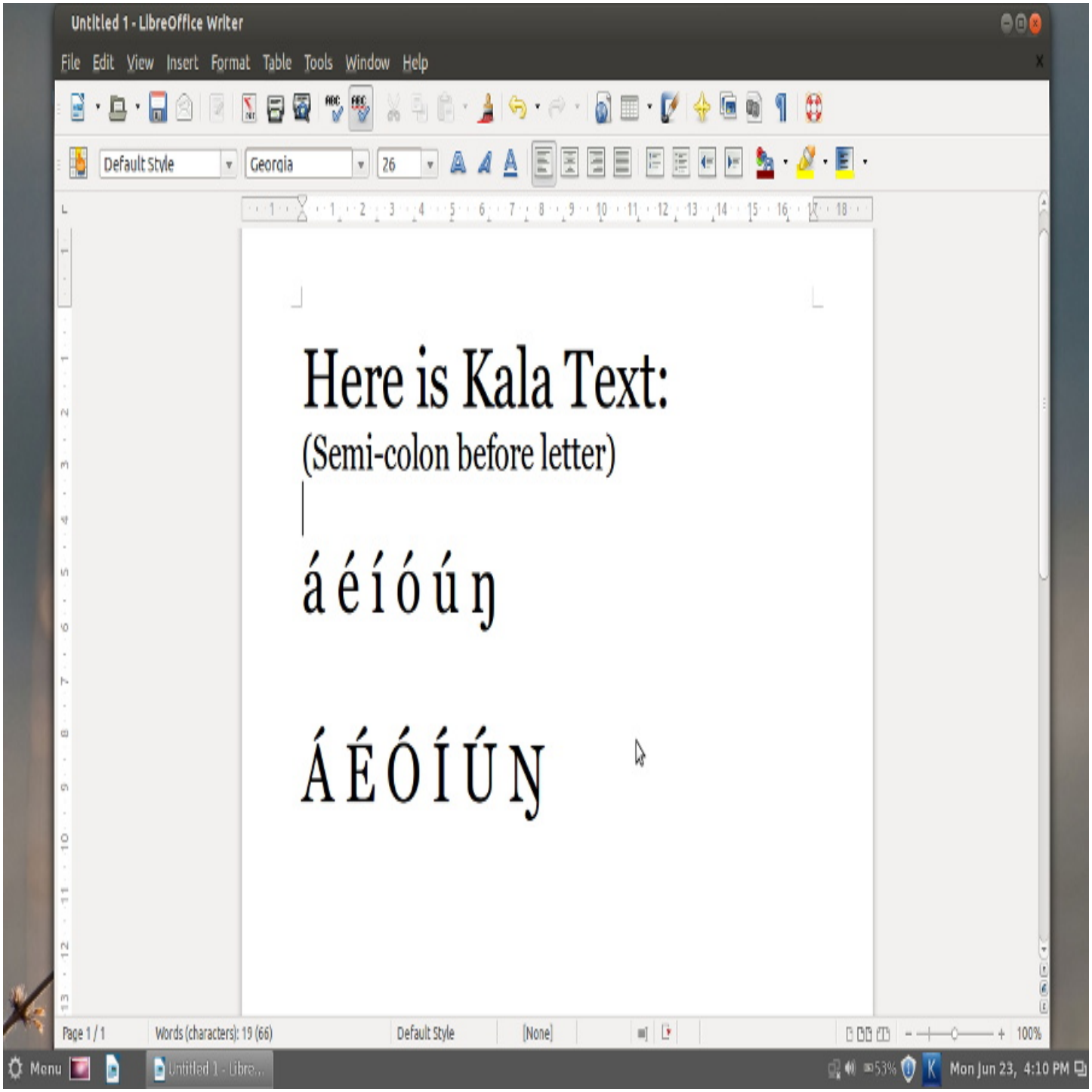
Here we are adding the new Kala keyboard to the list of possible keyboards on the system. The result shows up in the new list of possible keyboards.

Using the Keyboard

There is the possibility (first time use) that the Technician needs to "reset" by a right click on the keyboard icon on the toolbar. Also, if the system is not running well at this point, then consider logging out as the user and then re-log in, as the user.

Now there is one final point, that might lead to frustration at first. The KMFL system DOES NOT operate in environments where it does not make sense to use it. This means that a typical text editor or word-processor, like Libre Office Writer must be open first on the desktop for KMFL to actually run.

Open Libre Office Writer as an example:



Note with the combination (left shift key), one can rotate through the available keyboards (input methods) and their icon is displayed in the toolbar (lower right corner here) accordingly. Therefore you know the "mode" that KMFL is presently running. In the above example, the "M" blue icon is displayed, and this is the icon that was placed in the icon folder previously. will return to a "normal" keyboard for the system, mostly likely English.

Special Note: Some users may not like the shape of certain characters. Try a different font. For example here, the capitol "eng" character has a more angled appearance in the font called "Georgia". Under "Arial" one finds a more rounded capital "eng" shape, which might seem "ugly" to the end user.

Troubleshooting a Paratext Installation

Paratext stores some of its setup and registration information in certain hidden folders within the user's home directory. On rare occasions a Technician may need to go into one of the following Paratext directories and manually adjust some settings to do one of the "rescue operations" listed below:

1. Paratext can't find the ParatextProjects folder.

Paratext may not be able to find its ParatextProjects folder in certain rare situations. In one situation, a user tries to run Paratext, but Paratext gives an error prompt saying that it cannot locate the ParatextProjects folder. Instead, Paratext asks the user to

indicate the location of its ParatextProjects data folder. Some users may have no idea what the problem is and are not able to handle this situation. Thus they become locked out of Paratext and are not able to continue their work. This situation can happen for one of the following reasons:

a. The user accidentally deleted or moved the ParatextProjects folder from the Home directory

The solution in the case of accidentally moving the ParatextProjects folder would be to look in other nearby folders within the user's Home directory to see if the ParatextProjects folder was accidentally moved to a different location. Sometimes computer users can accidentally drag a folder into an adjacent folder when they are trying to double-click on a folder. You should search for the ParatextProjects folder on the user's machine (use the search icon in Nemo). If you find the ParatextProjects folder in a different location, the remedy would be to simply move it back to the user's Home folder. If the ParatextProjects folder was accidentally deleted, the remedy would be to restore it from a recent backup of the user's Paratext data. The user should be backing up their data at least daily using wasta-backup, right?

b. Paratext is confused about the location or spelling of its ParatextProjects folder.

It is also possible that Paratext may have stored a faulty folder location for its ParatextProjects folder. In this case you can check the contents of an xml file to see where Paratext expects to find its ParatextProjects folder. This xml file can be found at the following location:

~/.config/paratext/registry/LocalMachine/software/scrchecks/1.0/settings_directory/values.xml

Edit this values.xml file and either remove the string reference, or correct it to point to the actual location.

2. The Paratext Registration information is incorrectly stored.

If Paratext won't start up because it has the wrong or corrupted Registration Information, you can remove the existing Registration information (to force it to ask for the registration info on startup) or change it by editing the information in the following xml file:

~/.local/share/Paratext73/RegistrationInfo.xml

This file contains all the registration information including the Paratext user's Name, the Registration Code, the user's Email, and the SupporterName.

Preparing and Updating a Wasta-Offline External Hard Drive Mirror

The Full Wasta-Offline external USB hard drive mirror can be a valuable asset for users of Wasta-Linux who are working in locations that do not have access to the Internet for obtaining software updates. This small, light-weight USB drive contains the full wasta-offline software mirrors for the **Ubuntu**, **Linux Mint**, **SIL**, **LibreOffice** and **Wasta-Linux** distributions. The Ubuntu mirrors include all **LTS** packages for both **Precise** and **Trusty**. The Linux Mint mirrors include the **LTS** packages for both **Maya**, **Qiana**, **Rebecca**, **Rafaela**, and **Rosa**. Both **32-bit** and **64-bit** packages are included for all repos. All together this represents approximately **430GB** of data on this USB hard drive.

As of April 2016, the mirrors and repositories contained on this hard drive are:

Mirror	Repos
archive.canonical.com	partner
archive.ubuntu.com	main multiverse restricted universe
extra.linuxmint.com	main
extras.ubuntu.com	main
packages.linuxmint.com	backport import main upstream
packages.sil.org	main
ppa.launchpad.net/libreoffice/libreoffice-4-2	main
ppa.launchpad.net/libreoffice/libreoffice-4-4	main
ppa.launchpad.net/libreoffice/libreoffice-5-0	main
ppa.launchpad.net/libreoffice/libreoffice-5-1	main
ppa.launchpad.net/wasta-linux/wasta	main
ppa.launchpad.net/wasta-linux/wasta-apps	main
security.ubuntu.com	main multiverse restricted universe

Once this full Wasta-Offline drive is available, it can be maintained from a location that has the fastest, most affordable Internet access, and then periodically sent out to regions where it can be used to update the software of the Wasta-Linux computers in those regions quickly and without further expenses - apart from the Internet.

The scripts and associated files that are used to prepare and update a Wasta-Offline mirror may be downloaded from their GitHub

site by opening a terminal and executing the following command:

git clone https://github.com/pngbill-scripts/wasta-scripts.git[36]

which will clone the script files and documentation into a wasta-scripts directory. The repository containing these wasta-offline script files are already included with a USB external hard drive supplied by Bill Martin, but are provided at GitHub for anyone who wants to create a full wasta-offline mirror from scratch. Also any updates or bug fixes to the documentation will be made available here in the future, and to the scripts at their home on GitHub.

Wasta-Offline How To Procedures

How to update your Linux computer's software with the "Full Wasta-Offline Mirror for Precise & Trusty"

This "How to" tells how to use a "Full Wasta-Offline Mirror" USB hard drive together with the wasta-offline program to update the software of a Wasta-Linux computer
See the illustrated guide here[37] for an illustrated tutorial for using the Wasta-Offline program.
Here is a basic outline of the steps:

1. Simply plug this drive into any Wasta-Linux computer. The wasta-offline program should start automatically in about 10 seconds.
2. Enter the computer user's password and click OK.
3. When the initial "wastaOffline" dialog appears leave the default selection "OFFLINE ONLY" selected and click OK.
4. Two successive progress dialogs will appear while wasta-offline is "Adding wasta Offline apt-mirror repositories..." and "Updating software lists..." They will work from 30 seconds to 2 minutes, and close automatically when finished.
5. The "wasta Offline: Ready" dialog will appear. You will leave this window open while using other programs such as "Update Manager" or "Synaptic Package Manager" to complete any software updates or installs. No Internet connection will be used as this is an entirely "offline" process.
6. Start up the Update Manager by clicking on the Update Manager's icon in the panel or by selecting it from the Administration menu.
7. Enter the computer user's password and click OK. The Update Manager's main window appears.
8. If Update Manager's main window shows that software updates are available, click on the "Install Updates" button on the tool bar. The updates will then be downloaded (from the external hard drive to the computer - not from the Internet), and then they will be installed. The process may take a few minutes depending on the number of updates.
9. When the "Update is complete" message appears, click on the "Close" button. The Update Manager automatically closes.
10. If you are not going to install any new software, you can click on the "Finished" button back on the "wasta Offline: Ready" message window. Before wasta-offline ends, it will silently point your software sources back to their previous repositories.
11. If you want to install new software from the external hard drive mirror, leave the "wasta Offline: Ready" window open.
12. To install new software from the external hard drive, start up the Synaptic Package Manager by selecting it from the Administration menu.
13. Enter the computer user's password and click OK. The Synaptic Package Manager's main window appears.
14. Click on the "Search" button on the tool bar, and enter the name (or part of a name) of the software you want to install. After a few seconds you should see a list of software packages that were found matching your search criteria.
15. If the software you want to install is listed, click in the check box next to the software package, and select the "Mark for Installation" option. A tick mark will appear in the check box. If a dialog appears asking "Mark additional required changes?" click on the "Mark" button.
16. Click on the "Apply" button in the main tool bar. Another dialog may open asking "Apply the following changes?". Click on the "Apply" button on this dialog. The package manager then downloads the software program from the external hard drive to the computer - not the Internet, and the new software will be installed. After the installation has finished the package manager will show the program in its main window listed with the check box showing it as installed (green filled check box).
17. Close the Synaptic Package Manager.
18. When all updates and installs are finished, remember to click on the "Finished" button on the "wasta Offline: Ready" message window, to close it. Before wasta-offline ends, it will silently point your software sources back to their previous repositories.

How to update a USB external Drive and bring it up to date - synchronizing it with a master mirror (using the sync_Wasta-Offline_to_Ext_Drive.sh script)

This "How to" tells how to synchronize a slightly out-of-date Wasta-Offline Mirror that resides on an external USB drive up-to-date against an up-to-date master mirror that resides on a local computer:

1. Plug the slightly out-of-date Full Wasta-Offline Mirror for Precise and Trusty into the USB port of the computer that has an up-to-date copy of the Wasta-Offline Mirror.
2. After about 10 seconds the wasta-offline program will start automatically (if the computer is running wasta-linux), and

display a dialog asking for your password. This time click on "Cancel" to close the dialog and stop the wasta-offline program from running. We don't need the wasta-offline program running while we update the mirror (although it causes no problem if it is running).

3. Open the File Manager (Nemo) and locate the external hard drive. It should be called "LM-UPDATES" under the "Devices" list in Nemo. Click on "LM-UPDATES" to highlight the drive and display its contents in the right-hand pane of Nemo.
4. In the root directory of the LM-UPDATES drive you should see a script there that is called: sync_Wasta-Offline_to_Ext_Drive.sh.
5. Highlight the sync_Wasta-Offline_to_Ext_Drive.sh script and press Enter (or double-click on the sync_Wasta-Offline_to_Ext_Drive.sh file). Select the "Run in Terminal" button. The script will open a terminal window and start running. If the script starts running in a terminal window skip to step 6 below.

Note: If the script opens in the gedit editor instead of asking you if you want to "Run in Terminal", you can make the following adjustment in the Nemo settings:

- Quit the gedit editor, and start the Nemo file manager (click the "Files" icon on the panel)
- Select "Preferences" from Nemo's "Edit" menu.
- Select the "Behavior" tab at the top of the Preferences dialog.
- Under the "Executable Text Files" section, select the "Ask each time" button.
- Click on the "Close" button. Now when you double-click on an executable script file, Nemo should give you the option of "Run in Terminal".

If, after following sub-steps above, Nemo still opens the script in the gedit editor instead of offering to "Run" it, it may be that the script is not set to have executable permissions.
To check and set permissions on the script, you can make the following adjustment to the script file's permissions:

- Start the Nemo file manager (click the "Files" icon on the panel)
- Navigate to the device or folder containing the script that you want to check.
- Right-click on the script's name in the list, and select "Properties".
- In the "Properties" dialog select the "Permissions" tab at the top of the dialog.
- Make sure there is a tick in the check box to the right of "Execute" that says "Allow executing file as program".
- Click on the "Close" button. Now when you double-click on an executable script file, Nemo should give you the option of "Run in Terminal".

6. The script will ask you for your user password. Type the password (blindly since it doesn't echo what you type) and press Enter.
7. The sync_Wasta-Offline_to_Ext_Drive.sh assumes by default that the master mirror on the local computer is located at: /data/master/wasta-offline/apt-mirror, and the mirror on the external USB drive is at: /media/LM-UPDATES/wasta-offline/apt-mirror (if running Precise 12.04 version of Wasta-Linux), or the mirror on the external USB drive will be at: /media/$USER/LM-UPDATES/wasta-offline/apt-mirror (if running Trusty 14.04 version of Wasta-Linux. If your mirror locations are not at these defaults, you will need to launch the sync_Wasta-Offline_to_Ext_Drive.sh script from the command line and provide it with parameters that specify the source and destination apt-mirror directories used in your individual configuration. The parameters should point to the wasta-offline directory of the source and destination mirrors.
8. Once the sync_Wasta-Offline_to_Ext_Drive.sh script is running, it will check timestamp of the mirror on the external USB drive and compare it with the timestamp of the master local mirror. Since the master mirror is presumed to be more up-to-date, you will be asked to confirm whether you want the mirror at the external USB drive destination to be updated from the master mirror's NEWER version:

```
Comparing time stamps of the destination and source mirrors...
  Timestamp of mirror at destination is: 1415071069
  Timestamp of mirror at source is: 1415059606
An OLDER copy of the wasta-offline mirror already exists!
Replace it with the NEWER mirror from the external hard drive? y/n
Please press the y or n key, or hit any key to abort - countdown 60
```

Here again, the "countdown" will take 60 seconds to count down. If you don't respond within 60 seconds the "y" (yes) response will be assumed if the destination mirror is OLDER or if there is no timestamp on the destination mirror. (In all other situations, a "n" (no) response will be assumed.)

9. Assuming the script continues from the above prompt, it calls rsync to update the mirror files at the external drive destination with those in the master mirror. If the mirrors on external USB drives are synchronized periodically, the updating process goes quickly, usually just a few minutes to complete.
10. When the sync_Wasta-Offline_to_Ext_Drive.sh script finishes the terminal will display these two lines:

```
The sync_Wasta-Offline_to_Ext_Drive.sh script has finished.
FINISHED:  Press to exit...
```

Press the Enter key to end the terminal session and close its window. This needs to be done, otherwise the USB external hard

drive will indicate it is "Busy" when you try to safely remove it.

How to update the "Full Wasta-Offline Mirror for Precise & Trusty" with the latest software updates from the Internet or from the Ukarumpa FTP site (using the update-mirror.sh script)

This "How to" tells how to update the mirror using any Linux computer that has access to the Internet, or access to a local FTP server that has up-to-date versions of the software mirrors mentioned above. The update-mirror.sh script is provided for this purpose. The process of updating this full mirror could involve downloading many
gigabytes of data, so be sure you have cheap access for downloading data from the Internet, or have access to a free FTP server that contains the data. Depending on
the speed of your connection to the Internet (or to the FTP server), the update process could take many hours (or even days). Hence, do the updating process on a
computer that can be left to run unattended for a long time - perhaps overnight is often a convenient time.

1. Plug the Full Wasta-Offline Mirror for Precise and Trusty into the USB port of the computer that has a connection to the server or Internet.
2. After about 10 seconds the wasta-offline program will start automatically (if the computer is running Wasta-Linux and wasta-offline is pre-installed), and wasta-offline will display a dialog asking for your password. This time click on "Cancel" to close the dialog and stop the wasta-offline program from running. We don't need the wasta-offline program running while we update the mirror (although it causes no problem if it is running).
3. Open the File Manager (Nemo) and locate the external hard drive. It should be called "LM-UPDATES" under the "Devices" list in Nemo. Click on "LM-UPDATES" to highlight the drive and display its contents in the right-hand pane of Nemo.
4. In the root directory of the LM-UPDATES drive you should see a script there that is called: update-mirror.sh.
5. Highlight the update-mirror.sh script and press Enter (or double-click on the update-mirror.sh file). Select the "Run in Terminal" button. The script will open a terminal window and start running. If the script starts running in a terminal window skip to step 6 below.

Note: If the script opens in the gedit editor instead of asking you if you want to "Run in Terminal", you can make the following adjustment in the Nemo settings:

- Quit the gedit editor, and start the Nemo file manager (click the "Files" icon on the panel)
- Select "Preferences" from Nemo's "Edit" menu.
- Select the "Behavior" tab at the top of the Preferences dialog.
- Under the "Executable Text Files" section, select the "Ask each time" button.
- Click on the "Close" button. Now when you double-click on an executable script file, Nemo should give you the option of "Run in Terminal".

If, after following sub-steps above, Nemo still opens the script in the gedit editor instead of offering to "Run" it, it may be that the script is not set to have executable permissions. To check and set permissions on the script, you can make the following adjustment to the script file's permissions:

- Start the Nemo file manager (click the "Files" icon on the panel)
- Navigate to the device or folder containing the script that you want to check.
- Right-click on the script's name in the list, and select "Properties".
- In the "Properties" dialog select the "Permissions" tab at the top of the dialog.
- Make sure there is a tick in the check box to the right of "Execute" that says "Allow executing file as program".
- Click on the "Close" button. Now when you double-click on an executable script file, Nemo should give you the option of "Run in Terminal".

6. The script will ask you for your user password. Type the password (blindly since it doesn't echo what you type) and press Enter.
7. The script then checks for the existence of certain files and programs, and will install the apt-mirror program on the computer if it is not already installed.
8. The script will then ask you where it should get software updates from, giving you the following choices to choose from:

```
Where should the Wasta-Offline Mirror get its software updates?
  1) Get software updates from the SIL Ukarumpa local network FTP server.
  2) Get software updates directly from the Internet (might be expensive!)
  3) Get software updates from a custom network path that I will provide.
  4) Quit - I don't want to get any software updates at this time.
Please press the 1, 2, 3, or 4 key, or hit any key to abort - countdown 60
```

The "countdown" will take 60 seconds to count down. If you don't respond within 60 seconds the 4) Quit option will automatically happen and the script will stop without doing anything more. If it quits before you can read the options and decide, just press Enter to finish the script, and begin again at step 5 above. The usual option to select when at Ukarumpa would be "1) Get software updates from the SIL Ukarumpa local network FTP server". You should have a speedy DSL connection to the server,

and the computer's access to the FTP server should be confirmed before you make your selection at this menu, otherwise the script will abort with an error if it cannot access the FTP server.

If you are in a country with cheap/unlimited data access to the Internet, and have a broadband connection, then you would choose 2) Get software updates directly from the Internet. Be forewarned - updating the full mirror may involve downloading anywhere from a few hundred Magabytes to tens of Gigabytes of data and take hours (if not days) to complete - especially if it has been a long time since the full mirror was updated.

If you are in a country/location that has access to a local software mirror that you can access without cost, you might choose option 3) and enter a network path to the mirror. The path you enter will need to be the full path to the directory that contains the mirror repos listed in the chart above. This might take some experimentation to get the path you enter correct.

9. Once you respond with 1, 2, or 3, of the 4 items in the above menu, and the apt-mirror program has started, apt-mirror will continue downloading any updated mirror data until all the data has been downloaded - which may take a long time.
10. When apt-mirror finishes downloading the data, it automatically runs the postmirror.sh script located in the local mirror's /wasta-offline/apt-mirror/var/ directory. This script automatically calls a clean.sh script (located in the same directory), and then prompts the user with this menu of choices:

```
Run the postmirror2.sh script to correct Hash Sum mismatches errors?
  1) No, don't run the script. There are no Hash Sum mismatches (default)
  2) Yes, run the script and get (75MB) of metadata from the Internet, or
  3) Yes, run the script and get (75MB) of metadata from the local FTP site
Please press the 1, 2, or 3 key, or hit any key to abort - countdown 60
```

The "countdown" will take 60 seconds to count down. If you don't respond within 60 seconds the 1) No, don't run the script... selection will automatically happen and the script will not check for "Hash Sum mismatches". Not checking for "Hash Sum mismatches" should be the usual choice - unless you or others have experienced "Hash Sum mismatch" errors when trying to update the software on computers while using the wasta-offline program against this mirror.

11. The script will keep running, and if an external USB drive is attached, the script will check timestamp of the mirror on the external USB drive and compare it with the timestamp of the local mirror that was just updated by apt-mirror. Depending on the timestamps, you will be asked to confirm whether or not you want the mirror at the external USB drive destination to be updated. The information and prompt will look something like this:

```
Comparing time stamps of the destination and source mirrors...
  Timestamp of mirror at destination is: 1415071069
  Timestamp of mirror at source is: 1415059606
An OLDER copy of the wasta-offline mirror already exists!
Replace it with the NEWER mirror from the external hard drive? y/n
Please press the y or n key, or hit any key to abort - countdown 60
```

Here again, the "countdown" will take 60 seconds to count down. If you don't respond within 60 seconds the "y" (yes) response will be assumed if the destination mirror is "OLDER" or if "there is no timestamp" on the destination mirror. In all other situations, a "n" (no) response will be assumed, and automatically selected if the countdown reaches zero.

12. The prompts in 10 and 11 above have countdown times so that a user can allow the most common actions to take place unattended - that is, not checking for "Hash Sum mismatches", and the usual updating of a mirror on an attached external USB drive which will normally be OLDER than the just-updated local master mirror.

If you need to interrupt the script for any reason, you can do so by typing CTRL+C within the terminal session window (you may need to do CTRL+C repeatedly to truly stop the script). If apt-mirror was still running when a CTRL+C interrupt signal is given, the apt-mirror program will stop. You can start up the update-mirror.sh script again at any time and apt-mirror will continue downloading the software changes from the point where it left off. It won't have to re-download any data that has already been downloaded. Hence, you can repeatedly run the updata-mirror.sh script and interrupt it as many times as it takes to get all the available software updates - or choose an alternate menu response that was bypassed by the countdown in 10 or 11 above.

13. When the updata-mirror.sh script finishes the terminal will display these two lines:

```
The updata-mirror.sh script has finished.
FINISHED:  Press to exit...
```

Press the Enter key to end the terminal session and close its window. This needs to be done, otherwise the USB external hard drive will indicate it is "Busy" when you try to safely remove it.

How to Make a Master or Backup Wasta-Offline Mirror by Copying all of the data from the Existing Mirror on the External USB Drive to the local computer (using the make_Master_for_Wasta-

Offline.sh script)

This "How to" tells how to create a full Wasta-Offline master mirror on a local computer (or a backup of the full Wasta-Offline mirror) copying from an existing Wasta-Offline mirror contained on an external USB drive. Making a master mirror is desirable since it can be used as the machine that regularly calls apt-mirror to keep its master mirror up-to-date, and can then be the source mirror from which one or more external drive(s) can be synchronized and brought up to date locally without having to download mirror update afresh for each copy of the mirror. The computer containing the master mirror can be set to automatically download mirror updates from a free local FTP server that has up-to-date versions of the software mirrors mentioned above, or (if cheap "unlimited" Internet is available) from the Internet.

IMPORTANT: Before you create a full Wasta-Offline master mirror, be sure that the computer has enough space available to hold a full mirror. It should have at least 500GB of free space for the mirror (and future updates).
The make_Master_for_Wasta-Offline.sh script is provided for this purpose. Here are the steps:

1. Plug the Full Wasta-Offline Mirror for Precise and Trusty into the USB port of the computer that has a connection to the server or Internet.
2. After about 10 seconds the wasta-offline program will start automatically (if the computer is running wasta-linux), and display a dialog asking for your password. This time click on "Cancel" to close the dialog and stop the wasta-offline program from running. We don't need the wasta-offline program running while we update the mirror (although it causes no problem if it is running).
3. Open the File Manager (Nemo) and locate the external hard drive. It should be called "LM-UPDATES" under the "Devices" list in Nemo. Click on "LM-UPDATES" to highlight the drive and display its contents in the right-hand pane of Nemo.
4. In the root directory of the LM-UPDATES drive you should see a script there that is called: **make_Master_for_Wasta-Offline.sh**. Highlight the make_Master_for_Wasta-Offline.sh script and press Enter (or double-click on the make_Master_for_Wasta-Offline.sh file). Select the "**Run in Terminal**" button. The script will open a terminal window and start running. If the script starts running in a terminal window skip to step 5 below.

Note: If the script opens in the gedit editor instead of asking you if you want to "Run in Terminal", you can make the following adjustment in the Nemo settings:

- Quit the gedit editor, and start the Nemo file manager (click the "Files" icon on the panel)
- Select "Preferences" from Nemo's "Edit" menu.
- Select the "Behavior" tab at the top of the Preferences dialog.
- Under the "Executable Text Files" section, select the "Ask each time" button.
- Click on the "Close" button. Now when you double-click on an executable script file, Nemo should give you the option of "Run in Terminal".

If, after following sub-steps above, Nemo still opens the script in the gedit editor instead of offering to "Run" it, it may be that the script is not set to have executable permissions. To check and set permissions on the script, you can make the following adjustment to the script file's permissions:

- Start the Nemo file manager (click the "Files" icon on the panel)
- Navigate to the device or folder containing the script that you want to check.
- Right-click on the script's name in the list, and select "Properties".
- In the "Properties" dialog select the "Permissions" tab at the top of the dialog.
- Make sure there is a tick in the check box to the right of "Execute" that says "Allow executing file as program".
- Click on the "Close" button. Now when you double-click on an executable script file, Nemo should give you the option of "Run in Terminal".

5. The script will ask you for your user password. Type the password (blindly since it doesn't echo what you type) and press Enter.
6. If there is no master mirror already located at /data/master/wasta-offline/apt-mirror/ the script calls another script called sync_Wasta-Offline_to_Ext_Drive.sh with appropriate parameters so that it copies the full mirror from the external USB drive to that /data/master/wasta-offline/apt-mirror/ location on the local computer, creating the master mirror tree at that /data/master folder off the root directory.

If a mirror exists at the /data/master/wasta-offline/apt-mirror location, the timestamps of the mirrors will be checked and you will be told whether the destination mirror is "Newer", "Older", or the "Same" as the mirror being copied from. Depending on the timestamps, you will be asked to confirm whether you want the mirror at the external USB drive destination to be updated or not:

```
Comparing time stamps of the destination and source mirrors...
   Timestamp of mirror at destination is: 1415071069
   Timestamp of mirror at source is: 1415059606
An OLDER copy of the wasta-offline mirror already exists!
Replace it with the NEWER mirror from the external hard drive? y/n[38]
Please press the y or n key, or hit any key to abort - countdown 60
```

Here again, the "countdown" will take 60 seconds to count down to zero. If you don't respond within 60 seconds the "y" (yes) response will be assumed if the destination mirror is OLDER or if there is no timestamp on the destination mirror. In all other situations, a "n" (no) response will be assumed.

Note: If you want the master mirror created at a different location than the default location of /data/master/wasta-offline/apt-mirror/, you can do so by running the make_Master_for_Wasta-Offline.sh script manually from the command line in a Terminal with a call like this (made from the /media/LM-UPDATES directory if using Wasta-Linux Precise 12.04, or from the /media//LM-UPDATES directory if using Wasta-Linux Trusty 14.04):

```
cd /media/LM-UPDATES or: cd /media//LM-UPDATES
bash make_Master_for_Wasta-Offline.sh
```

where the should be the path to the destination mirror tree's apt-mirror directory. For example, if the destination is to be in the **/home/bill/** directory, the would be: **/home/bill/wasta-offline/apt-mirror** and the actual command line command would be (on a Trusty system):

```
cd /media/bill/LM-UPDATES
bash make_Master_for_Wasta-Offline.sh /home/bill/wasta-offline/apt-mirror
```

See the script file itself for how to use other parameter configurations when manually invoking the script from a command line.

7. The script uses rsync to create the master mirror (or update if one already existed). The creation of a new mirror at the destination will take about 4.5 hours to complete regardless of whether you are using a USB 2.0 or USB 3.0 connection.
8. When the make_Master_for_Wasta-Offline.sh and sync_Wasta-Offline_to_Ext_Drive.sh scripts finish the terminal will display these lines:

```
The sync_Wasta-Offline_to_Ext_Drive.sh script has finished.
The make_Master_for_Wasta-Offline.sh script has finished.
FINISHED:  Press to exit...
```

Press the Enter key to end the terminal session and close its window. This needs to be done, otherwise the USB external hard drive will indicate it is "Busy" when you try to safely remove it.

How to clone a new Wasta-Offline Mirror to a New External USB Hard Drive, Copying from a Master Wasta-Offline Mirror (using the sync_Wasta-Offline_to_Ext_Drive.sh script)

This "How to" assumes that you have a master mirror on the local computer and that you want to make a new Wasta-Offline Mirror on a new or empty External USB hard drive.

1. Plug in a new or empty external USB hard drive into the USB port of the local computer that has an existing master copy of Wasta-Offline Mirror on its hard drive (the default location of the master mirror is **/data/master/wasta-offline/apt-mirror**). This new/empty USB hard drive should have at least 500GB of capacity.
2. Open the File Manager (Nemo) on the local computer and locate the wasta-offline mirror. The default location for it would be **/data/master/wasta-offline/apt-mirror/**. Navigate to the base directory of the mirror (in our case the /data/master directory), where you should see its contents displayed in the right-hand pane of Nemo.
3. In the root (/data/master) directory you should see a script there that is called: **sync_Wasta-Offline_to_Ext_Drive.sh**. Highlight the **sync_Wasta-Offline_to_Ext_Drive.sh** script and press Enter (or double-click on the **sync_Wasta-Offline_to_Ext_Drive.sh** file). Select the "Run in Terminal" button. The script will open a terminal window and start running. If the script starts running in a terminal window skip to step 4 below.

Note: If the script opens in the gedit editor instead of asking you if you want to "Run in Terminal", you can make the following adjustment in the Nemo settings:

- Quit the gedit editor, and start the Nemo file manager (click the "Files" icon on the panel)
- Select "Preferences" from Nemo's "Edit" menu.
- Select the "Behavior" tab at the top of the Preferences dialog.
- Under the "Executable Text Files" section, select the "Ask each time" button.
- Click on the "Close" button. Now when you double-click on an executable script file, Nemo should give you the option of "Run in Terminal".

If, after following sub-steps above, Nemo still opens the script in the gedit editor instead of offering to "Run" it, it may be that the script is not set to have executable permissions. To check and set permissions on the script, you can make the following adjustment to the script file's permissions:

- Start the Nemo file manager (click the "Files" icon on the panel)
- Navigate to the device or folder containing the script that you want to check.
- Right-click on the script's name in the list, and select "Properties".
- In the "Properties" dialog select the "Permissions" tab at the top of the dialog.

- Make sure there is a tick in the check box to the right of "Execute" that says "Allow executing file as program".
- Click on the "Close" button. Now when you double-click on an executable script file, Nemo should give you the option of "Run in Terminal".

4. The script will ask you for your user password. Type the password (blindly since it doesn't echo what you type) and press Enter.
5. If the external hard drive is formatted as a FAT, FAT32, or other non-Linux file system, it needs to be formatted to a Linux Ext4 file system to receive a clone of the master mirror, and so the script will search for and display any such eligible USB drives that are mounted on the system, and present the user with a list of such drive(s), and prompt the user to select the number of the USB drive to be used. Normally only one such drive would be listed, but if more such USB drives are mounted they would be listed as 2), 3), etc. The prompt will look similar to this:

```
Seaching for USB storage drive(s) on this computer...
 1) WDC WD50:00LPVX-22V0TT0:/dev/sdd:/media/MyData:MyData:500GB
Type the number of the USB drive to use, or hit any key to abort - countdown 60
```

If the countdown goes to 0 before a response is given, no USB drive will be selected, error messages issued and the sync process will be aborted. If the user does make a selection, the following information and prompt will be displayed:

```
You have selected this USB device mounted at /dev/sdd:
  WDC WD50:00LPVX-22V0TT0:/dev/sdd:/media/MyData:MyData:500GB

WARNING: This USB drive will be formatted - ALL data on it erased. OK (y/n)?
Type 'y' to proceed with formatting, any other key to abort - countdown 60
```

Again, if the countdown goes to 0 before a response is given, the formatting process will be aborted, error messages issued and the sync process will be aborted.

6. Once the external hard drive is deemed eligible to receive the mirror data (has the required Linux file system, and a capacity of at least 500GB), the sync process will proceed.
7. When the **sync_Wasta-Offline_to_Ext_Drive.sh** script finishes the terminal will display these lines:

```
The sync_Wasta-Offline_to_Ext_Drive.sh script has finished.
FINISHED:  Press to exit...
```

Press the Enter key to end the terminal session and close its window. This needs to be done, otherwise the USB external hard drive will indicate it is "Busy" when you try to safely remove it.

How to Fix "Hash Sum mismatch" Errors that Appear When Trying to Use the "Full Wasta-Offline Mirror for Precise & Trusty" external USB Drive (using the update-mirror.sh and postmirror2.sh scripts)

If users get "Hash Sum mismatch" errors when they try to use the Wasta-Offline Mirror to update their computer , it means that some of the mirror's metadata files were not properly synchronized or updated when the apt-mirror program attempted to update the mirror. This situation can usually be rectified by allowing the **postmirror2.sh** script to run at the end of an apt-mirror update. Follow these steps:

1. Plug the Full Wasta-Offline Mirror for Precise and Trusty into the USB port of the computer that has a connection to the server or Internet.
2. After about 10 seconds the wasta-offline program will start automatically (if the computer is running wasta-linux), and display a dialog asking for your password. This time click on "Cancel" to close the dialog and stop the wasta-offline program from running. We don't need the wasta-offline program running while we update the mirror (although it causes no problem if it is running).
3. Open the File Manager (Nemo) and locate the external hard drive. It should be called "LM-UPDATES" under the "Devices" list in Nemo. Click on "LM-UPDATES" to highlight the drive and display its contents in the right-hand pane of Nemo.
4. In the root directory of the LM-UPDATES drive you should see a script there that is called: **update-mirror.sh**. Highlight the **update-mirror.sh** script and press Enter (or double-click on the **update-mirror.sh** file). Select the "Run in Terminal" button. The script will open a terminal window and start running. If the script starts running in a terminal window skip to step 5 below.

Note: If the script opens in the gedit editor instead of asking you if you want to "Run in Terminal", you can make the following adjustment in the Nemo settings:

- Quit the gedit editor, and start the Nemo file manager (click the "Files" icon on the panel)
- Select "Preferences" from Nemo's "Edit" menu.
- Select the "Behavior" tab at the top of the Preferences dialog.
- Under the "Executable Text Files" section, select the "Ask each time" button.
- Click on the "Close" button. Now when you double-click on an executable script file, Nemo should give you the option of

"Run in Terminal".

If, after following sub-steps above, Nemo still opens the script in the gedit editor instead of offering to "Run" it, it may be that the script is not set to have executable permissions. To check and set permissions on the script, you can make the following adjustment to the script file's permissions:

- Start the Nemo file manager (click the "Files" icon on the panel)
- Navigate to the device or folder containing the script that you want to check.
- Right-click on the script's name in the list, and select "Properties".
- In the "Properties" dialog select the "Permissions" tab at the top of the dialog.
- Make sure there is a tick in the check box to the right of "Execute" that says "Allow executing file as program".
- Click on the "Close" button. Now when you double-click on an executable script file, Nemo should give you the option of "Run in Terminal".

5. The script will ask you for your user password. Type the password (blindly since it doesn't echo what you type) and press Enter.
6. At the end of the apt-mirror update, apt-mirror automatically calls the **postmirror.sh** script. This **postmirror.sh** script will give the user the following options:

```
Run the postmirror2.sh script to correct Hash Sum mismatches errors?
 1) No, don't run the script. There are no Hash Sum mismatches
 2) Yes, run the script and get (75MB) of metadata from the Internet, or
 3) Yes, run the script and get (75MB) of metadata from the local FTP site
Please press the 1, 2, or 3 key, or hit any key to abort - countdown 60
```

Select item 2) or 3) - whichever is appropriate for your situation - to download the metadata files afresh to the mirror. Note: This prompt has a timer. If no response is given within 60 seconds, "1) No, don't run the script. There are no Hash Sum mismatches" is automatically selected and the script will end without attempting to correct any Hash Sum mismatches.

If **1** is entered, the script finishes with, "The postmirror2.sh will not be called. Script completed."
If **2** is entered, postmirror2.sh is called without any parameter (Internet use assumed)
If **3** is entered, postmirror2.sh is called with a "ftp" parameter and "ftp" will be interpreted to be equivalent to:
ftp://ftp.sil.org.pg/Software/CTS/Supported_Software/Ubuntu_Repository/mirror/
within the postmirror2.sh script (unless postmirror2.sh is called manually and a different parameter is used for the URL).

7. When the update-mirror.sh, postmirror.sh, and postmirror2.sh scripts finish the terminal will display these lines:

```
The update-mirror.sh script has finished.
FINISHED:  Press to exit...
```

Press the Enter key to end the terminal session and close its window. This needs to be done, otherwise the USB external hard drive will indicate it is "Busy" when you try to safely remove it.

8. You should then run the wasta-offline program against the mirror on the USB drive and run the Update Manager, and click on the Update Manager's "Refresh" button. There should no longer be any "Hash Sum mismatch" errors detected.

How to Set Up a Computer to Get Software Updates from the Internet or from the Ukarumpa FTP Site On a Scheduled Daily or Periodic Basis

It is best to schedule the downloading of software updates to be done to a master copy of the Wasta-Offline Mirror that exists on a local computer that can be left running overnight. The "How to Make a Master or Backup Wasta-Offline Mirror..." instructions above tell how to create such a master copy of the Wasta-Offline Mirror. Once a master mirror exists on a computer, updates to that mirror can be scheduled as a cron job. When apt-mirror is installed on a computer, it creates a sample cron job template file at: **/etc/cron.d/apt-mirror**. Here is how to schedule such a cron job:

1. Open a Terminal window (**CTRL+ALT+T**).
2. Change to the **/etc/cron.d/** directory: **cd /etc/cron.d**
3. Edit the apt-mirror file as root: **sudo gedit apt-mirror**
4. The content of the apt-mirror file will initially look like this:

```
#
# Regular cron jobs for the apt-mirror package
#
#0 4     * * *   apt-mirror      /usr/bin/apt-mirror > /var/spool/apt-mirror/var/cron.log
```

Remove the **#** character in front of the last line to activate that line. The **4** represents 4:00am as the time to run apt-mirror. That would be OK if you regularly leave the computer running overnight. Alternately, you may want to change the **4** to a different time such as **9** which will represent 9:00am - a time more suitable for a computer that is generally running during the day but not

at night.

5. After doing the above edits to the apt-mirror file, save the file and quit gedit.
6. Note: When updated and synced with the supplied scripts, chown calls are made to set the Wasta-Offline Mirror's owner:group to apt-mirror:apt-mirror. This is necessary for the cron job to work on schedule.
7. With a cron job updating a master Wasta-Offline Mirror on a regular schedule, it is then a fairly quick job to synchronize one or more external USB drive(s) to bring them up-to-date, following the steps in the "How to" above called: "How to update a Wasta-Offline Mirror on a USB external Drive and bring it up to date - synchronizing it with a master mirror".
8. You should regularly check the apt-mirror cron job's log to make sure that it is doing its updates as scheduled. The log file is at: **/var/spool/apt-mirror/var/cron.log** If you double click on the cron.log file at the above location, it should open in gedit and you can check the date of the last run of apt-mirror, and see the output of the apt-mirror run as well as the output of the postmirror.sh script - checking for any reported errors.

How to Tweak the update-mirror.sh Script to Add or Remove a Repository from the apt-mirror Updating Process

The content of an apt-mirror generated software mirror is determined by the contents of a configuration file called mirror.list that is located at: **/etc/apt/mirror.list** on the computer that runs the apt-mirror program. The provided script called update-mirror.sh takes care of adjusting the computer's mirror.list file - ensuring that it points to the appropriate external software mirrors. Where the mirror.list file's software sources point to, depend on the user's choice at the following menu when the update-mirror.sh script is run:

```
Where should the Wasta-Offline Mirror get its software updates?
  1) Get software updates from the SIL Ukarumpa local network FTP server.
  2) Get software updates directly from the Internet (might be expensive!)
  3) Get software updates from a custom network path that I will provide.
  4) Quit - I don't want to get any software updates at this time.
Please press the 1, 2, 3, or 4 key, or hit any key to abort - countdown 60
```

See the "How to update the 'Full Wasta-Offline Mirror for Precise & Trusty' with the latest software updates from the Internet or from the Ukarumpa FTP site (using the update-mirror.sh script)" above for the details.

This "How to" tells how to tweak the script to remove or add repositories that compose the "full" Wasta-Offline Mirror.

The **update-mirror.sh** script actually uses a bash function called **generate_mirror_list_file ()** which is located in the **bash_functions.sh** file included with the other scripts in this distribution. It is in the **generate_mirror_list_file ()** function that tweaks can be made to determine what content gets generated into the computer's **mirror.list** file. That **generate_mirror_list_file ()** function utilizes a "here-document" form of redirection that tells the shell to read from the current script's source-text until the line containing EOF is seen. This effectively duplicates the textual content of that segment of the function between its "**cat < < EOF...**" statement and its final
"**EOF**" line copying that into a fresh version of the mirror.list file on the user's computer. Variables within the text are expanded into their current text values before the
mirror.list file is written out to disk. One can control the active content of one's mirror.list file by adding new software repositories to this "here-document" template, or
by commenting out existing software repositories. For example, here are two lines from the middle of that "here-document":

```
deb-amd64 $1packages.sil.org/ubuntu precise main
deb-i386 $1packages.sil.org/ubuntu precise main
```

The **$1** variable will be substituted with a string of characters that was used as a parameter when the generate_mirror_list_file () function was called from the update-mirror.sh script. That parameter was either **http://** or the much longer string: **ftp://ftp.sil.org.pg/Software/CTS/Supported_Software/Ubuntu_Repository/mirror/**. If one wanted to remove this repository from the mirror, one could put a # character at the beginning of each of the above two lines. That effectively "comments out" those two lines from being read by the apt-mirror program.

Including more repositories is fairly simple. You just add them to the "here-document" or uncomment any that are there but are currently commented out. For example, there are two lines (for each of the precise and trusty) sections of the template that refer to the virtualbox repositories, but they are initially commented out when they are written to the user's mirror.list file. If a user desires to include the virtualbox repositories in the mirror, they could modify the lines in the "here-document" by removing the initial **#** characters ("uncommenting" them), so that instead of being:

```
#deb-amd64 $1download.virtualbox.org/virtualbox/debian precise contrib
#deb-i386 $1download.virtualbox.org/virtualbox/debian precise contrib
```

they would become:

```
deb-amd64 $1download.virtualbox.org/virtualbox/debian precise contrib
```

```
deb-i386 $1download.virtualbox.org/virtualbox/debian precise contrib
```

in the "here-document", and would consequently be activated when written out to the user's mirror.list file on-the-fly at the time the update-mirror.sh script is run.

Note: If you remove repositories by commenting them out of the resulting mirror.list configuration file, those repositories will be purged from the mirror itself, and if you later want those repositories back in the mirror, they would have to be downloaded again.

---

[1] https://lingtran.net/mailto:brian_chapaitis@sil.org

[2] http://itconnect.info/group/linux-desktop/page/wasta-linux

[3] https://lingtran.net/mailto:cts-netengr4@sil.org.pg

[4] https://lingtran.net/

[5] https://lingtran.net/#Creating_a_bootable_memory_stick

[6] https://lingtran.net/The+Consultant+-+Advisor#Boot_to_a_%22Live_Session%22_of_Wasta_Linux

[7] https://lingtran.net/The+Consultant+-+Advisor#Ensure_your_Computer's_Hardware_Functions_Properly_Under_Wasta_Linux

[8] https://lingtran.net/The+Consultant+-+Advisor#Deciding_How_to_Install_Wasta_Linux

[9] http://tiki.lingtransoft.info/The+Consultant+-+Advisor#Installing_Dual_Boot

[10] http://tiki.lingtransoft.info/The+Consultant+-+Advisor#Installing_as_only_OS

[11] http://tiki.lingtransoft.info/The+Consultant+-+Advisor#Installing_as_only_OS

[12] http://tiki.lingtransoft.info/The+Consultant+-+Advisor#Installing%20Wasta%20as%20a

[13] http://tiki.lingtransoft.info/The+Consultant+-+Advisor#Installing_Dual_Boot

[14] http://tiki.lingtransoft.info/The+Consultant+-+Advisor#Installing_Dual_Boot

[15] https://lingtran.net/The+Consultant+-+Advisor#Installing_Wasta_as_the_Primary_and_Only_Operating_System

[16] https://lingtran.net/The+Consultant+-+Advisor#Re-Installing_Wasta_Over_an_Existing_Wasta_System

[17] http://itconnect.info/group/linux-desktop/page/wasta-linux-customizing

[18] http://itconnect.info/group/linux-desktop/page/wasta-linux-launchpad-setup

[19] http://itconnect.info/group/linux-desktop/page/wasta-linux-update-package

[20] https://lingtran.net/The+Consultant+-+Advisor#Consultant-AddinganIcontotheDesktop

[21] http://linux.lsdev.sil.org/wiki/index.php/Installing_KMFL_on_Ubuntu

[22] https://lingtran.net/%C3%A1

[23] https://lingtran.net/%C3%A9

[24] https://lingtran.net/%C3%AD

[25] https://lingtran.net/%C3%B3

[26] https://lingtran.net/%C3%BA

[27] https://lingtran.net/%C3%81

[28] https://lingtran.net/%C3%89

[29] https://lingtran.net/%C3%8D

[30] https://lingtran.net/%C3%93

[31] https://lingtran.net/%C3%9A

[32] https://lingtran.net/;

[33] https://lingtran.net/%C5%8B

[34] https://lingtran.net/%C5%8A

[35] http://help.keyman.com/developer/8.0/docs/tutorial_keyboard.php

[36] https://github.com/pngbill-scripts/wasta-scripts.git

[37] http://tiki.lingtransoft.info/tiki-index.php?page_ref_id=253#Consultant-SoftwareImprovements:HowtoUpdateSoftwareeasilybyMemoryStick

[38] https://lingtran.net/y/n