UPDATE (3/3/2012): The how-to has been updated to reflect changes for Backtrack R2. They were very minor and using the previous method still works. The only real change is:

- lvm2 is now part of the ISO. That means we no longer have to use apt-get to install it. However, we still need to install hashalot, so it doesn't save us a step.
- Added a note at the end about using dd to backup your install per a very good suggestion by Richard in comment 241.

UPDATE: This update has been tested with BT5R1 and works as is. Before we get started, here are a few housekeeping items:

- There is a PDF version of this article available here. (Will be available soon.)
- Finally, if you want to be notified of updates to this page, subscribe to my RSS feed here.

I put quotes around full in the title because technically the whole disk isn't encrypted. We use LVM and the native encryption routines included in Ubuntu to encrypt all partitions except for a small boot partition that never contains any data.

This is a fairly involved process, but I have done my best to document each detail. Please let me know if I missed anything or you have any questions. I can be reached via the contact form on the 'About' page of this website or via the comments below.

I strongly recommend you read through this guide at least once before starting.

I will be making a PDF available in the near future.

As in all my how-tos, user entered text is **bold** and comments are preceded by a # sign and generally not part of the output of a command. Finally, a couple of posts from the Ubuntu Community Documentation site were instrumental in getting this working.

https://help.ubuntu.com/community/EncryptedFilesystemOnIntrepid https://help.ubuntu.com/community/EncryptedFilesystemLVMHowto

WARNING: Before you start, please be aware that you can cause the system you are using to build this with to not boot correctly. During the install process below there is a warning about indicating where you want the boot loader to be installed. Be very careful at this point. First we are going to need some stuff.

Tools and Supplies

1. A USB thumbdrive for the install - minimum capacity 16GB. Actually, you can squeeze this onto an 8GB drive, but you are out of room at that point.
2. A Backtrack 5 DVD or an additional USB thumbdrive (minimum 2GB, must be Backtrack 5)
3. Optional: UNetbootin - A tool to transfer an iso image to a USB drive.
4. Working internet connection once Backtrack 5 is booted.

Let's get started!

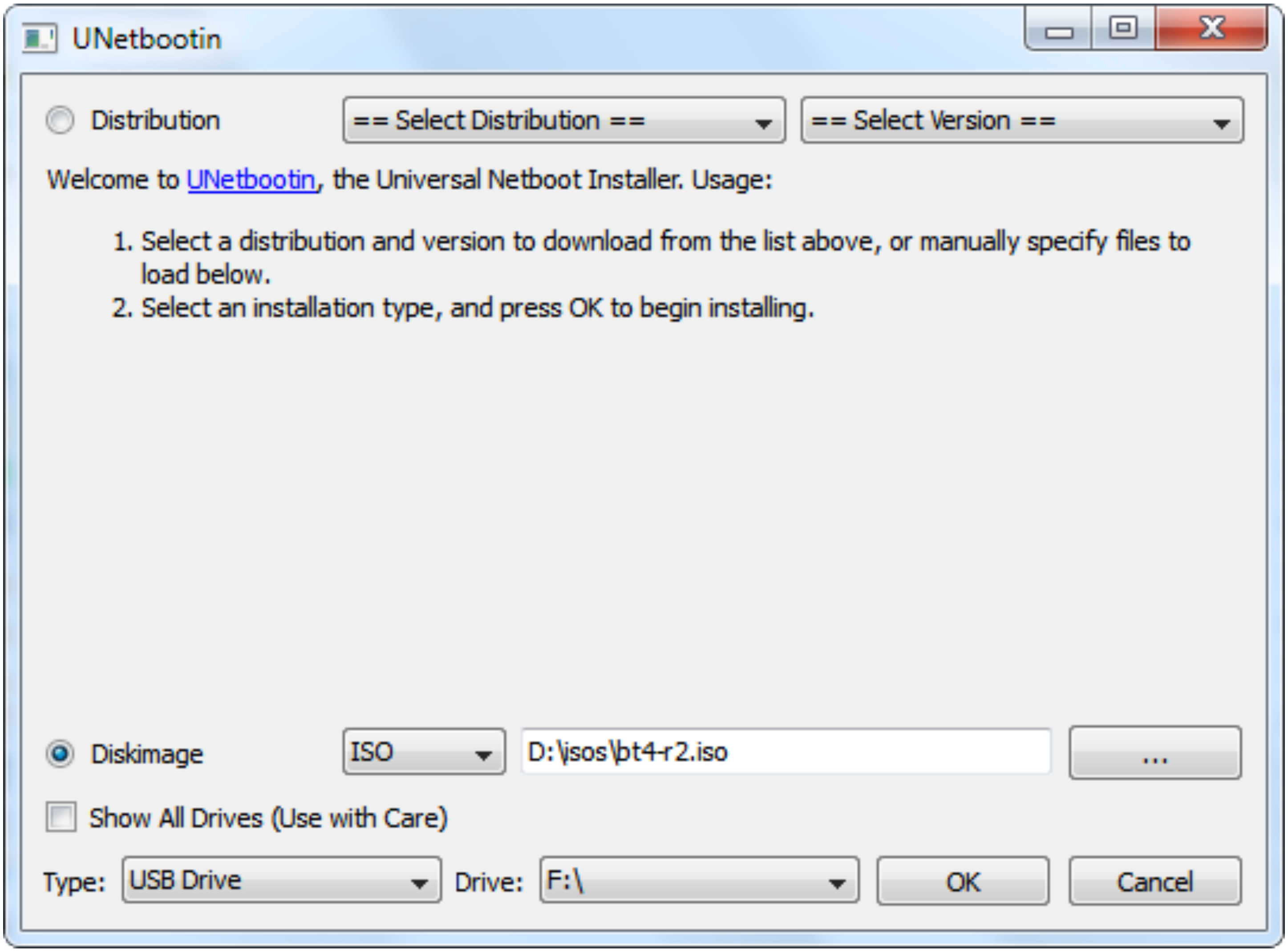
First, we need to grab a copy of the Backtrack 5 ISO.

Backtrack 5 Download Page

For my tests, I used the 64-bit Gnome build. I have not tested this how-to with all versions of Backtrack 5, but they should all behave similarly with the possible exception of the ARM build. I have no experience with Backtrack on that platform.

Now that we have the goods in hand, we can get to cooking.

This tutorial is based on booting Backtrack 5 first. That means that you need some form of bootable Backtrack media. It can be a create the thumb drive. Below is a screenshot of using UnetBootin to install Backtrack, version 4 in this case, on a USB drive. Again, you will need version 5. I'm just lazy right now :)



○ Distribution

Welcome to UNetbootin, the Universal Netboot Installer. Usage:

1. Select a distribution and version to download from the list above, or manually specify files to load below.
2. Select an installation type, and press OK to begin installing.

● Diskimage ISO D:\isos\bt4-r2.iso

□ Show All Drives (Use with Care)

Type: USB Drive Drive: F:\

It is as simple as selecting the image we want to write to the USB drive, the drive to write it to, and then clicking the 'OK' button. Warning: Make sure you pick the correct destination drive. You don't want to shoot yourself in the foot. :)

Partitioning

The first step is the physical partitioning of the drive. Boot up Backtrack from your DVD or USB drive. If you boot with the default menu item "Backtrack Text", you will not need to start networking as it will have started automatically. You can verify that networking is up and running by executing:

```
ifconfig
```

and checking that your interface is up and has an IP address assigned. If networking isn't configured, the following commands will start it.

```
/etc/init.d/networking start
```

We do need to start the graphical interface.

```
startx
```

We will also need to figure out which drive is our target drive. The following command will show the drives available and you can determine from that which is the new USB drive. Open a terminal windows and execute the following.

```
dmesg | egrep hd.\|sd.
```

We need to physically partition the target drive as follows:

1. The first partition needs to be a primary partition, 500 MB in size, set to type ext4. Also remember to make this partition active when you are creating it. Otherwise you might have some boot problems.
2. The rest of the drive should be configured as an extended partition and then a logical partition created on top of it.

Below are the steps to take to get the drive partitioned. A '# blah blah' indicates a comment and is not part of the command and user typed commands are **bolded**. One note, we will need to delete any existing partitions on the drive. Also, the cylinder numbers below are specific to my test machines/thumb drives, yours may be different. Finally, if you are using this how-to to install to a internal hard drive, you probably want to add a swap partition.

```
fdisk /dev/sdb # use the appropriate drive letter for your system

# delete existing partitions. There may be more than one.
Command (m for help): d
Partition number (1-4): 1

# create the first partition
Command (m for help): n
Command action e   extended p   primary partition (1-4) p
Partition number (1-4): 1
First cylinder (1-2022, default 1): <enter>
Using default value 1 Last cylinder, +cylinders or +size{K,M,G} (1-2022, default 2022): +500M

# create the extended partition
Command (m for help): n
Command action e   extended p   primary partition (1-4) e
Partition number (1-4): 2
First cylinder (66-2022, default 66): <enter>
Using default value 66 Last cylinder, +cylinders or +size{K,M,G} (66-2022, default 2022): <enter>
Using default value 2022

# Create the logical partition.
Command (m for help): n
Command action l    logical (5 or over) p   primary partition (1-4) l
First cylinder (66-2022, default 66): <enter>
Using default value 66 Last cylinder, +cylinders or +size{K,M,G} (66-2022, default 2022): <enter>
Using default value 2022

# Setting the partition type for the first partition to ext3 Command (m for help): t
Partition number (1-4): 1
Hex code (type L to list codes): 83

# Setting the first partition active
Command (m for help): a
Partition number (1-4): 1
Command (m for help): w
```

If you happen to get an error that mentions something like "..the partition table failed with error 16:...", you need to reboot before continuing with the how-to. You might be able to get away with continuing, but there is a good chance you will experience some problems. After rebooting, you will need to re-execute the **startx** command and the **cryptsetup luksOpen** commands.

If you happen to get an error with mentions something like "..the partition table failed with error 22:..." you can run **partprobe** to re-read things. At least, this worked in my case.

It is now time to get a couple additional packages installed that we need for LVM and encryption. First we need to update the local repositories and then install lvm2 and hashalot. Output has been ommitted.

```
# For Backtrack 5 R1 type the following.
apt-get update
apt-get install hashalot lvm2

# For Backtrack 5 R2 we only need hashalot. Type the following.
apt-get update
apt-get install hashalot
```

Our next step is to enable encryption on the logical partition we created above and make it available for use. Before we do that though, there is an optional step we can take if we want to make sure no one can tell where our data is on the drive. It isn't really necessary since anything written will be encrypted, but if we want to be thorough and make sure no one can see where our data even sits on the drive, we can fill the logical partition with random data before enabling encryption on it. This will take some time, as much as a couple hours or more. Execute the following command:

```
dd if=/dev/urandom of=/dev/sdb5
```

The following commands will setup encryption services for the partition and open it for use. There are several ciphers that can be used, but the one indicated in the command is supposed to be the most secure and quickest for Ubuntu 8.10. Please note that the case of the command luksFormat is required.

```
cryptsetup -y --cipher aes-xts-plain --key-size 512 luksFormat /dev/sdb5
WARNING! ======== This will overwrite data on /dev/sdb5 irrevocably. Are you sure? (Type uppercase yes): YES
Enter LUKS passphrase: (enter passphrase) [type passphrase]
Verify passphrase: (repeat passphrase) [type passphase]
Command successful.

cryptsetup  luksOpen /dev/sdb5 pvcrypt
Enter LUKS passphrase: [type passphrase]
key slot 0 unlocked. Command successful.
```

If you should happen to get a "cannot access device" error when trying to perform the **cryptsetup** setup commands above, make sure the USB drive has not been mounted. That can happen sometimes. Now that that's all done, we can create our root and swap partitions using LVM. Again, the commands below will do so. 7.3 GB was the largest I could make my root partition. Play around with it a little and you may be able to make it a bit larger or you may have to make it a bit smaller.

```
pvcreate /dev/mapper/pvcrypt
Physical "volume /dev/mapper/pvcrypt" successfully created

vgcreate vg /dev/mapper/pvcrypt
Volume group "vg" successfully created

lvcreate -n root -l 100%FREE vg
Logical volume "root" created.
```

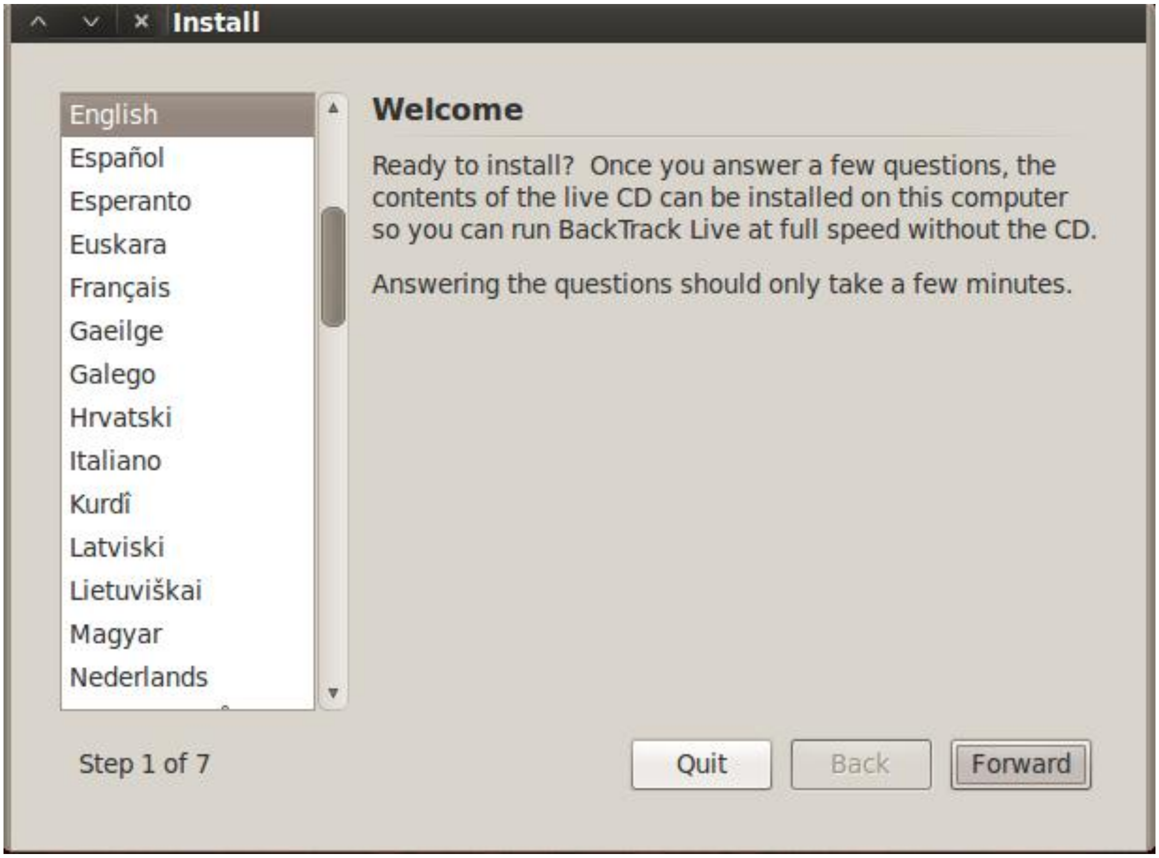
The final step is to format the logical volumes we just created. I have not included the output below for brevity's sake.

```
mkfs.ext4 /dev/mapper/vg-root
```

If you want to try and eek out every last bit of performance and help your flash drive last longer, you can alternatively use the following command to disable journaling on the root partition. I have not tested this yet, but it should work just fine. Remember that this will open you to a greater possibility of unrecoverable drive corruption. Notice that the first letter o is small case and the

```
tune2fs -o journal_data_writeback /dev/mapper/vg-root
tune2fs -O ^has_journal /dev/mapper/vg-root
e2fsck -f /dev/mapper/vg-root
```

Believe it or not, we are finally ready to start installing Backtrack. To do, double-click on the install.sh icon on the desktop. This will start the graphical installer. Select you language of choice and click the 'Forward' button.



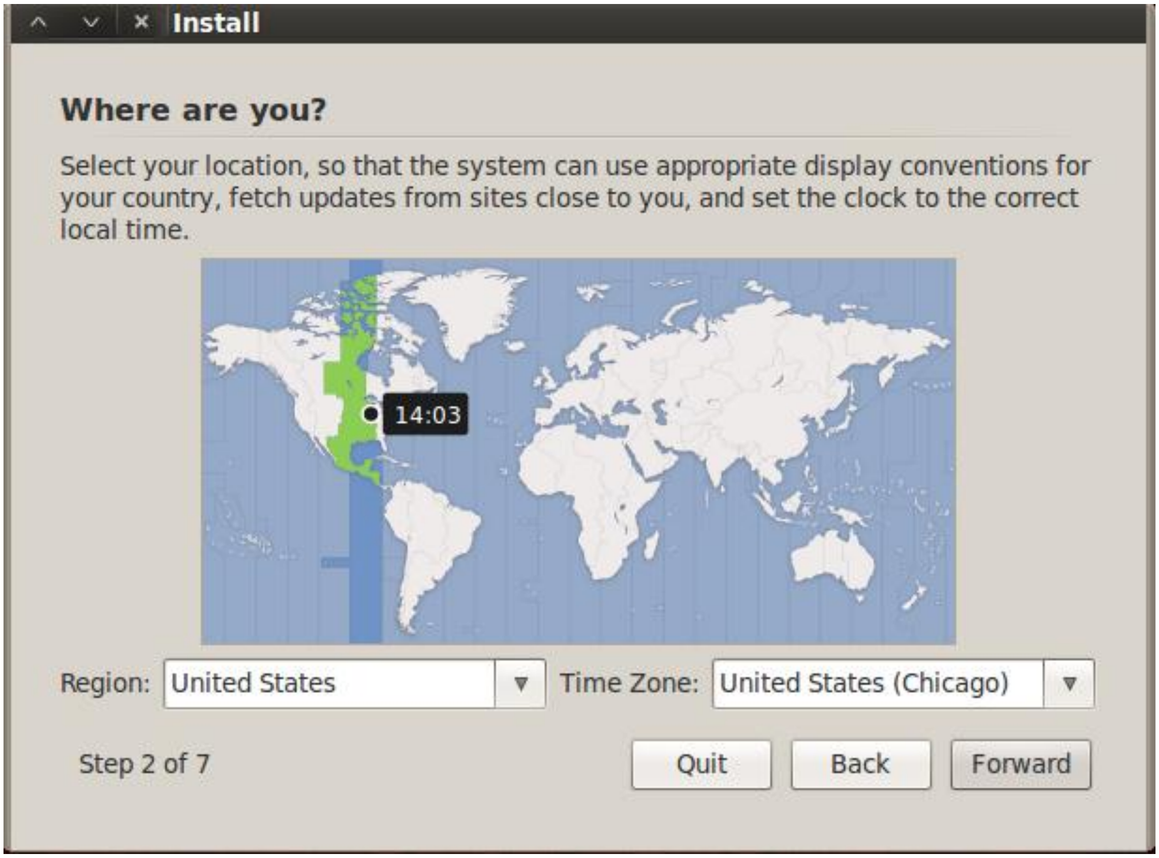
Welcome

Ready to install? Once you answer a few questions, the contents of the live CD can be installed on this computer so you can run BackTrack Live at full speed without the CD.

Answering the questions should only take a few minutes.

Step 1 of 7

Next, select you timezone and click the 'Forward' button.



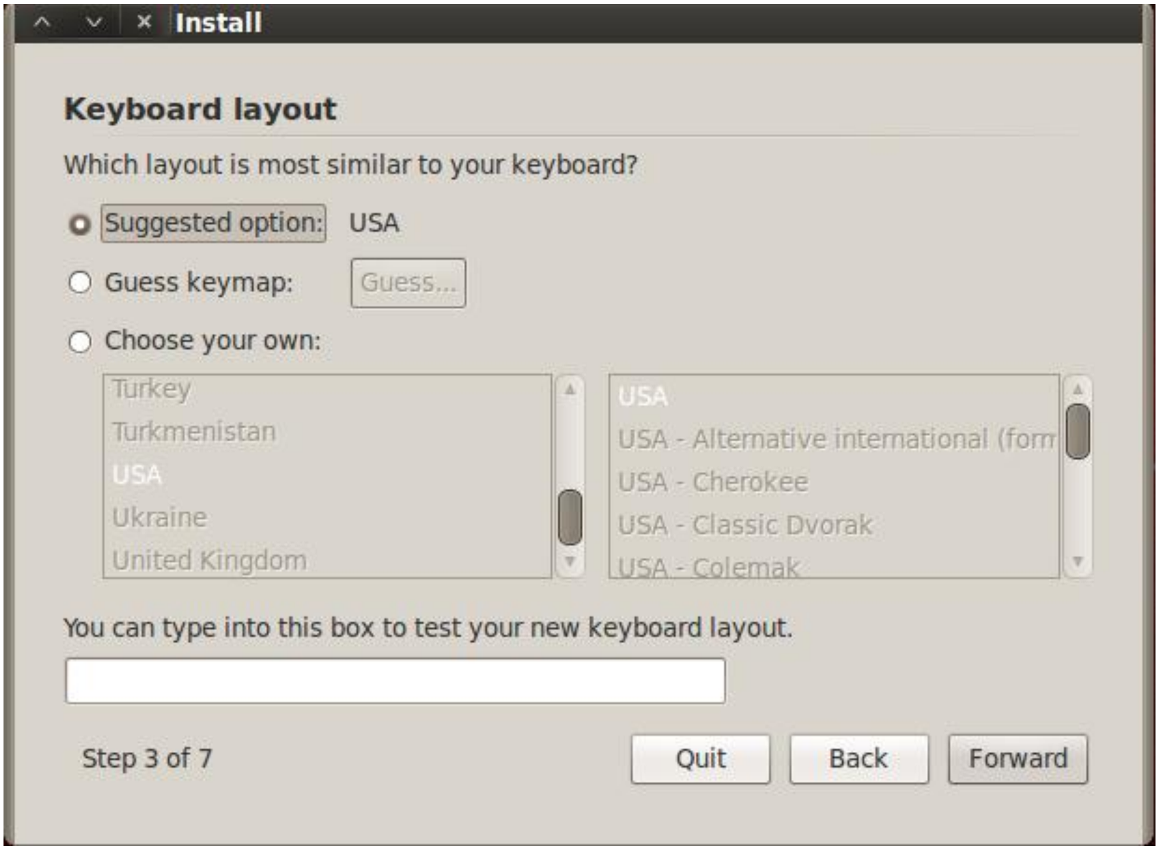
Where are you?

Select your location, so that the system can use appropriate display conventions for your country, fetch updates from sites close to you, and set the clock to the correct local time.

Region: United States ▾ Time Zone: United States (Chicago) ▾

Step 2 of 7

The next step is to select our keyboard layout. Pick yours and click the 'Forward' button. I can not vouch for any keyboard layout other than English.



Keyboard layout

Which layout is most similar to your keyboard?

● Suggested option: USA

○ Guess keymap:

○ Choose your own:

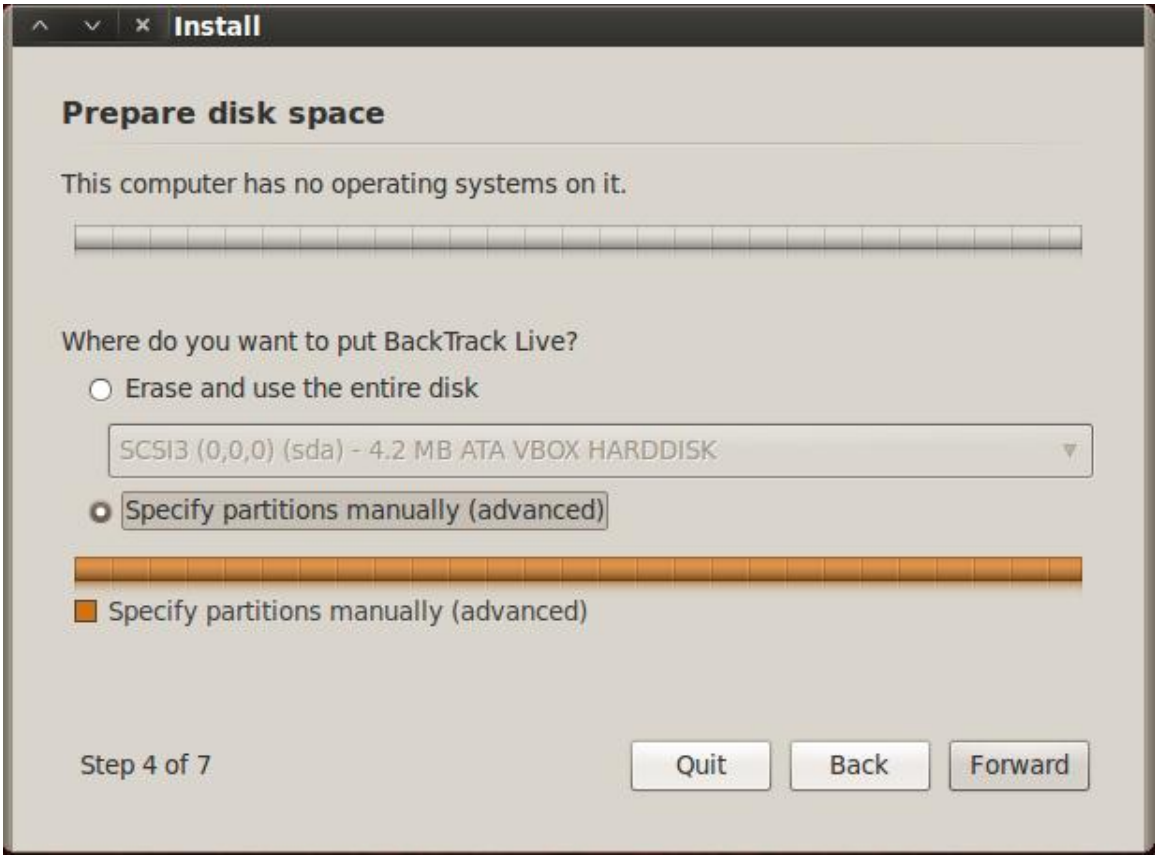
Turkey
Turkmenistan
USA
Ukraine
United Kingdom

USA
USA - Alternative international (forr
USA - Cherokee
USA - Classic Dvorak
USA - Colemak

You can type into this box to test your new keyboard layout.

Step 3 of 7

Click on 'Specify partitions manually' and click the 'Forward' button.



Prepare disk space

This computer has no operating systems on it.

Where do you want to put BackTrack Live?

○ Erase and use the entire disk

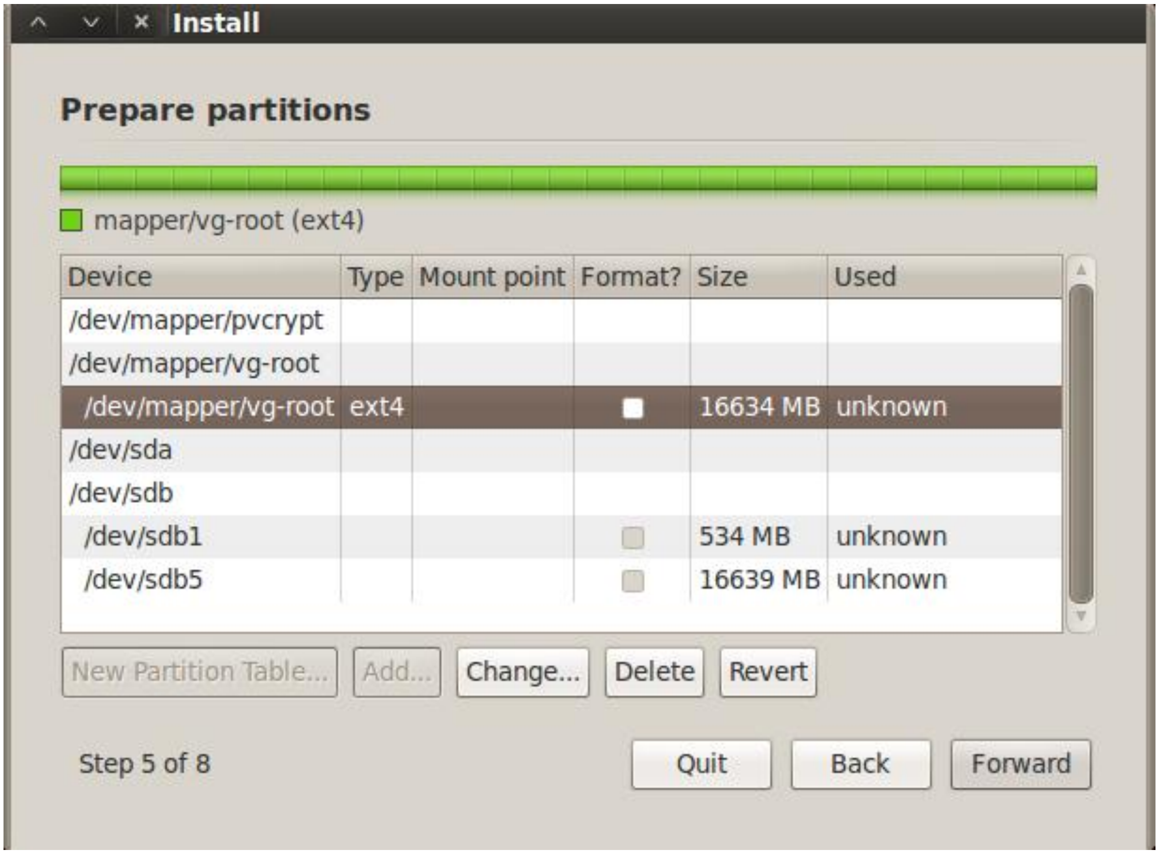
SCSI3 (0,0,0) (sda) - 4.2 MB ATA VBOX HARDDISK

● Specify partitions manually (advanced)

🟧 Specify partitions manually (advanced)

Step 4 of 7

We are not going to indicate the mount points for our partitions. First let's setup our root partition. Click on the row with vg-root in it and click the 'Change' button.



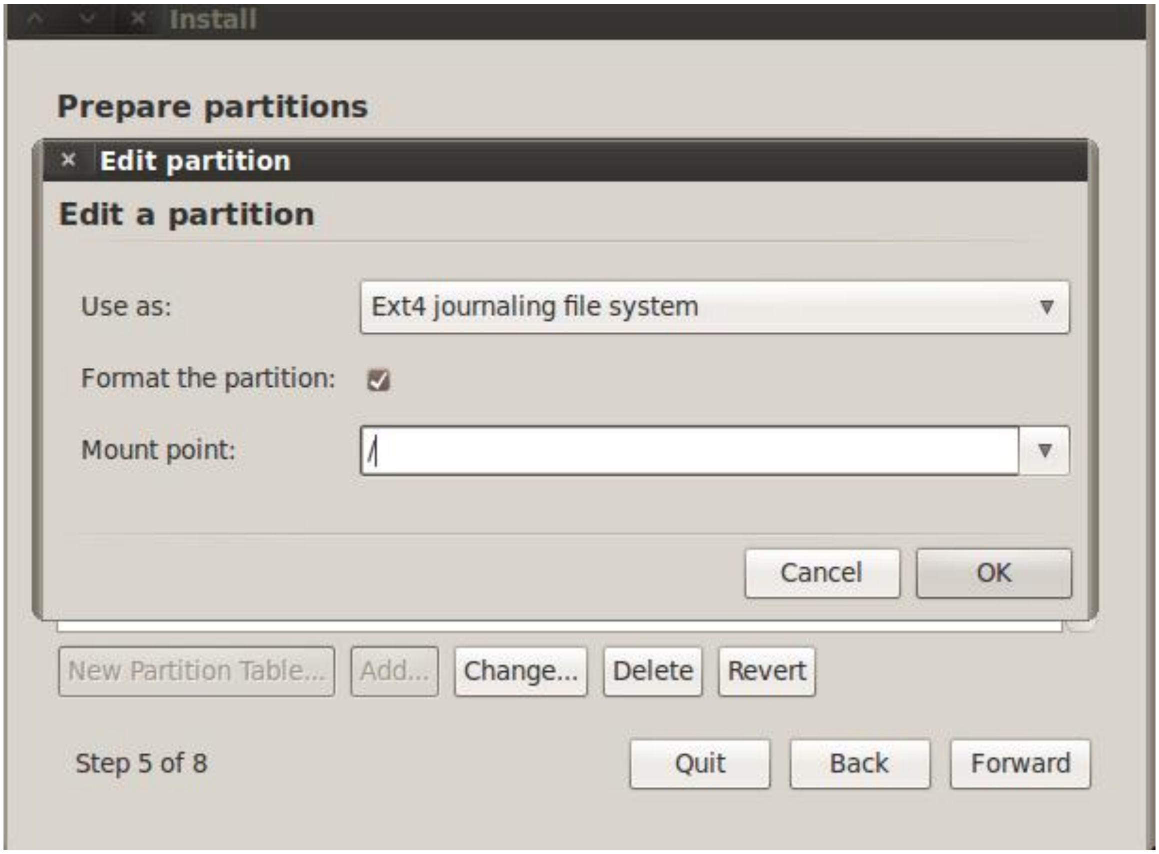
Prepare partitions

🟩 mapper/vg-root (ext4)

Device	Type	Mount point	Format?	Size	Used
/dev/mapper/pvcrypt					
/dev/mapper/vg-root					
/dev/mapper/vg-root	ext4		□	16634 MB	unknown
/dev/sda					
/dev/sdb					
/dev/sdb1			□	534 MB	unknown
/dev/sdb5			□	16639 MB	unknown

Step 5 of 8

Select ext4 from the dropdown menu for 'Use as:', click 'Format the partition:', enter '/' without the quotes for the mount point and click the 'OK' button. The system will re-read the partition table and redisplay it.

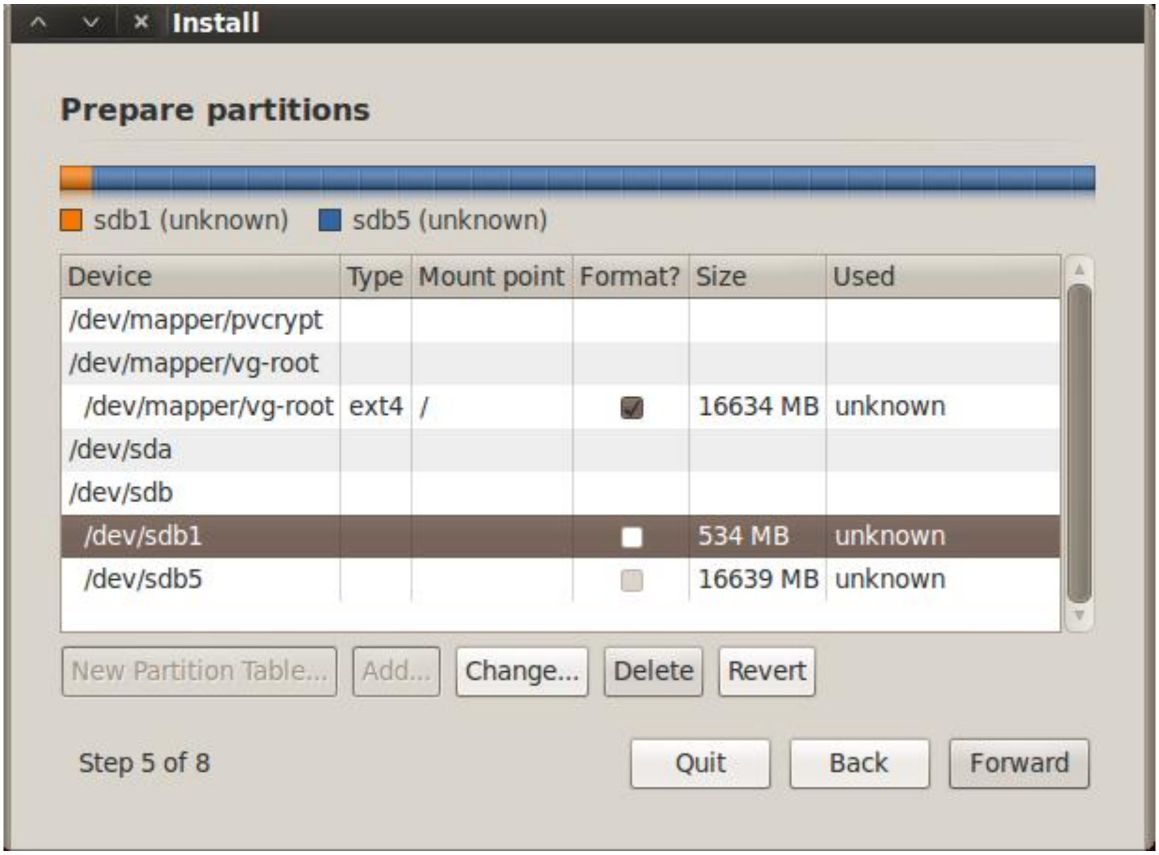


Use as: Ext4 journaling file system

Format the partition: ☑

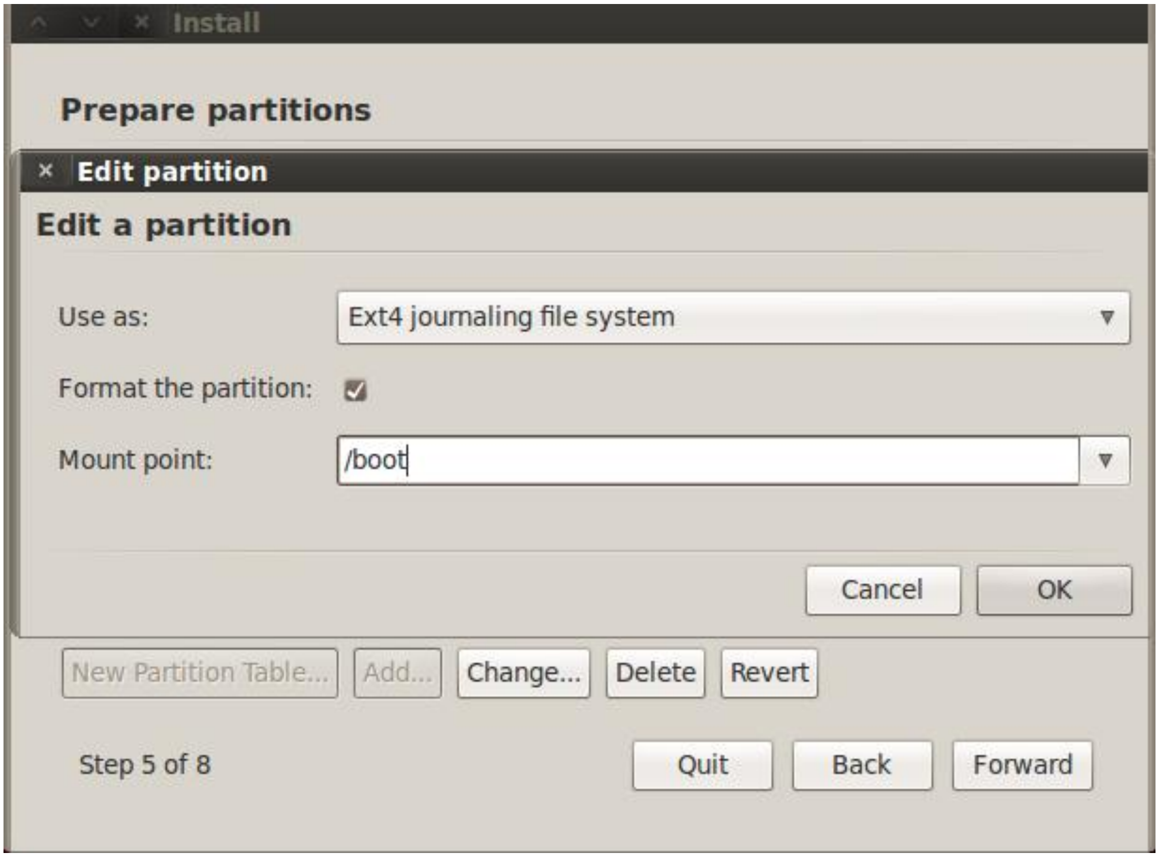
Mount point: /

Now for the boot partition. Click the row with you boot parition in it, /dev/sdb1 in my case, and click the 'Change' button.



Device	Type	Mount point	Format?	Size	Used
/dev/mapper/pvcrypt					
/dev/mapper/vg-root					
/dev/mapper/vg-root	ext4	/	☑	16634 MB	unknown
/dev/sda					
/dev/sdb					
/dev/sdb1			☐	534 MB	unknown
/dev/sdb5			☐	16639 MB	unknown

Again, select ext4 and click the format checkbox. Enter '/boot' without the quotes for the mount point and click the 'OK' button. The disk partition will be re-read and the display updated.



Prepare partitions

Edit a partition

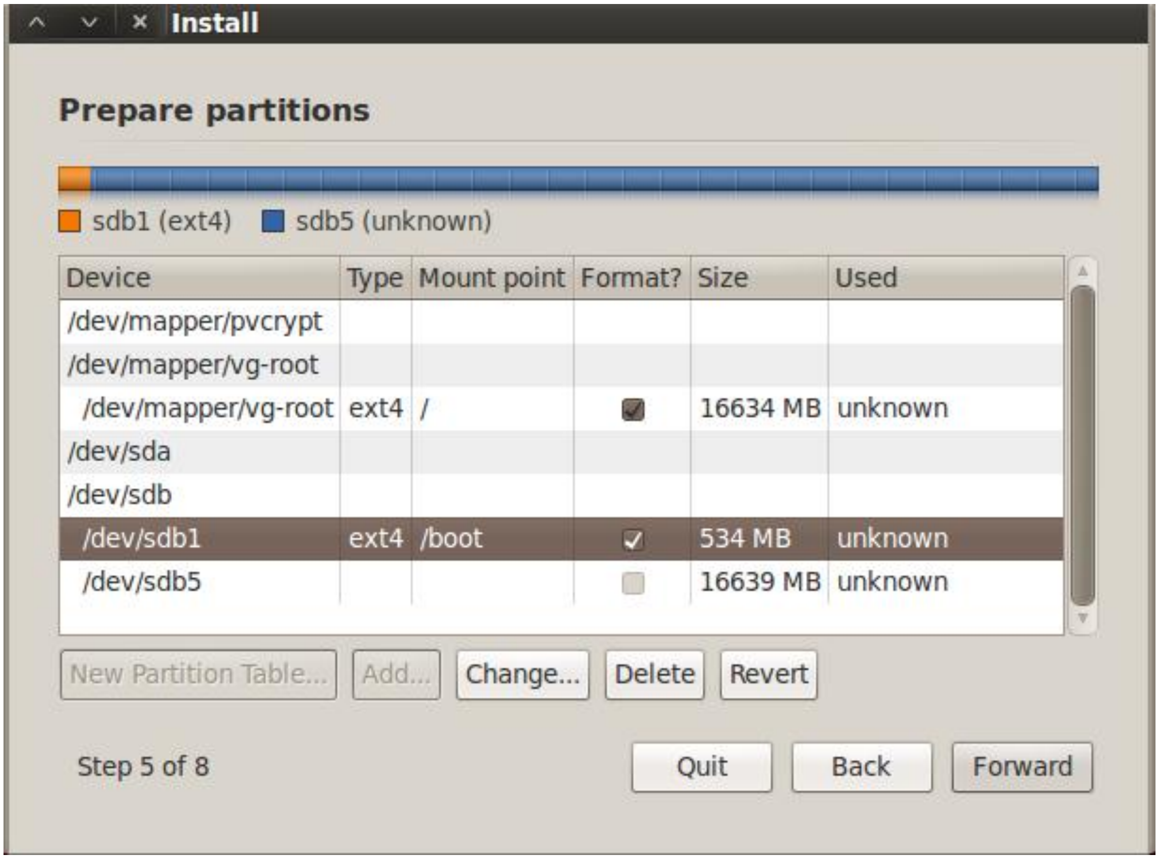
Use as: Ext4 journaling file system

Format the partition: ☑

Mount point: /boot

Step 5 of 8

Click the 'Forward' button.



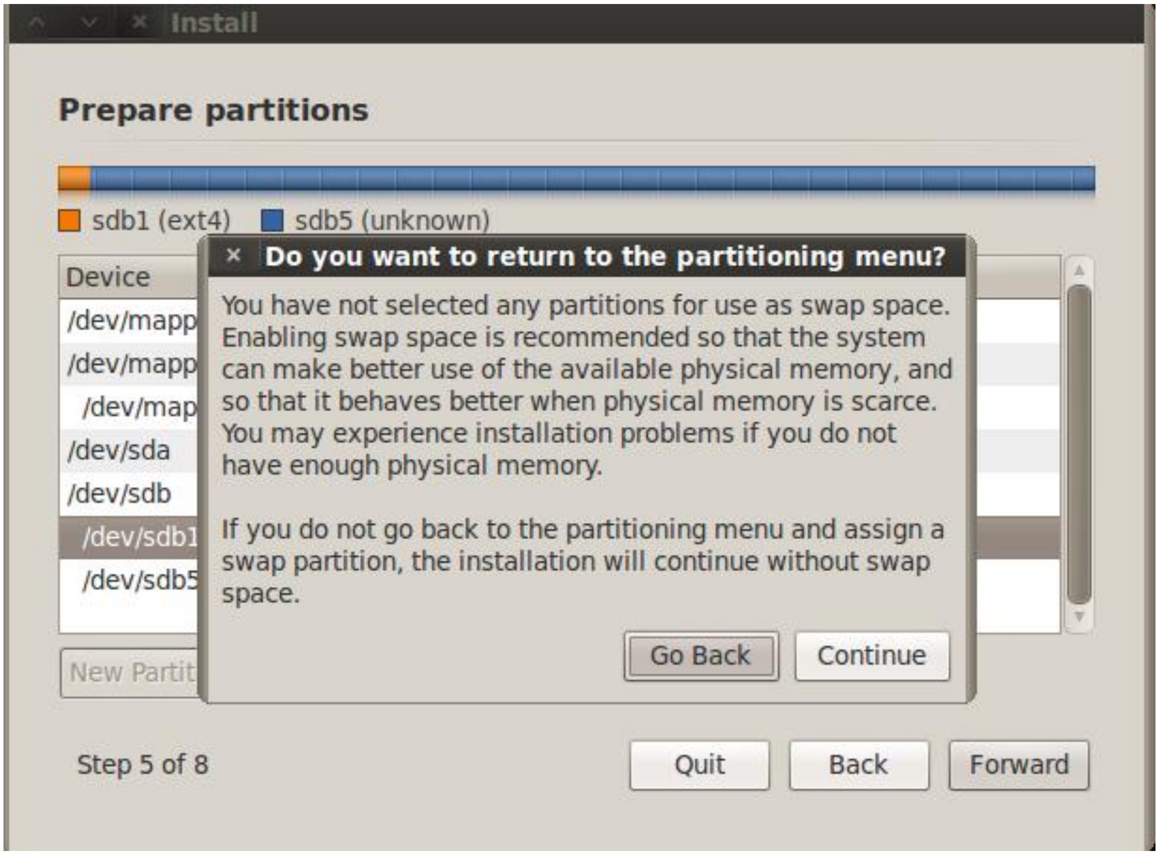
Prepare partitions

🟧 sdb1 (ext4) 🟦 sdb5 (unknown)

Device	Type	Mount point	Format?	Size	Used
/dev/mapper/pvcrypt					
/dev/mapper/vg-root					
/dev/mapper/vg-root	ext4	/	☑	16634 MB	unknown
/dev/sda					
/dev/sdb					
/dev/sdb1	ext4	/boot	☑	534 MB	unknown
/dev/sdb5			☐	16639 MB	unknown

Step 5 of 8

You will get this message if you are installing to a USB drive and not using a swap partition. Click the 'Continue' button.

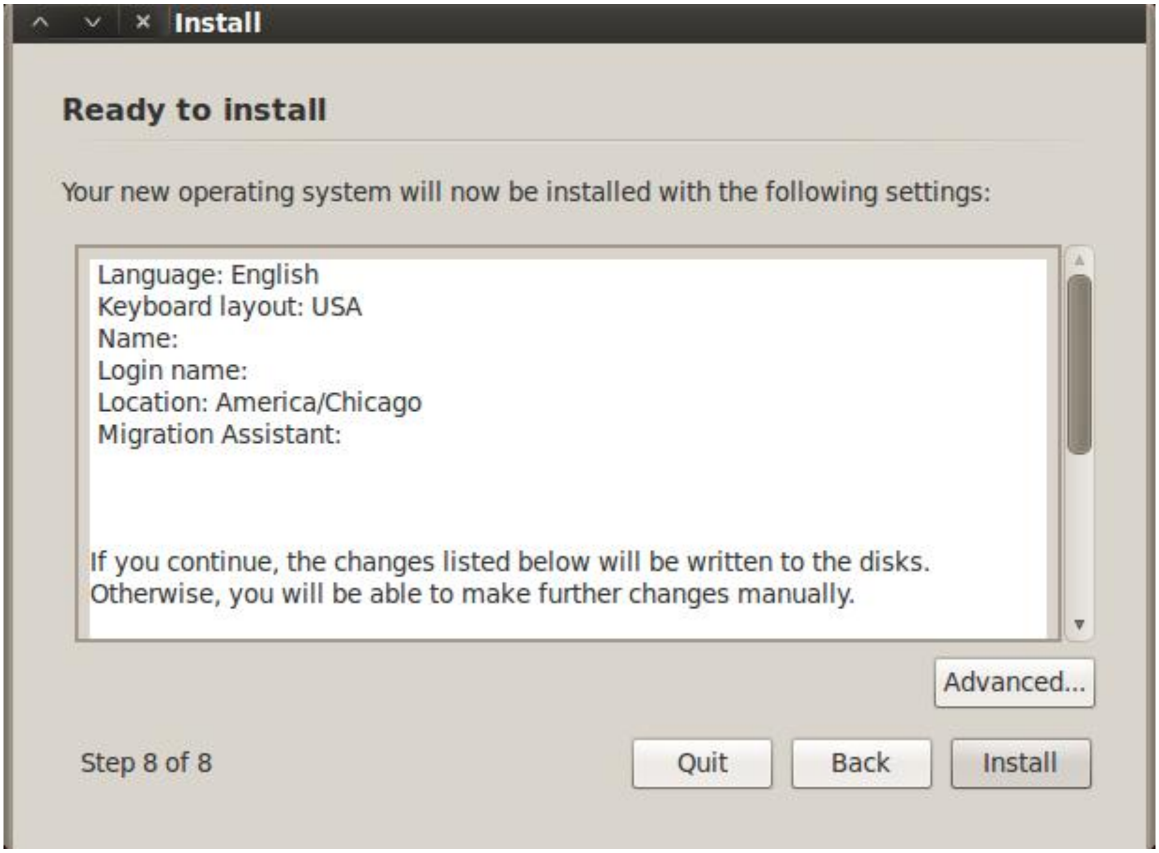


Do you want to return to the partitioning menu?

You have not selected any partitions for use as swap space. Enabling swap space is recommended so that the system can make better use of the available physical memory, and so that it behaves better when physical memory is scarce. You may experience installation problems if you do not have enough physical memory.

If you do not go back to the partitioning menu and assign a swap partition, the installation will continue without swap space.

WARNING: You must click on the advanced tab on the next page and select your USB drive as the target for installing the bootloader. You will break your system if you do not.



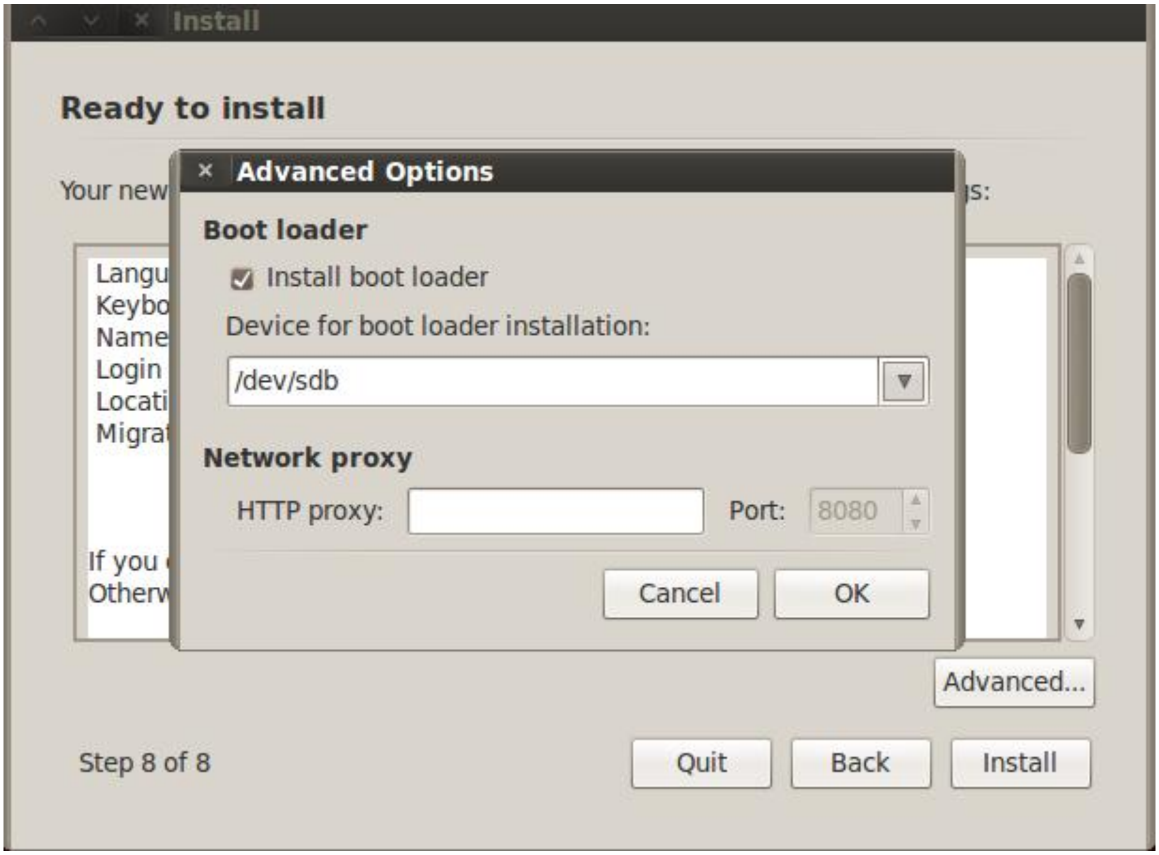
Ready to install

Your new operating system will now be installed with the following settings:

Language: English
Keyboard layout: USA
Name:
Login name:
Location: America/Chicago
Migration Assistant:

If you continue, the changes listed below will be written to the disks. Otherwise, you will be able to make further changes manually.

Don't forget! Make sure you select the target disk for your install as the device for the boot loader to be installed on or you run the risk of making the system you are doing this on non-bootable. Then click on the 'OK' button.



Ready to install

Your new

Langu
Keybo
Name
Login
Locati
Migrat

If you
Otherw

Boot loader

☑ Install boot loader

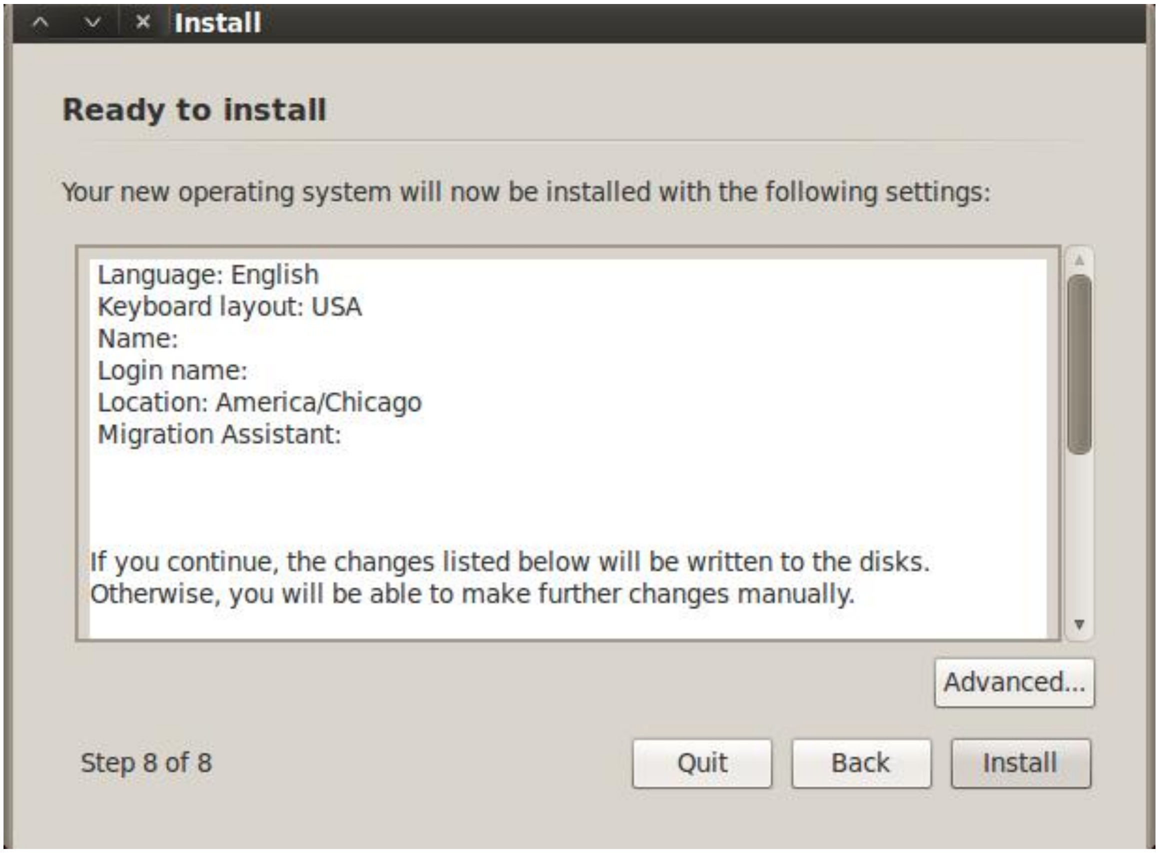
Device for boot loader installation: /dev/sdb

Network proxy

HTTP proxy: Port: 8080

Step 8 of 8

Click the 'Install' button to start the install.



Ready to install

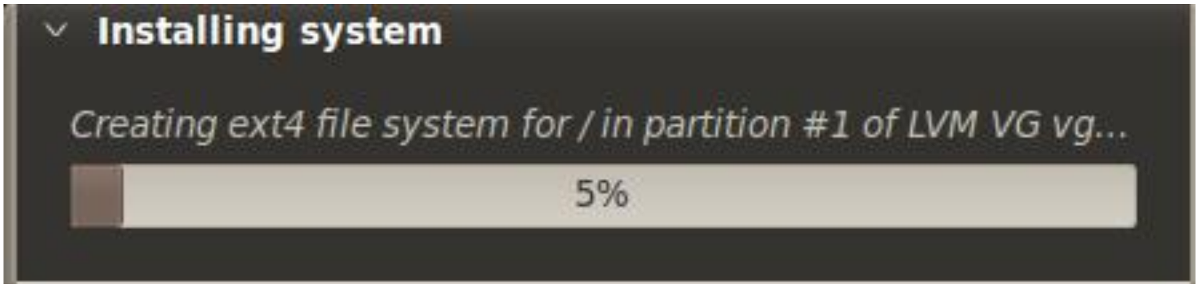
Your new operating system will now be installed with the following settings:

Language: English
Keyboard layout: USA
Name:
Login name:
Location: America/Chicago
Migration Assistant:

If you continue, the changes listed below will be written to the disks.
Otherwise, you will be able to make further changes manually.

Step 8 of 8

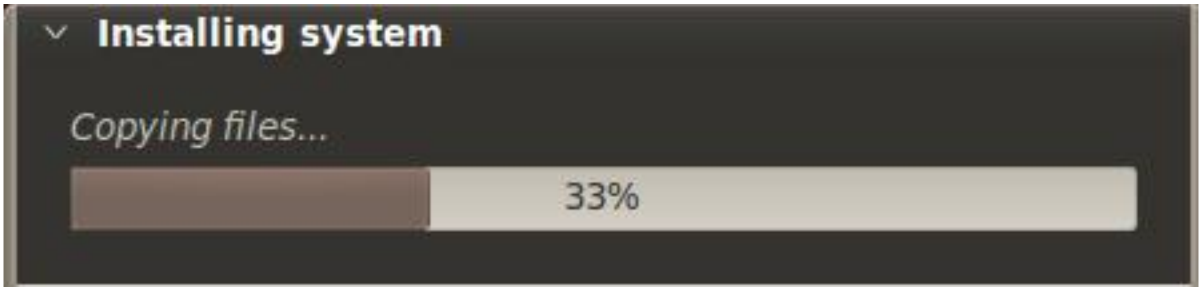
This will take some time. Go get a coke or beverage or your choice and relax for a bit.



Installing system

Creating ext4 file system for / in partition #1 of LVM VG vg...

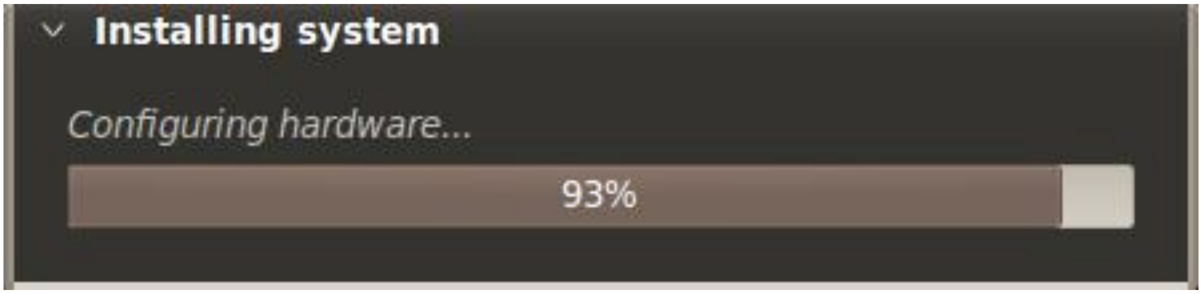
More waiting.



Installing system

Copying files...

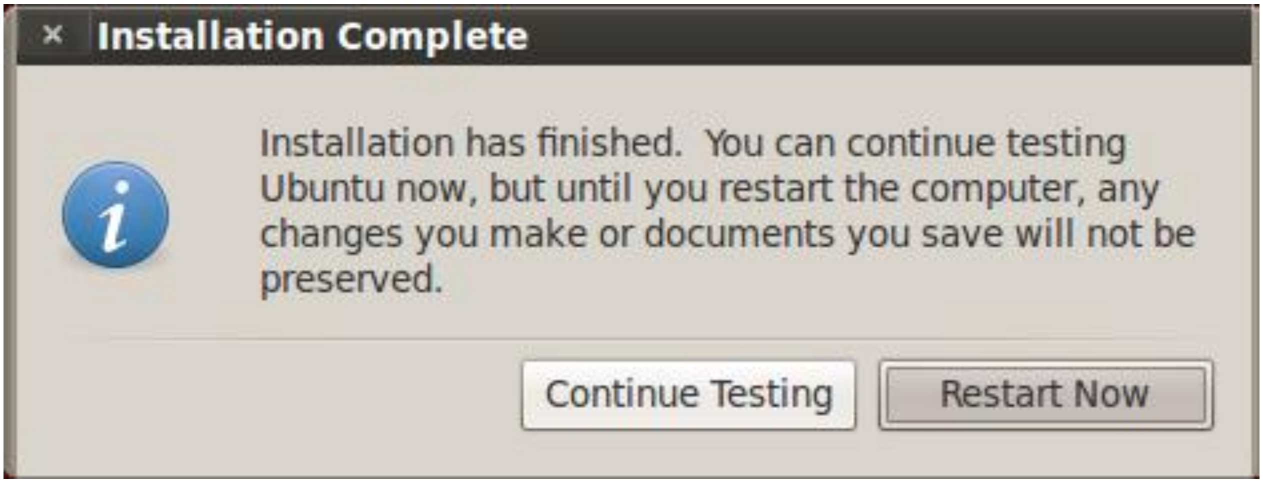
and...more waiting. If it seems like the system is stuck at 99% forever, that's normal, at least in every case where I have done the install.



Installing system

Configuring hardware...

Finally! **Important!** Click on the 'Continue Testing' button. **DO NOT** click on the 'Restart Now' button or you have to redo a bunch of stuff.



Installation Complete

Installation has finished. You can continue testing Ubuntu now, but until you restart the computer, any changes you make or documents you save will not be preserved.

We have now installed the main distribution to our thumb drive. The next step is to configure the newly installed system to use LVM and open the encrypted partition. However, before we do that we need to figure out the UUID of our encrypted volume. We want to do this so that we don't run into problems if the device name of the drive changes from machine to machine. The command we used to use to do this was vol_id. This has changed with Backtrack 5. We now use **blkid**. So execute **blkid**as below.

Make a note of the ID_FS_UUID value which is in italics above. We will need it later. Note: your output will be different than mine. Now time to configure our newly installed system. The first thing we have to do is make the newly installed system active so we can make changes to it. We do that by mounting the partitions and chrooting to it.

```
mkdir /mnt/backtrack5
mount /dev/mapper/vg-root /mnt/backtrack5
mount /dev/sdb1 /mnt/backtrack5/boot
chroot /mnt/backtrack5
mount -t proc proc /proc
mount -t sysfs sys /sys
```

To make everything truly operational, we can mount /dev/pts, but every time I try I have problems unless I reboot first. That is a real pain, so I just don't mount /dev/pts. We will get a couple warnings/errors as we go along, but they do not affect our install. The magic to making all this work is to rebuild the initrd image that is used to boot our system. We need to include some things, load some modules, and tell it to open the encrypted volume, but first we have to go through the whole process of installing software again. We have to do this because we are essentially right back where we started when we booted the live cd. Do the following again.

```
apt-get update
apt-get install hashalot lvm2
```

The next step is to configure how initramfs-tools will create our initrd file. This involves editing one files, the /etc/crypttab file. We used to have to edit /etc/fstab, but it appears we don't need to do that any longer. Mine was correct with /dev/mapper/vg-root as the root entry. If my change it isn't correct in your installation, follow the directions below to correct it. I use the vi editor, but you can use your favorite editor. **vi /etc/crypttab**We need to add the following line to the file. If you are new to vi, hit the o key and the type the following:

```
pvcrypt      /dev/disk/by-uuid/<uuid from above>         none         luks
```

When you are done typing that line, hit the esc key and then type ':wq' without the quotes to save and exit vi. The file should look like this. The uuid is unique to my case. Make sure yours matches your system.

```
# <target device>   <source device>   <key file>   <options> pvcrypt      /dev/disk/by-uuid/09330b5a-5659-4efd-8e9d-0abc404c5162    none         luks
```

Fixing the /etc/fstab file if necessary

If we need to edit the /etc/fstab file, do the following. Again, use your favorite editor or vi. **vi /etc/fstab**The file will look something like below. The UUIDs will be different though.

```
# /etc/fstab: static file system information. # # <file system> <mount point>   <type>  <options>       <dump>  <pass> proc            /proc           proc    defaults        0       0 # /dev/mapper/vg-root UUID=c8d9b9a0-2198-4966-bc3a-39259df6a2c2 / ext4 relatime,errors=remount-ro 0 1 # /dev/sdb1 UUID=6af425ad-99b8-44a5-9ee1-0349141f9b1f /boot   ext4    relatime 0       2
```

The only line we need to change is the line for vg-root which is bolded above. For those new to vi, position the cursor on first 'U' of the line using your arrow keys and type 'dd', then move the cursor to the '#' in the line above and type the letter o, then type the line below, hit the esc key and type ':wq' without the quotes to save the file. The line needs to look like below when done:

```
/dev/mapper/vg-root / ext4  defaults 0 1
```

```
update-initramfs -u
```

If all goes well, you are now ready to cross your fingers and reboot.

IMPORTANT

In my case, the system boots and looks like it is hanging at the Backtrack 5 splash screen. Simply press F8 to get to the console, where it is waiting for you to enter your luks passphrase. Type that bad boy in and, if all goes well, your system will boot.

SUPER IMPORTANT

Do not run aptitude safe-upgrade! It will remove some vital tools. Run apt-get upgrade instead which appears to leave things installed that need to be installed. If you should happen to run aptitude safe-upgrade, ignore the warning about removing packages, type 'Y' and let it do its thing, you will need to run the following command before you reboot or your install will be broken.

```
apt-get install cryptsetup ecryptfs-utils keyutils
```

If you have problems, you can use the troubleshooting directions below to get back to the state where you can try to figure out how what went wrong.

System All Booted

Once you have a booting system, you are ready to login. The default userid is **root** and the default password is **toor**. You are now ready to login and being playing. Don't forget to change the root password as soon as you login the first time.

That's it.

You can make some final tweaks if you want like starting GNOME at boot, but for all intents and purposes you have successfully installed Backtrack 5 to a USB drive and don't have to worry about sensitive information being intercepted if it gets lost of stolen.

Backing It Up

Richard, in comment 241, mentioned backing up his completed install periodically just in case something goes wrong with his USB drive.

This is a fantastic idea.

There are several ways you can accomplish this.

First, on a Linux or other UNIX variant, like OpenBSD or Mac OS X, you can use the dd command.

Note: You will use the device identifier of the **DRIVE**, not a partition, unless you want to dd each partition separately. That seems a bit silly though. For instance, /dev/sdb is the whole drive, while /dev/sdb1 is just the first partition.

# Do not boot to the USB drive for this. Execute the following to create a binary copy of your drive.

```
dd if=/dev/[your device] of=/[destination]/backtrack5USB.img
```

You will need to have free space available on the target drive equivalent to the size of the USB drive. You can compres the image after the dd is complete using gzip or bzip2.

On a Windows machine, you will need to use a utility that will create a binary copy of the USB device. There are several products out there that will do this. Once such free product that will do this is USB Image Tool. This is freeware tool which creates an exact duplicate image of a USB drive. It does require .NET. There are many other options.

Troubleshooting

If you run into any problems, you don't have to start over. As long as your encrypted volume is built correctly and you have the correct LUKS passphrase, you can get back to the place you were with the Live CD. Simply boot with the original Live CD/USB drive and enter the following.

```
/etc/init.d/networking start
apt-get update
apt-get instal hashalot lvm2 # lvm2 not needed for R2
cryptsetup luksOpen /dev/[your logical partition] pvcrypt
mkdir /mnt/backtrack5
mount /dev/mapper/vg-root /mnt/backtrack5
mount /dev/[boot partition] /mnt/backtrack5/boot
chroot /mnt/backtrack5
mount -t proc proc /proc
mount -t sysfs sys /sys
mount -t devpts devpts /dev/pts
```

You can now do any trouble shooting you need to do and try to reboot again. One note, if you want to check the UUID of your partition, do it before you chroot.

-Kevin