Trinityhome

Trinity Rescue Kit | CPR for your computer

Getting started with TRK

Foreword and conventions of this documentation:

Consulting help

-Trinity Rescue Kit 3.4 has manpages for almost every utility, even the ones specific to TRK (new since 3.4). So if you need help on a certain command, like f.i. Winpass, just type 'man winpass' at the commandline. All manpages themselves are always online available at http://trinityhome.org/manpages

-This helppage is also available as a single document locally on the TRK medium. **You can call on this documentation as one big page from TRK by typing 'trkhelp'** at the command prompt (or chosen from the startup menu). This will start the builtin Links browser in graphical mode opening all of the documentation at once. To be able to switch between this help and your commandprompt, you must be in text mode

Most commands also have a built-in help. Most of the time 'command -h' or 'command –help' will help you a lot further.

Use the key "q" to quit links and type 'trkhelp -t' to run in text mode. Once in textmode, use alt+F2 to go the second console of TRK. In total there are 6 consoles, each switchable with their respective alt+ function key.
If you are not online, you can consult the local documentation which is on the TRK medium by entering 'trkhelp -l'.
You can combine these two parameters: 'trkhelp -l -t' gets you the local helpfiles in textmode.

-All TRK manpages are also available online in html format. Browse them here: http://trinityhome.org/manpages/

Conventions

-literal commands that you can execute in TRK or Linux are put between 'single quotes'. Omit the quotes when using the real commandline. Exceptions on these quotes will be mentionned (when single quotes really appear in the command).

-"double quotes" are used to emphasise words, unless they are used inside commands.

-<trkmedium> stands for the rootfolder of the medium on which TRK runs. TRK can be run from CD, usb stick/disk, fixed harddisk or from network over PXE. These specific bootmethods will be explained later in this document.

-this documentation is intended for people who at least have some experience with

computer troubleshooting or know how to install their own Windows. If you have absolutely no idea of this, I recommend you call someone who knows more.

0. Quick and dirty guide to using TRK

This page is intended for the really impatient who are passing by here and probably just want to reset a password in Windows. The procedures assume you are using MS Windows.

I'll make sure I don't type too much text for you to read.

In short...

TRK is not a software you install on your computer in Windows but rather a completely independent operating system based on Linux and which runs from CD (or USB stick or network).

To get the latest version of TRK, go to the download page or download the latest copy here.

The quickest way to get you running TRK is to download and run the self-burning TRK version.

If you want to see how the self burning of TRK is done, see the page on 0.1 Self burning TRK

Should you want to burn the iso with a 3rd party software called magiciso, see 0.2 Burning TRK with Magiciso

For booting from TRK, see 0.3 Booting from TRK

For password resetting, see 0.4 Resetting passwords

If you know how to burn an isofile, skip section 0.1 and 0.2.

If you know how to boot from CD, skip section 0.3

For password resets, you can equally skip section 0.4 because TRK 3.4 now has a simple menu from which you can select whatever you need to do.

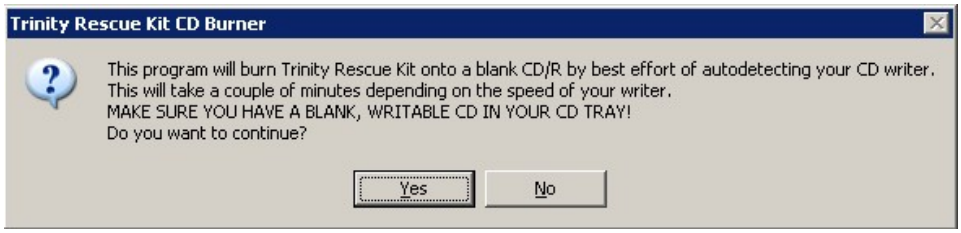
0.1 The easiest way to get it onto a CD: a self burning TRK

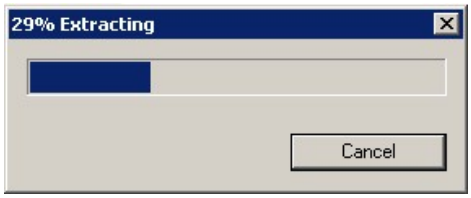
-Download and save the latest copy of the self-burning TRK, e.g. "trinity-rescue-kit.3.4-build-372.exe"

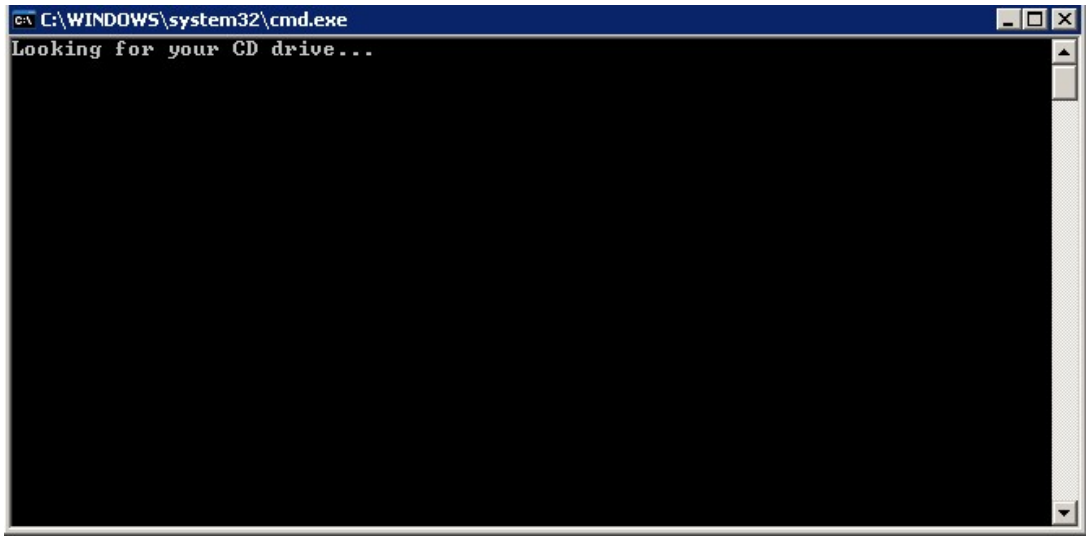
-Rightclick on "trinity-rescue-kit.3.4-build-372.exe" and select "Run as administrator"

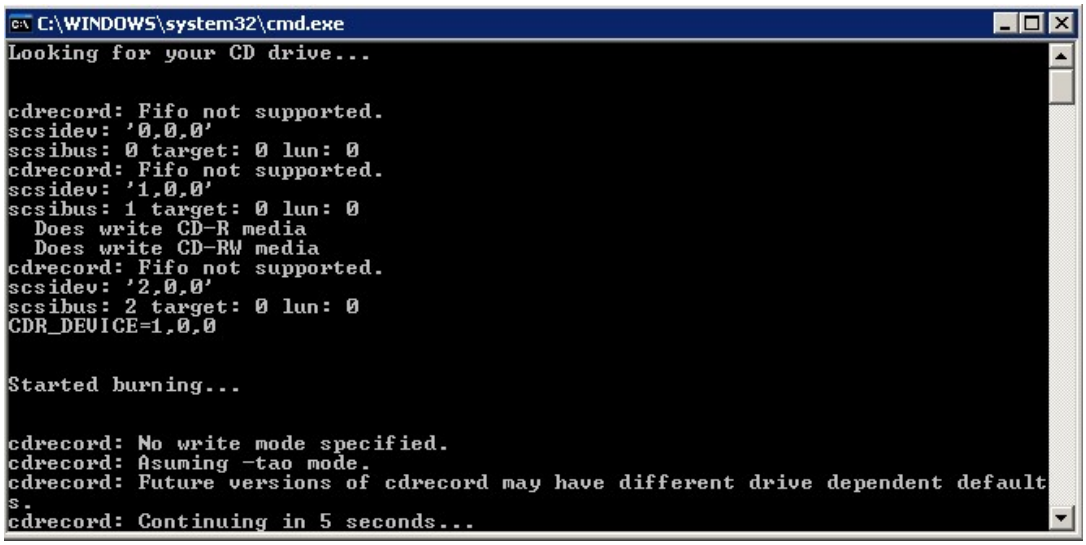
Now see the screenshots

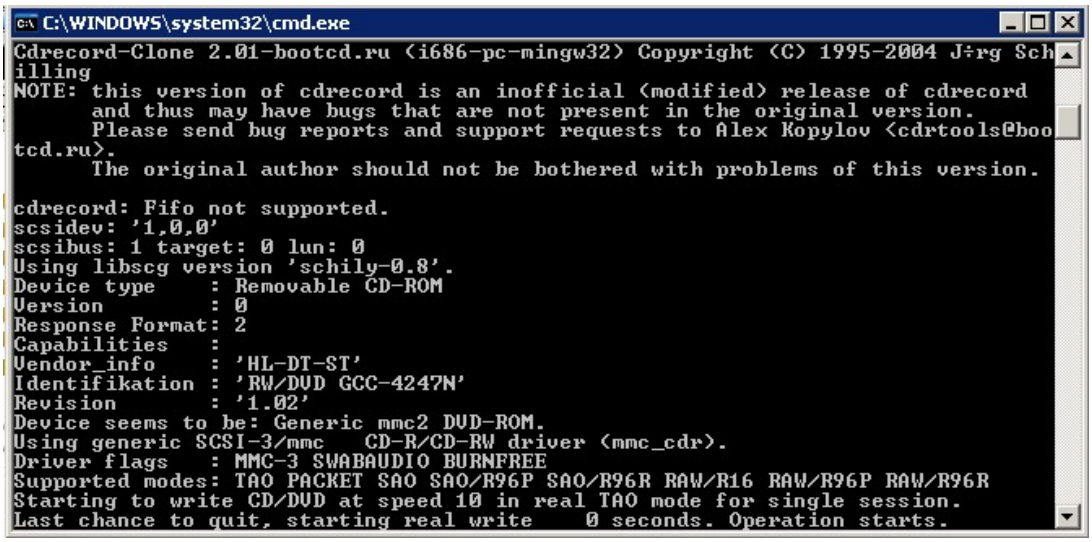
Ok, you've put your blank CD in the tray, now answer yes and see all of the next screens pass by. There 's no more work for you on the burning part!

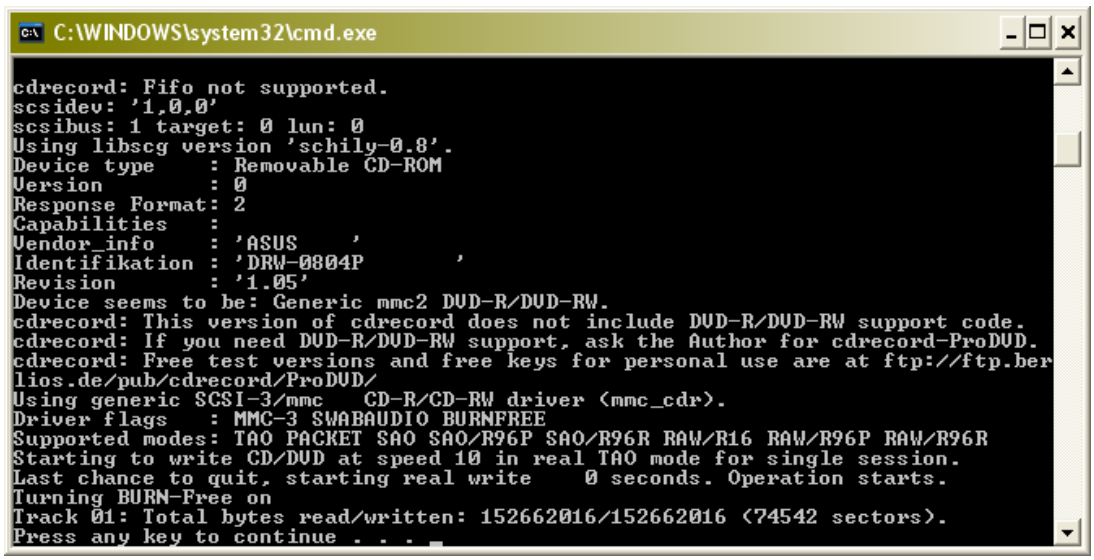







C:\WINDOWS\system32\cmd.exe
Looking for your CD drive...
cdrecord: Fifo not supported.
scsidev: '0,0,0'
scsibus: 0 target: 0 lun: 0
cdrecord: Fifo not supported.
scsidev: '1,0,0'
scsibus: 1 target: 0 lun: 0
Does write CD-R media
Does write CD-RW media
cdrecord: Fifo not supported.
scsidev: '2,0,0'
scsibus: 2 target: 0 lun: 0
CDR_DEVICE=1,0,0
Started burning...
cdrecord: No write mode specified.
cdrecord: Asuming -tao mode.
cdrecord: Future versions of cdrecord may have different drive dependent default
s.
cdrecord: Continuing in 5 seconds...


C:\WINDOWS\system32\cmd.exe
Cdrecord-Clone 2.01-bootcd.ru (i686-pc-mingw32) Copyright (C) 1995-2004 J÷rg Sch
illing
NOTE: this version of cdrecord is an inofficial (modified) release of cdrecord
and thus may have bugs that are not present in the original version.
Please send bug reports and support requests to Alex Kopylov <cdrtools@boo
tcd.ru>.
The original author should not be bothered with problems of this version.
cdrecord: Fifo not supported.
scsidev: '1,0,0'
scsibus: 1 target: 0 lun: 0
Using libscg version 'schily-0.8'.
Device type : Removable CD-ROM
Version : 0
Response Format: 2
Capabilities :
Vendor_info : 'HL-DT-ST'
Identifikation : 'RW/DVD GCC-4247N'
Revision : '1.02'
Device seems to be: Generic mmc2 DVD-ROM.
Using generic SCSI-3/mmc CD-R/CD-RW driver (mmc_cdr).
Driver flags : MMC-3 SWABAUDIO BURNFREE
Supported modes: TAO PACKET SAO SAO/R96P SAO/R96R RAW/R16 RAW/R96P RAW/R96R
Starting to write CD/DVD at speed 10 in real TAO mode for single session.
Last chance to quit, starting real write 0 seconds. Operation starts.


C:\WINDOWS\system32\cmd.exe
cdrecord: Fifo not supported.
scsidev: '1,0,0'
scsibus: 1 target: 0 lun: 0
Using libscg version 'schily-0.8'.
Device type : Removable CD-ROM
Version : 0
Response Format: 2
Capabilities :
Vendor_info : 'ASUS '
Identifikation : 'DRW-0804P '
Revision : '1.05'
Device seems to be: Generic mmc2 DVD-R/DVD-RW.
cdrecord: This version of cdrecord does not include DVD-R/DVD-RW support code.
cdrecord: If you need DVD-R/DVD-RW support, ask the Author for cdrecord-ProDVD.
cdrecord: Free test versions and free keys for personal use are at ftp://ftp.ber
lios.de/pub/cdrecord/ProDVD/
Using generic SCSI-3/mmc CD-R/CD-RW driver (mmc_cdr).
Driver flags : MMC-3 SWABAUDIO BURNFREE
Supported modes: TAO PACKET SAO SAO/R96P SAO/R96R RAW/R16 RAW/R96P RAW/R96R
Starting to write CD/DVD at speed 10 in real TAO mode for single session.
Last chance to quit, starting real write 0 seconds. Operation starts.
Turning BURN-Free on
Track 01: Total bytes read/written: 152662016/152662016 (74542 sectors).
Press any key to continue . . .

...and that 's it, now boot from it.

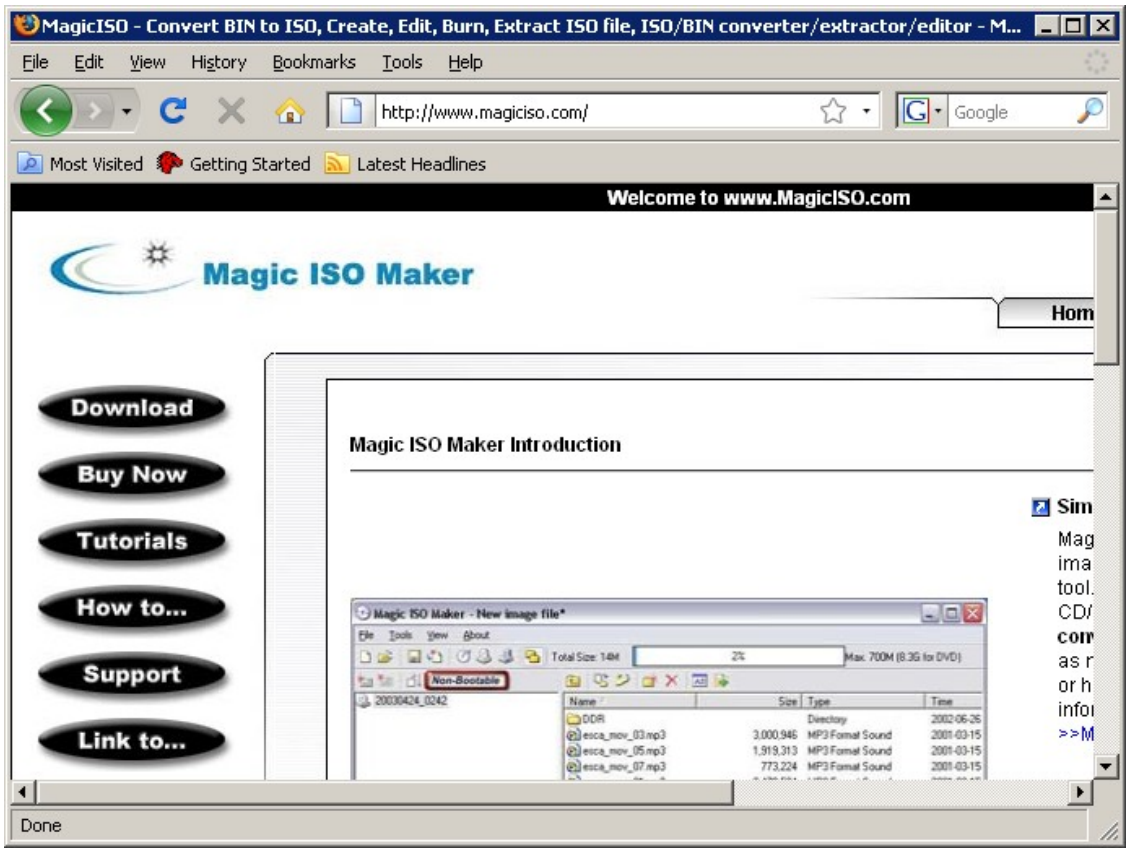
0.2 Burning TRK with Magiciso

There are many other programs that can burn ISO files to a CD. Magiciso is just used as an example because it's one of the most easy to use. An alternative might be the very lightweight BurnCDCC or another free and full blown CD writer is CD Burner XP.
Users from Windows 7 can just burn an ISO to disk without installing any additional software.

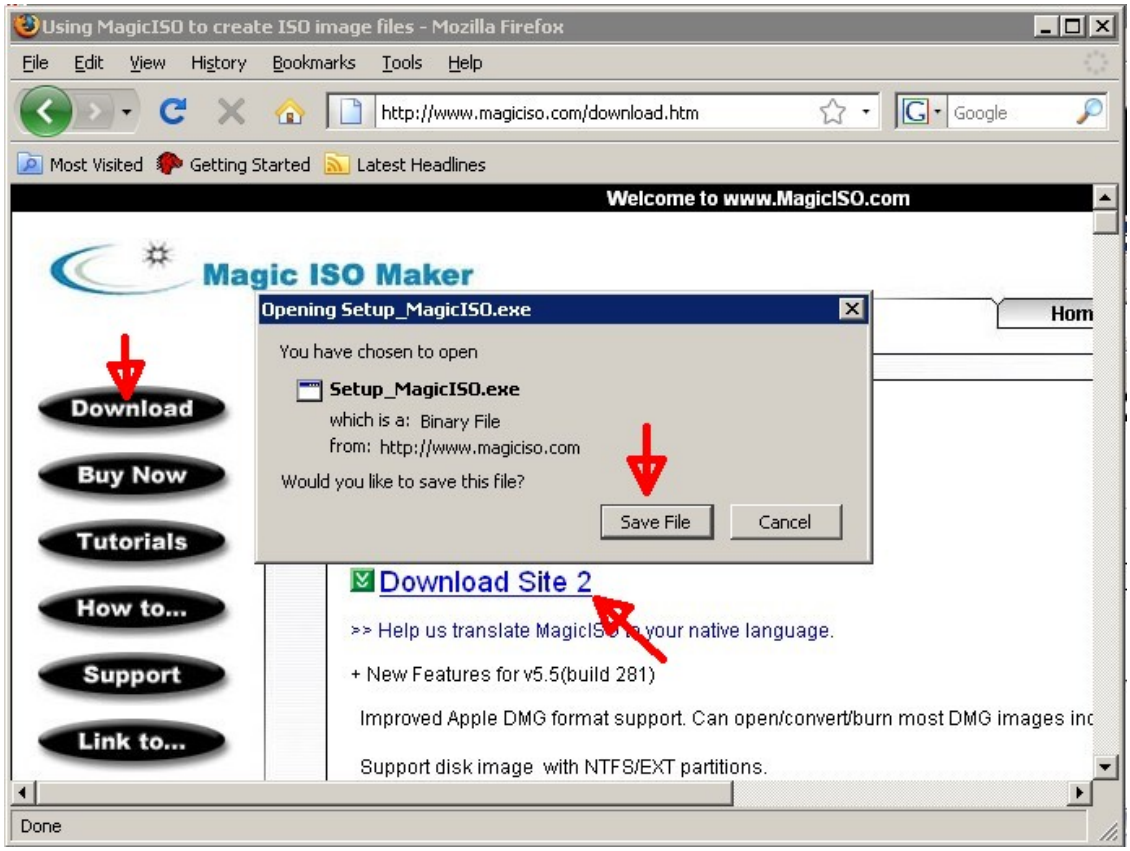
Remember that to install programs or run the self burning TRK, you have to be an administrator of your local computer.

The screenshots speak for themselves

1



2


Using MagicISO to create ISO image files - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://www.magiciso.com/download.htm
Google
Most Visited Getting Started Latest Headlines
Welcome to www.MagicISO.com
Magic ISO Maker
Opening Setup_MagicISO.exe
You have chosen to open
Setup_MagicISO.exe
which is a: Binary File
from: http://www.magiciso.com
Would you like to save this file?
Save File
Cancel
Download
Buy Now
Tutorials
How to...
Support
Link to...
Download Site 2
>> Help us translate MagicISO to your native language.
+ New Features for v5.5(build 281)
Improved Apple DMG format support. Can open/convert/burn most DMG images inc
Support disk image with NTFS/EXT partitions.
Done

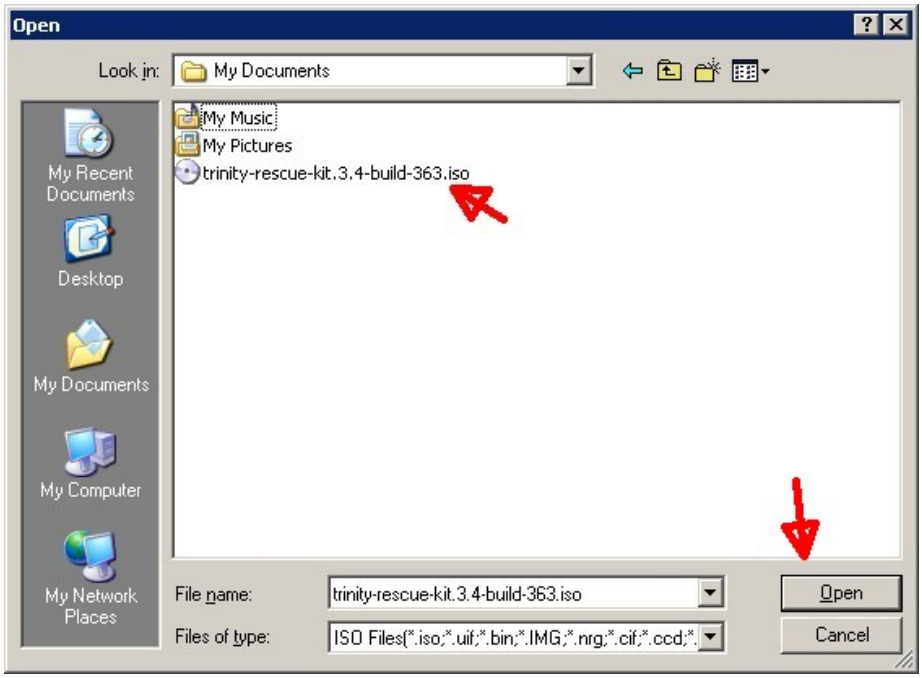
3


Magic ISO Maker(Unregister)
Magic ISO Maker
Thanks for trying Magic ISO Maker
Low Price --Order today for only $29.95
Free updates--Free all kind of updates.
Secure Ordering--We use only secure internet information.
Free Support--Fast support for registered users.
No limits --All limits are removed in the registered version.
Five Great Reasons to Order Now!
You have been using an unregistered version of MagicISO.Unregistered version can not save an image greater than 300MB.You must purchase the program. You can purchase registered version by online. Simply click the "Buy Now!" button to purchase online with email delivery!
http://www.MagicISO.com
Support@MagicISO.com
Buy Now!
Register
Try It!

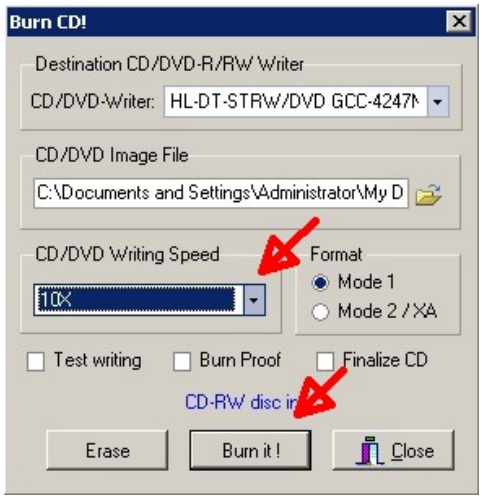
4


Burn CD/DVD!
Destination CD/DVD-R/RW Writer
CD/DVD-Writer:
HL-DT-STRW/DVD GCC-4247N
CD/DVD Image File
CD/DVD Writing Speed
Format
Mode 1
Mode 2 / XA
Test writing
Burn Proof
Finalize CD
CD-RW disc in, but not blank
Erase
Burn it !
Close

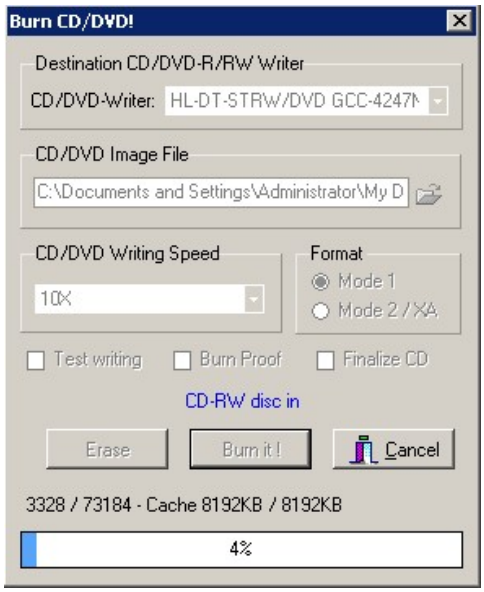
5


Open
Look in:
My Documents
My Music
My Pictures
trinity-rescue-kit.3.4-build-363.iso
My Recent Documents
Desktop
My Documents
My Computer
My Network Places
File name:
trinity-rescue-kit.3.4-build-363.iso
Files of type:
ISO Files(*.iso;*.uif;*.bin;*.IMG;*.nrg;*.cif;*.ccd;*.
Open
Cancel

6


Burn CD!
Destination CD/DVD-R/RW Writer
CD/DVD-Writer: HL-DT-STRW/DVD GCC-4247N
CD/DVD Image File
C:\Documents and Settings\Administrator\My D
CD/DVD Writing Speed
10X
Format
Mode 1
Mode 2 / XA
Test writing
Burn Proof
Finalize CD
CD-RW disc in
Erase
Burn it !
Close

7


Burn CD/DVD!
Destination CD/DVD-R/RW Writer
CD/DVD-Writer: HL-DT-STRW/DVD GCC-4247N
CD/DVD Image File
C:\Documents and Settings\Administrator\My D
CD/DVD Writing Speed
10X
Format
Mode 1
Mode 2 / XA
Test writing
Burn Proof
Finalize CD
CD-RW disc in
Erase
Burn it !
Cancel
3328 / 73184 - Cache 8192KB / 8192KB
4%

8


Burn CD/DVD! (Success)
Destination CD/DVD-R/RW Writer
CD/DVD-Writer: HL-DT-STRW/DVD GCC-4247N
CD/DVD Image File
C:\Documents and Settings\Administrator\My D
CD/DVD Writing Speed
16X
Format
Mode 1
Mode 2 / XA
Test writing
Burn Proof
Finalize CD
No CD-R/RW disc in
Erase
Burn it !
Close

0.3 Booting from TRK

To be able to boot from TRK, I can give you a few tips, but there is never 1 uniform way on how to do it because every computer is different and every brand has different shortcut keys.

In many cases, the fact that the CD with TRK is in your CD tray when you boot your computer might be enough to get it booting from it.

In other cases, you get the option to select the bootdevice at startup. Depending on the brand, this might be with any different function key or even "esc" and "enter". Most of the time, the power on screen tells you what key to press.

Ultimately you must set the bootsequence in the bios of the computer. The bios is basic configuration of a computer before even any operating system or software has been started.

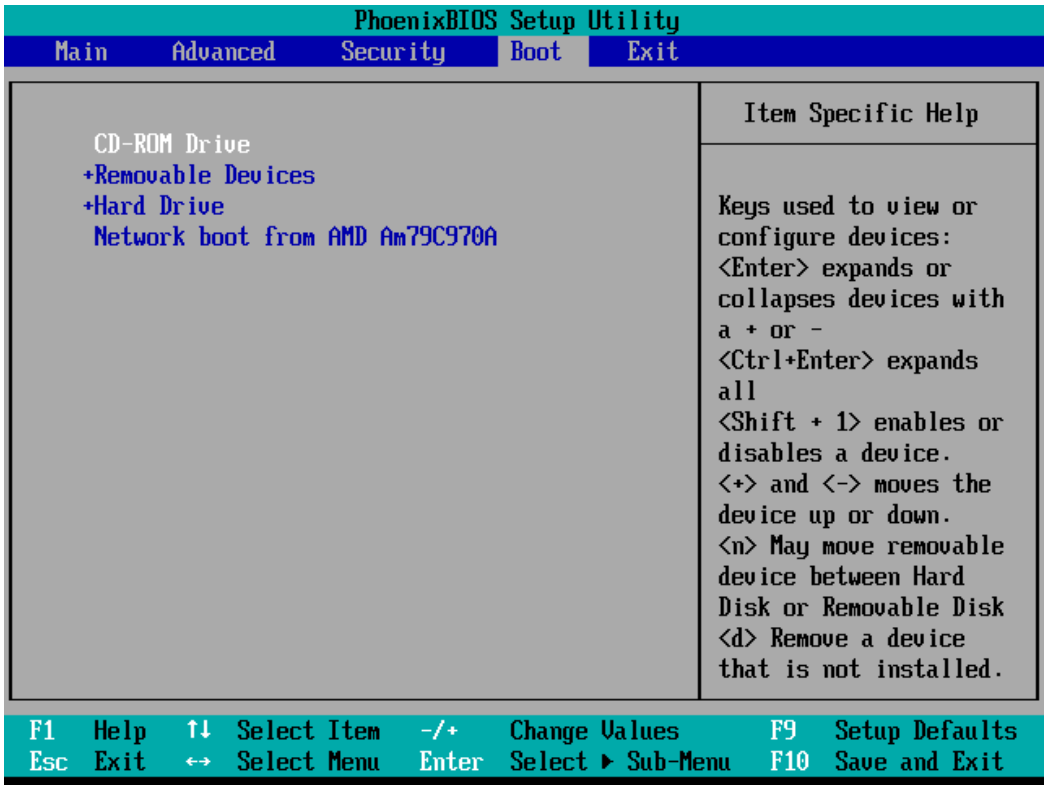
Here 's a few screenshots I've taken from VMWare, which basically behaves like any other physical computer.

Here 's the initial bootsplash which lasts only a few seconds.

The text speaks for itself.

Hitting "esc" is enough to select a temporary bootdevice.

Alternatively if you hit F2, you can go into the bios and set the bootsequence permanently. Remember that in that case when you have a bootable CD in your tray, your PC will always boot from that. It will also make startup slower because it will first look for a cd and its bootability.



Now this is only an example for VMWare. Any other computer is different in the fact that it can be any of the other function keys you need to press to enter the bios or choose a bootdevice. Read the text from the splash screen.

To get a better explanation on setting the bootsequence, take a look at the documentation done for Hiren's Bootcd:

http://www.hiren.info/pages/bios-boot-cdrom

0.4 Resetting passwords

Once you 've managed to burn TRK to CD and set the right bootsequence, you can start doing stuff with it, like resetting the password (that's what you came here for didn't you?).

See this little movie which runs you through it from A to Z and do some further reading on the usage of winpass.

You will notice that once you started TRK, the simple menu that you get is self explanatory.

1. TRK for Linux newbies

This section gives a quick introduction about the concepts of Linux, a live distribution and how you should see and control your computer from the viewpoint of TRK

1.1 What is TRK? What 's a live distribution?

This is a definition for people that have absolutely no idea of what an operating system means. If you don 't understand what I just said, you need to read the following text. The people who do know what it all means, might take the following definition as incorrect, but that 's just so I can explain in simple language what I mean.

“Trinity Rescue Kit" or **TRK is a collection of programs that can start a computer in an alternate way if it failed to start normally.**

TRK is a so called "live distribution" of Linux. Linux is in fact the "brain" and "the senses" that drives your computer, all the programs are the limbs that allow you to control it. All this put together make up a so called operating system. Because of confusion in terminology and because of the so many flavors, we talk about Linux being the operating system and what makes it complete with programs is called the distribution. And as for distributions, a lot of flavors exist (hundreds, maybe even thousands). One of these flavors is Trinity Rescue Kit. TRK is a "live" distribution because it can boot from removable media, perform hardware detection on-the-fly and automate as many configuration tasks as possible.

TRK operates completely from RAM (= compare this with the short term memory of your brain) and read-only media. This means also it doesn ‘t touch nor change

anything stored on your computer until you tell it to do so. Another implication this will have is that by starting your computer with Trinity Rescue Kit you have the guarantee that no viruses that might be on your local computer can become active. In the case of your computer running Microsoft Windows in normal circumstances, it just is impossibe by design that a windows virus can run on TRK because TRK is Linux.

You have the possibility of using 5 different antivirus scanners with TRK (current version of this writing: 3.4). One of them, the free open source Clamav is integrated in the distribution, the other 4, F-prot, Bitdefender Vexira and Avast get downloaded from the Internet when launched. For Avast you need a free license key handy, for which you need to register on their site.

More on what TRK is can be read on the frontpage

1.2 What is different between accessing your PC from Windows and accessing from TRK?

Instead of running programs on your computer using Windows, you are now starting your computer with something completely different. Trinity Rescue Kit is not designed to give you the same environment you get normally, but to provide you the means and the utilities to perform rescue and repair operations that might not be (anymore) possible on your computer in normal Windows operation mode.

Because this is Linux, you will not see your drives in the same way you do under Windows (or DOS), but they will appear as logically assigned devices. So instead of the C:-drive, you will get /dev/hda1 in which ‘hda’ is your first available harddisk ('h' in hda is for ide drives, 's' is for scsi, sata and removable drives), hda1 is the first (primary) partition on which a filesystem may reside. The filesystem used in general for Windows is NTFS. This stands for New Technology FileSystem, but in the mean time the "New" in technology is already more than 12 years old. Nevertheless, it has gotten some improvements over the years and it is, I must admit, a good filesystem. The other filesystem natively supported by Windows is FAT, which come in the flavors FAT12 (for floppies), FAT16 (for small disks up to 2Gb) and FAT32 (for bigger disks).

What 's also different from Windows is that these filesystems are not accessible by default in Linux, you have to so called "mount" them. Where in Windows you will get a drive C: with your files on, in Linux you have to call the command 'mount' and mount the filesystem against a subdirectory. An example of this: let 's say your drive C: is /dev/hda1 (/dev contains the collection of device references on your system). You have a directory /mnt0. In this case you type 'mount /dev/hda1 /mnt0'. When you invoke the command 'mount' afterwards without any parameters, you will see that /dev/hda1 is mounted on /mnt0. If you cd to that directory and type 'ls' (=equivalant of 'dir' in Windows), you will get a directory listing of what 's available on that C: drive.

Now with this all explained, you should see the picture: TRK runs on your computer but treats it as a doctor inspecting a dead body: everything is there, but the person is not. You can now perform surgery on the body and try to revive it.

1.3 Getting around with common linux commands (cd, cp, mv, rm, more, grep, mount)

This is a small tour on commands you will find useful when working with TRK (and Linux in general). Let 's take as a convention that commands you have to type are put between 'single quotes'. At the commandline you omit these quotes (unless I say not to).

What I 'm going to teach here is basically how you work with files, like copying, moving, editing, etc...

First, let's start with changing directories.

People that have worked with the commandline in Windows or Dos will recognize many things. The big difference with this is that directories in Linux are separated by forward slashes instead of backslashes in Windows. Another big difference is that Linux folders and files are case sensitive: capitals have to be typed as capitals, otherwise the file or folder will not be found.

-cd

The command to change a directory (or folder how it 's called in Windows) is 'cd' f. i. you want to change to the directory /tmp you type 'cd /tmp'. If the folder contains spaces, there are two ways you can get into them: or you put the foldername between double quotes or you use so called escape characters to represent the spaces.

Let 's say we want to cd to the folder 'Documents and Settings' you can type 'cd "Documents and Settings"' or 'cd Documents\ and\ Settings' where the backslash (\) in Linux is used to 'escape' characters, i.e. you treat the characters litterally instead of as a control character. Enough of that.

Another way of easily changing directories is by typing the first letters of it and then pressing the tab-key. F.i. type 'cd Doc' and press tab. The command will complete as 'cd Documents\ and\ Settings'. If more files or folders match, the tab command will show you the possible options left. Very nice feature, saves you lots of time and painful fingers.

-cp

File copying, the second thing you 'll probably need.
Once you 're in the right directory, you need to be able to copy files to other locations.
Here 's how to do it, together with the most important switches.
Take a file called file1, located in /tmp/ and you want to copy it to /home: simple command 'cp /tmp/file1 /home' or when you first cd-ed to /tmp: 'cp file1 /home/'.

Source-target.
Suppose you want to copy multiple files, you can use a wildcard with '*'. If /tmp contains file1 and file2 and copy them both at the same time: 'cp /tmp/file* /home/'
If you want to copy files and folders recursively, together with all the attributes, use the switch '-a'
To see what gets copied while it 's busy, add the '-v' parameter too (verbose).
Say /tmp contains 'file1', 'file2', a subfolder called 'testfolder' which also contains 'file3', then perform 'cp -av /tmp/* /home' This will copy the complete contents of /tmp, including subfolders to /home.
If cp asks you to overwrite already existing files, you can force that by adding '-f' (force) to your command:
'cp -avf /tmp/* /home'

-mv

Moving files. This is the same principle as copying files but easier, e.g.
'mv /tmp/* /home' moves all the contents, including subdirs to /home
Adding the parameters -v and f moves them verbosely without prompting to overwrite existing files.

-rm

Remove files.
Remove 1 file, example: 'rm /tmp/file1'
Remove files recursively, without prompting: 'rm -rf /tmp/*'

-more

Viewing files and output of commands. This is a util you can use any time there 's too much output coming to your screen or you need to look into a file.
E.g.: 'more /tmp/file1' shows you the contents of file1, but gives it a page at a time. To go to the next page, press space. The arrows and enter key scroll down line by line.
You can use this command also in combination with other commands to halt their output so you can read what it says.
Example: 'dmesg | more' : 'dmesg' gives you the output of your kernel startup procedure and recent system messages, but it 's maybe about 300 lines of output. So in this case we so called "pipe" the output of 'dmesg' to 'more' using the "|" sign. In this way I have also explained you with an example the use of "command piping"

-command piping

Continueing on this subject, let 's see what other uses command piping can do for us. It can be used to filter out a certain line with a specific keyword.
Let 's say you want to know whether there 's a file called Document.doc' somewhere in a subdirectory, but you don 't know which. Then use this command from within the base directory you want to search in: 'find ./ | grep -i document.doc' (the -i parameter upper- or lowercase characters)
You can also pipe the output of a command to a file instead of the screen. F.i. to put the complete filelisting of a directory tree to a file, do like this 'find ./ > /tmp/filelist.txt'

-editing files

Here 's quickly how to use vi, the most common text editor on Linux. Beware: this does not edit Word documents or any other document format that is in binary format.
Open a file or create a new file: 'vi /tmp/file1'

Move your cursor around with your arrows to the line you want to edit.
To insert text, type 'i', this will put you in insert mode. To remove text, use 'x', (go out of insert mode first with escape). To remove or cut a complete line, use 'dd'. You can paste this line elsewhere with 'p'.
This is basically editing in vi. To save a document, go out of insert mode and type ':wq' (colon write quit). To exit without saving: ':q!' (colon quit exclamation mark).
If you don 't like vi, you can use pico which is a bit simpler to use, but less common on the different Linux systems.

-mounting filesystems

When working with Linux and more specifically here with Trinity Rescue Kit, it is imperative that you understand the way you "talk" to filesystems.
Whereas Windows just assigns a driveletter to any local filesystem it knows and finds (which is only NTFS and FAT), Linux does it all by invoking "mount" of a filesystem against a directory where you mount it.
Trinity Rescue Kit has a utility called "mountallfs" that searches for every filesystem on the local computer's disk drives and mounts it in a directory that has the same basename as the device where the filesystem resides. More on that later in this documentation. In other, normal Linux distributions, local filesystems are detected or created on install.

Let 's talk now how to perform manual mounting.
Mounting can be performed with any filesystem, regardless of it being local or on the network.

* Mounting a local filesystem can be done like this:

To know what device contains te filesystem you want to mount, you can look at a file called "/proc/partitions" This will tell you the partition lay-out of your disks, which will most likely contain filesystems. A common "/proc/partitions" file may look like this:

/dev/hda
/dev/hda1
/dev/hda2

/dev/hda claims in fact the whole disk. Under Windows it is impossible to create a filesystem in there, under Linux it is possible but improbable and not recommended. Most likely you will find a filesystem on /dev/hda1 and /dev/hda2, which will be you C: and D:-drive under Windows in general.
Mounting this is quite easy, in general you don 't have to give any paramters with it, Linux will detect the type of filesystem.
'mount /dev/hda1 /mnt0'

Trinity Rescue Kit by default has two directories for manual mounting of filesystems. You can create as many as you like, in as many subdirs as you like. That 's all I 'm going to explain about local filesystem mounting. I recommand you use "mountallfs". More on that later.

* Mounting network filesystems.

This is a very interesting bit, because with Trinity Rescue Kit you will want to evacuate your files to another computer. In TRK (and most other Linux distributions) it is possible to talk to Windows filesharing technology. For those who want to know the name of this technology, it 's called SMB (=Server Message Block). TRK can act as a client as well as a server. In this case we 're talking about TRK as a client.

Let 's say you have a running windows machine and you 've configured it to share "myshare". If you have not configured a share, you can connect to the c$ hidden share, but then you need to deactivate "Use simple simple filesharing" in the folder options of your Windows explorer. But let us take the "myshare" share.

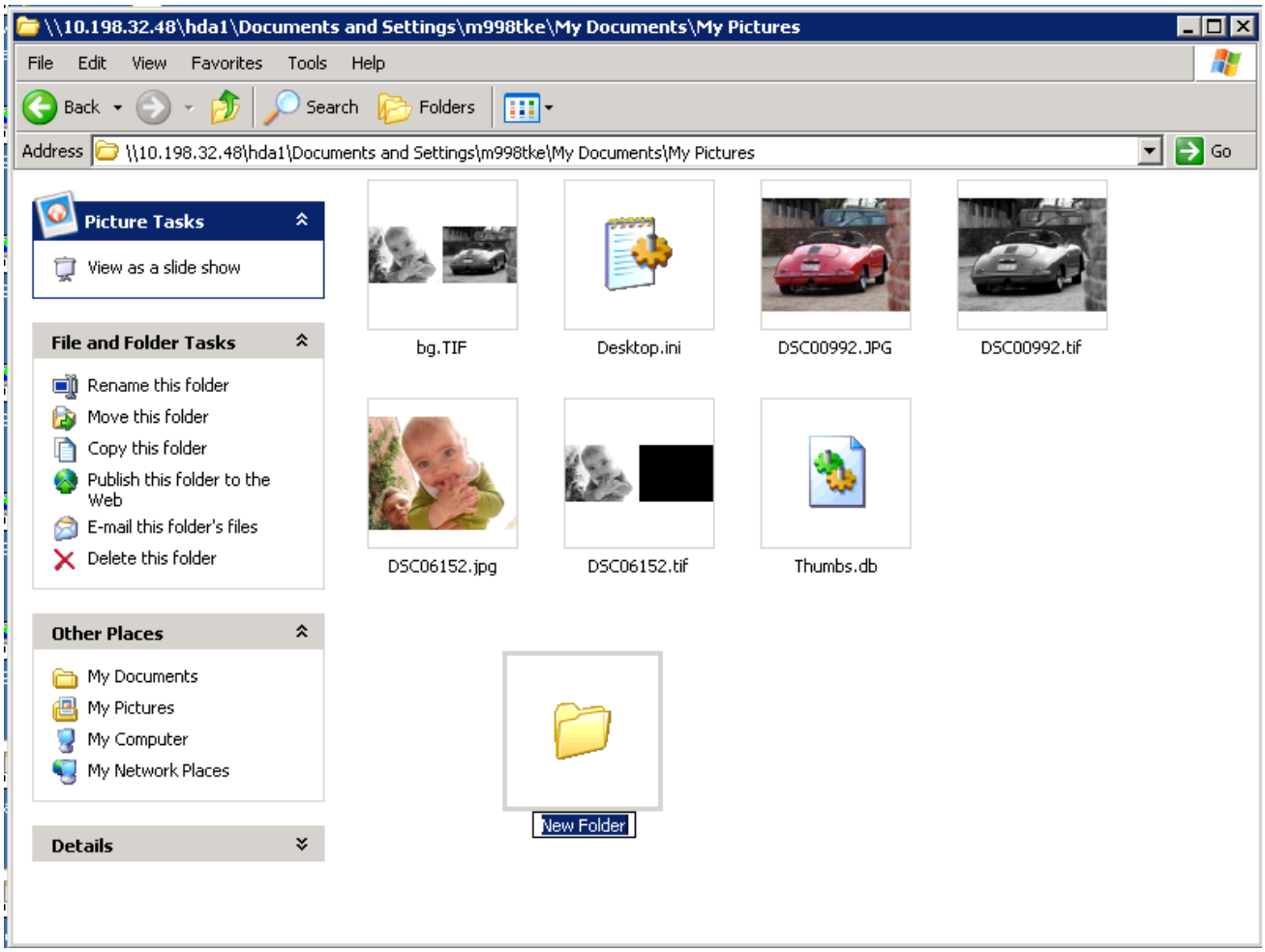
For the ease of use, it 's a good thing to create a user on your Windows machine called "root", give him a password and make it an administrator. But that 's not really necessary, you can also use the local "administrator" account, this will just require you to add a parameter to the mount command.

Let 's take the case of the "myshare" share, your Windows pc has 10.0.0.5 as ip-address (always faster to just point to the ip-address instead of the name) and you 've created a local user called root. Then here 's the command: 'mount //10.0.0.5/myshare /mnt0'

You will get prompted for a password in if no output is given, your share should appear under /mnt0. Make sure "myshare" has enough permissions for the user "root"

If you don't want to create a user, you don 't want to create a share and you did disable simple filesharing (or your windows machine is member of a domain), you can go ahead like this:
'mount -o username=administrator //10.0.0.5/c$ /mnt0' , which will prompt for the password an give you the credentials of "administrator". You can already pass the password (e.g. blahblah) in the commandline if nobody is looking over your shoulder: 'mount -o username=administrator,password=blahblah //10.0.0.5/c$ /mnt0'

Below is a screenshot of a Windows PC that has opened the disk of a remote TRK booted machine.



1.4 Reading information about your PC (dmesg, /proc/partitions)

Let 's summarize a little what 's already been said and look at reading info on your computer.

-/proc

Standard Linux always has a filesystem called "proc", which is a virtual filesystem in which files reside that have to do with your hardware and running processes. It 's a wonderful invention. We already talked about /proc/partitions, which is a file containing all local disk partitions.

Another interesting file is /proc/cpuinfo, which gives you information about your CPUs

/proc/meminfo gives you information about the memory usage. Don 't let yourself get misled by the Memfree line, which will always look very low. Actually, Linux

always reserves most part of the memory so it can make use of it in a fast way. What you need to look at is the Active and Inactive. The maximum amount of memory you will see will never be more than 4Gb, since the TRK kernel needs to keep maximum compatibility

Another useful file to read info on your cdrom drive is /proc/sys/dev/cdrom/info.

Those are about the important files in /proc you need to know about now.

-dmesg

The command "dmesg" gives you your kernel messages. Any hardware detected will give you a message somewhere in the output of this command. If you want to know the type of network card that has been detected, perform 'dmesg|more' and look for any mentions of eth0, eth1,...

What type of harddisk controller you have: dmesg. Just use it when you find yourself stuck on hardware questions. Also disk failures will be visible with this command. Network errors, link down, etc, one command.

Another way of looking at this information is through /var/log/messages (more /var/log/messages), which on normal Linux distributions contain output logs of previous boots too.

To know what device your newly inserted USB stick has, plug it in, let it settle for a few seconds and then run dmesg again. Or just run 'dmesg|tail' to see only the last added lines.
Here 's an excerpt of what you might read from dmesg. It tells you something about your network card:

```
eepro100.c:v1.09j-t 9/29/99 Donald Becker
http://www.scyld.com/network/eepro100.html
eepro100.c: $Revision: 1.36 $ 2000/11/17 Modified by Andrey V. Savochkin
<saw AT saw.sw.com DOT sg> and others
ACPI: PCI Interrupt 0000:05:08.0[A] -> GSI 20 (level, low) -> IRQ 16
eth0: OEM i82557/i82558 10/100 Ethernet, 00:08:02:C6:4E:9D, IRQ 16.
Board assembly 262285-001, Physical connectors present: RJ45
Primary interface chip i82555 PHY #1.
General self-test: passed.
Serial sub-system self-test: passed.
Internal registers self-test: passed.
ROM checksum self-test: passed (0x04f4518b).
```

-lspci and lsusb

This gives you any information on what 's on your PCI and respectively USB bus. This doesn 't only mean what 's in your PCI slots, but everything on the bus, so also onboard ethernet and usb controllers.

-lshw

Now here 's a great utility that can give you a complete listing of all your hardware, recognised and not recognised. When you run it, it will give you a LOT of output, so best here is to run it 'lshw | more', or if you only need specific info about f.i. disk drives, you can run 'lshw -C DISK'. Getting the info off your TRK can be done directly to the interweb (provided your network card got detected) by running 'lshw | wgetpaste', which will publish the output on http://pastebin.ca and return you a short url to where it can be found.

-smartctl

The smartmontools are part of Trinity Rescue Kit and not so common on normal Linux systems, yet they are a valuable addition to any system. What it does is read the s.m.a.r.t. information of disk drives so you can know when errors start to occur.

Just use it like this: 'smartctl -a /dev/sda' where sda is your first scsi or sata drive. Make sure smart is enabled in the computer's bios.

-acpi and acpitool

Two tools to read the battery and thermal information of your computer. Type acpi --help to get more info on possible options.

acpitool can give you much more information like fan speeds and cpu. Also certain laptop types are supported for their special features like brigthness on Asus laptops etc...

-df and du

Two standard utils provided in Linux. df shows you the usage of your mounted filesystems, du shows you the usage of a specific folder. Use it like 'df -h' and 'du -h' where "-h" stands for "human readable", making the output rounded to mega- and gigabytes.
This is in short how you can get to know your computer a little and how to jumpstart using Linux and Trinity Rescue Kit.

-lshw

Recently recommended and added, but looks very promising, lshw gives you a complete list of all your hardware in your computer. Best to pipe this to a file, because the list can get long.

2. TRK own commands and utils

Let 's look at the added value of Trinity Rescue Kit, with its own specific commands and utils.

2.1 Virusscan

Virusscan is a script that actually wraps 5 different virusscanners into one.
Only one of them is actually included on the TRK cd (Clamav), the others are downloaded from their website upon usage.

When running virusscan, it is highly recommended that your computer has a decent internet connection so you can get the latest virus signatures.

Scan engines

Currently, 5 AV engines and md5 file checksumming are implemented.

-ClamAV

This is the basic engine provided and is already preinstalled on TRK. It is very effective on mailservers but is quite slow and tends to crash when used as a commandline scanner. It also focuses more on mailworms and, from experience, has less effectiveness for local viri. Clamav is the only GPL licensed AV engine implemented. All others have some sort of free-for-non-commercial-use license and are closed source. The pros of Clamav are:
* very quick on new virus outbreaks
* included in TRK
* GPL licensed, so free for everyone
The cons:
* slow and very CPU and memory intensive
* detects the least viri of the 5 scanners in virusscan.

Because it's in fact a mailserver scanner, it will focus more on worms than on filth that comes from malicious websites and such.
* cannot disinfect inside files on its own. What is done in this case is quarantine the infected files into a tar.gz archive in <scandestination>/TRK-INFECTED/. Should a file be accidentally deleted, you can recover it afterwards and rescan it with another antivirustool

-F-Prot

This antivirus tool and all the others are not included in TRK but get downloaded from the Internet as soon as you call upon them. They disappear after a reboot of TRK. If you want them to be available after a reboot, you have to run updatetrk. This will be explained later in this documentation. The pros of F-prot:
* lightweight, not a big download
* pretty fast, low cpu usage
* good disinfection method
The cons:
* does not detect everything
* their website sometimes fails and download of f-prot is aborted

-BitDefender Scanner

It has a good average between filesize, cpu/memory load and virusdetection. It can detect many different types of malware. From what has been experienced so far, it may detect other viri and malware than the other 4. It's recommended to sweep with this after another one has already run.
Pros of BitDefender Scanner:
* detects quite some viri
* pretty fast
* detects alternate malware
Cons:
* sometimes doesn't detect very common viri
* slow update process

-Vexira

This AV engine hasn't been tested so much, but it looks like a good average AV engine.

-Avast

Avast is the latest addition to virusscan (and replaces Grisoft AVG because AVG lacks cleaning support in its new version). Avast is a great AV on Windows, very lightweight, but has not been tested in depth yet on Linux/TRK.
For this particular AV engine you need a registered, free license key which is sent to

you by mail.
Get it at http://www.avast.com/registration-free-antivirus.php
If you want to avoid entering the license key each time, it's recommened to run updatetrk

-MD5

This is not an antivirus engine but just reads all of your files and makes md5sums of it. It writes the result to a logfile in the same way like it does for an AV engine. The logfile format is: modification seconds since 1-1-1970 <space> md5sum <space> filepath.

To get complete and up to date info, please check out the online version of the manpage for virusscan: http://trinityhome.org/manpages/man8/virusscan.8.html

2.2 Winpass: reset your Windows XP - Vista - Seven password

Winpass is a bash wrapper script for **chntpw** supplied with Trinity Rescue Kit that resets MS Windows NT based (= NT, XP, Vista, Seven, 2003, 2008) local passwords. If you forgot your password, you can have it removed with winpass.

By default winpass without any arguments will reset the builtin administrator account of a locally installed Windows, but you can specify other accounts as well at the commandline. In fact, you can add any parameter from chntpw which will be parsed to the commandline. So winpass -l will list all usernames found in the SAM (=Windows user and password database). Should you have troubles that metacharacters are present in the username (such as the Ã˜ or something), you can still use the HEX reference to the username listed next to it. Be sure to prepend that with a '0x'. More info on that can be found in the chntpw manual.

Winpass does not reset any Active Directory passwords

-l
: list usernames contained in the SAM of the local computer and exit
-i
: interactively run chntpw. This option lists the local usernames and gives you the option to choose from them
-e, regedit
: run as registry editor. To get a list of commands in the editor, type "?"
-u "username"
: optional username if the user you wish to reset is not "Administrator". Be sure to add quotes for the username if it contains whitespaces
--restore
: restore the original SAM file, thus restoring the original password/user situation from before the first time winpass was ever run. Should you somehow have messed up the user accounts, you can go back to the original situation with this option. Since build 338 the option to overwrite the backup when winpass is run multiple times has been disabled to prevent yes-men users

answering yes to every question being asked. Removing the backup must now be done manually. You can find it in general under /sda1/WINDOWS/system32/config/SAM

Example:

Here's an example of resetting the password for user "John Doe". Note the double quotes around the username.

```
[root@trk]:(~)# winpass -u "John Doe"
Searching and mounting all filesystems on local machine
Remounting NTFS partitions with ntfs-3g
Result of mounting:
/dev/hda1 on /hda1 type fuseblk (rw,allow_other,blksize=4096)
Windows NT/2K/XP installation(s) found in:
1: /hda1/WINDOWS
Make your choice or 'q' to quit [1]: 1
Ok, continue
chntpw version 0.99.6 080526 (sixtyfour), (c) Petter N Hagen
Hive <SAM> name (from header):
<\SystemRoot\System32\Config\SAM>
ROOT KEY at offset: 0x001020 * Subkey indexing type is: 666c
Page at 0x8000 is not 'hbin', assuming file contains garbage
at end
File size 262144 [40000] bytes, containing 7 pages (+ 1
headerpage)
Used for data: 317/24808 blocks/bytes, unused: 6/3640
blocks/bytes.


Hive <SECURITY> name (from header):
<\SystemRoot\System32\Config\SECURITY>
ROOT KEY at offset: 0x001020 * Subkey indexing type is: 666c
Page at 0x10000 is not 'hbin', assuming file contains garbage
at end
File size 262144 [40000] bytes, containing 15 pages (+ 1
headerpage)
Used for data: 1108/53920 blocks/bytes, unused: 11/7040
blocks/bytes.




* SAM policy limits:
Failed logins before lockout is: 3
Minimum password length : 7
Password history count : 7
| RID -|---------- Username ------------| Admin? |- Lock? --|
| 01f4 | Administrator | ADMIN | |
| 03eb | ASPNET | ADMIN | |
| 01f5 | Guest | ADMIN | dis/lock |
| 03e8 | HelpAssistant | | dis/lock |
| 03f0 | John Doe | ADMIN | |
| 03ea | SUPPORT_388945a0 | | dis/lock |


---------------------> SYSKEY CHECK <-----------------------
SYSTEM SecureBoot : -1 -> Not Set (not installed, good!)
```

```
SAM Account\F : 1 -> key-in-registry
SECURITY PolSecretEncryptionKey: 1 -> key-in-registry
Syskey not installed!


RID : 1008 [03f0]
Username: John Doe
fullname: John Doe
comment :
homedir :


User is member of 2 groups:
00000221 = Users (which has 4 members)
00000220 = Administrators (which has 7 members)


Account bits: 0x0010 =
[ ] Disabled | [ ] Homedir req. | [ ] Passwd not req. |
[ ] Temp. duplicate | [X] Normal account | [ ] NMS account |
[ ] Domain trust ac | [ ] Wks trust act. | [ ] Srv trust act |
[ ] Pwd don't expir | [ ] Auto lockout | [ ] (unknown 0x08) |
[ ] (unknown 0x10) | [ ] (unknown 0x20) | [ ] (unknown 0x40) |


Failed login count: 0, while max tries is: 3
Total login count: 0


- - - - User Edit Menu:

1-Clear(blank)userpassword

2-Edit(setnew)userpassword(carefulwiththisonXPorVista)

3-Promoteuser(makeuseranadministrator)
(4 - Unlock and enable user account) [seems unlocked already]

q-Quiteditinguser,backtouserselect
Select: [q] > 1
Password cleared!


Hives that have changed:

#Name

0<SAM>-OK


Backup file already exists. Not touching this file. Please be
aware that 'winpass --restore' would restore the very original
file from before winpass was ever run
Writing /hda1/WINDOWS/system32/config/SAM
[root@trk]:(~)#
```

For the full documentation, refer to the online manpage:

http://trinityhome.org/manpages/man8/winpass.8.html

regedit

This is actually the same script as winpass, but called in this way it copies all registry hives and opens them (sam, security, system and software). Only drawback is that it cannot know which user registry to open, since they are located in different directories. Because chntpw is not scriptable, I cannot read registry hives from a shell script to determine the location of userhives.

Once running, refer to the documentation of chntpw/regedit on how to use it.

2.3 Mass Clone: a multicast disk cloning tool

Mclone or Mass Clone is a utility that creates and distributes harddisk images over the network via multicast.
It is very fast since it uses only one network packet for an infinite number of receivers. On a 100mbit switch it will average at about 93mbit. The only limitation here is the speed of the network, disk or cpu.
Main features:
-make exact copies of any operating system
-optimized for Windows XP/Vista/Seven imaging using ntfsclone. Other filesystems are copied with dd
-fast and scalable
-save to image and restore from image (to multicast) with optional 3 compression algorythms (gzip, bzip2 and 7-zip)
-restore original bootsector/ntfs c/h/s values. An old bug in many BIOSes sometimes gave wrong values for Cylinders/Heads/Sectors count. Although CHS is an old method for assigning disk geometry (LBA should be used), Windows XP and family still use it to assign addressing of their bootcode. Recent Linux kernels discard wrong C/H/S values and set it to the LBA values. This resulted sometimes in unbootable cloned Windows machines (the blinking cursor nightmare). Recently a patched version of relocntfs appeared (now called ntfsreloc) which is able to "force" original C/H/S values into your NTFS. Mclone does it automatically for you. Major feature over other cloning tools.
-run up to 50 different sessions separately over your LAN
-optional speed limitation. Just so your LAN doesn't get saturated.
-option to specify disks/partitions instead of just everything automatically
-option to skip bootsector
-option to skip C/H/S check

Since TRK is network bootable from another TRK without any modifications to your LAN's config, you could just boot one TRK from CD/usb stick, boot all your other computers over the network and run mclone on all of them.

An average image of about 4Gb on a 100mbit network is cloned in about 7 minutes

USAGE

In short: the computer to be cloned (the sender) runs 'mclone -s', all others run 'mclone' (the receivers).
Once all computers are ready and waiting, just press enter on one of them and cloning begins.

You can also save to an image file. The computer that will save the image as a set of files runs mclone with the option -o.
For example:

mclone -o standard-xp-install

will save an uncompressed image to "standard-xp-install", creating it as a directory in the current directory if it does not exist.
The command:

mclone -C 7-zip -o standard-xp-install

will add 7-zip compression to the image. But beware of compression. I've noticed that gzip is the only compressor that doesn't eat too much of your CPU so not to lose transfer speed. If size is more important to you, use bzip2 or 7-zip
In all of these cases you run 'mclone -s' on the computer from which you wish to create an image. (The Sender.)
Restoring from this recently created image is done with:

mclone -i standard-xp-install

No need to specify the compression anymore, mclone will use the correct decompressor.
So remember:
-o mode and 'mclone' without arguments is a receiving mode, so the other side must run mclone -s (sender mode) -i and -s mode are sender mode, the other side runs 'mclone' as a receiver

Here are all the options that you can specify.

The command: mclone without any arguments runs in client mode, meaning it will RECEIVE an image from a sender

-s
: SENDER mode. This mode will send the contents of the local harddisks to listening clients
e.g.: The sender runs 'mclone -s', the clients run 'mclone'. The sender will clone a one to many copy of itself

-o <path-to-image>
: This mode is a RECEIVER mode that stores an image from a sender to image files in the path specified by the argument 'path-to-image'
e.g.: 'mclone -o /data/xp-image'. Note: the dir does not need to exist, but please erase it if an image is already in there.

-i <path-to-image>
: This is a SENDER mode that sends an image to clients. You first need to have an image created with mclone -o of course.

e.g.: 'mclone -i /data/xp-image'

-n <session number>
: Specify as many as 50 sessions when performing multiple cloning. Client and sender need to specify the same number

e.g.: 'mclone -n 5' for client, 'mclone -s -n 5' for sender

-h
short help screen

Optional arguments **ONLY FROM SENDER MODE**:

-d <disks>
Skip disk detection and specify your own (comma separated if more than one). Only valid for SENDER mode (-s).
e.g.: 'mclone -s -d /dev/sda,/dev/sdc'
-p <partitions>
Skip partition detection and specify your own (comma separated if more than one.) Only valid for SENDER mode (-s).
e.g.: 'mclone -s -p /dev/sda1,/dev/sda2'
-c
Skip C/H/S check. This should not be necessary, but if you run into troubles with booting, it might help, although the opposite is more likely to be true
-b
Skip save bootsector. Should you not want to overwrite your bootsector, add this parameter.
-t <timeout>
set the timeout between the first client to connect and the last one to ride the train. This option is only used during actual image creation. Default is 10 seconds.
-r <bitrate>
Set the maximum bitrate. Set it in kilobit (k) or megabit (m). This is recommended when you are on a shared lan, because mclone will eat all the available bandwidth.
e.g.: 'mclone -s -r 80m' sets a maximum of 80 megabit or 10 megabytes per second (which is ok on a 100mbit switch)

Optional arguments for **IMAGE SAVE MODE**
-w
Use network compression. This option uses more CPU but saves bandwidth on your network. Use only on powerful machines. This option is permanent when you store to an image file, i.e. when enabled at image creation it will always be used during restore.
-C <compressor>
Save your image with compression. 3 compression methods are available (specified as written here): gzip, bzip2 and 7-zip e.g.: 'mclone -o /data/xp-image -C gzip'

For the full documentation, refer to the online manpage:

http://trinityhome.org/manpages/man8/mclone.8.html

2.4 Winclean

With winclean you can easily clean out unnecessary files, such as tempfiles or the contents of recycle bins, from a Windows based computer.
This utility will call upon mountallfs to search for all local filesystems.
It can clean several types of (practically) useless files.
The file location types are:
-Tempfiles from every user, located in %TEMP% (e.g.
C:\Documents and Settings\Username\Local Settings\Temp) and Temporary

Internet Files from Internet Explorer. The contents of the folder C:\Windows\Temp is also cleaned out.
-Recycle Bins (of all users) can be emptied as well
-Uninstall files from cumulative patches and service packs.
This is only valid for Windows XP (and 2000), not Vista or Seven (there it's interweaved in WinSXS and is too dangerous to touch). The removal of these redundant files can provide a significant performance increase since a lot of directory enumeration is done in the Windows system folder and the more files and folders there are, the slower it becomes. These folders are hidden by default in Windows explorer and are named similar to $Uninstall-KB123456$ for example.
-Dllcache: C:\Windows\system32\dllcache can contain more than 1000 files and take up several hundreds of megabytes.
This folder has little use if your machine is running as it should.
-Hanging printerjobs: sometimes spoolfiles are corrupt, or a printer has been physically removed for a long time or a spooljob is hundreds if not thousands of megabytes in size because someone tried to print a 10 megapixel picture of 1200dpi. Winclean can remove all pending jobs.
-Java cache: The cache files of java applications can take up hundreds of megabytes, sometimes even more. These can be safely deleted.
This option is currently only for Sun Java.

USAGE

winclean -d <destination mount point> -s <value> -v -R -f -a -t -r -u -c -p -j
where:
-d <destination mount point>: if not specified, mountallfs will be called
-s <value>: safe deletion of only tempfiles older than <value> days.
-v: verbose deletion of files
-R: report first which files could be deleted
-f: force continue without prompting
-a: clean all categories or...
-t: tempfiles
-r: recycle bins
-u: remove uninstall patches information
-c: Windows dllcache
-p: hanging printerjobs
-j: java cache

For the full documentation, refer to the online manpage:

http://trinityhome.org/manpages/man8/winclean.8.html

2.5 Mountallfs

This script is about the most handy addon to TRK. It 's a script that scans all local storage devices and mounts their filesystems to a standard mountpoint, nl. the name of its device name.

mountallfs is a script that searches all local partitions on all local drives for any filesystem and tries to mount it. It supports all kernel based filesystems as well as fuse-ntfs and ntfs-3g. Furthermore it has support for LVM volumes. Currently it also supports pseudo-hardware raid volumes found on many desktop sata controllers. umountallfs is its counterpart and unmounts all filesystems mounted by mountallfs. Always run umountallfs before resetting your computer if you don't intend to do a clean shutdown.

-g
: Mount your ntfs filesystems with the **ntfs-3g** driver. This is the default behavior since build 338 because it allows full read/write support on ntfs drives. Performancewise it has much more cpu overhead than the kernel based ntfs driver. Most modern PCs shouldn't have too many problems with it however. Please note that the mount result will mark the ntfs filesystems mounted as 'fuseblk', since this is a userland driver which interfaces with the kernel fuse module.

-f
: Mount your ntfs filesystems with the **ntfsmount fuse** driver. This is somewhat the predecessor to ntfs-3g (hence ntfs-3g is the third generation ntfs driver). This fuse option has more limited write support than ntfs-3g and is deprecated over ntfs-3g. The mount result also shows as being mounted as 'fuseblk'.

-k
: Mount your ntfs filesystems with the **kernel ntfs** driver. This is a read-only driver but uses less CPU resources and so has better performance. This was the default behavior before build 338 and is now a new option.

-l
: Activate Logical Volume Management. This option is for computers with LVM volumes and pseudo-hardware raid controllers. If you have such a pseudo-hardware raid controller and you configured your disks in mirror mode, ALWAYS use this option before doing any operation on your disk because otherwise each mirror member might get mounted separately and you **WILL** corrupt your filesystem on it. Default behavior for this option is also to mount ntfs volumes with ntfs-3g.

-q | --quiet
: Do not display any output from the mount result.

For the full documentation, refer to the online manpage:

http://trinityhome.org/manpages/man8/mountallfs.8.html

REMARKS

-If your filesystems were not found it might either mean they were too corrupted to mount or your disk controller was not detected by TRK. You might try more (scsi) drivers by typing 'tryscsi'. This is more the case on servers with more advanced disk controllers. However, 'tryscsi' might freeze your machine. Be warned!
-Mountallfs might also warn you that your ntfs volumes have been dismounted improperly. This is most likely the case when a Windows session was shutdown improperly. Mountallfs will prompt you to forcemount the ntfs volume, which invokes an ntfsfix and triggers a chkdsk at next reboot into Windows. Please, let this chkdsk run. It will solve about half of the possible problems that might occur on your ntfs volumes.
-Another thing you might encounter is that the volume (Windows) is hibernated and a hiberfil.sys is present. In that case you will be prompted to either remove the hiberfil.sys and forcemount the drive or leave it alone. Mounting the drive and leaving the hiberfil.sys could corrupt your Windows and make it crash when you resume from hibernation.
-/etc/mountallfstab is the separate config file mountallfs keeps after mounting. It uses it for umountallfs to unmount all the volumes previously mounted by mountallfs. Umountallfs also does a test that there are no more open processes on the volumes (most of the times your working directory of your shell). However, it does not take into account any mountpoints under your mountallfs mounted volumes. So you need to take care of those manually.

EXAMPLES

Command:'mountallfs -l' searches for LVM volumes and mounts the filesystems inside them.

Command:'mountallfs' without any option mounts all filesystems and remounts ntfs filesystems with ntfs-3g

2.6 Updatetrk

Trinity Rescue Kit is able to **update itself or better said, add features** through the script 'updatetrk'. This requires decent Internet connectivity, direct, over a router or via a proxy server (in that case, run '. setproxy' first)

The script is very simple to handle:
Suppose you 've booted from a PC with a Windows XP (2K or NT is also good), you can run updatetrk without any option.
Since TRK 3.4, updatetrk uses a script called 'getswap' which looks for extra virtual memory on local swap partitions and Windows pagefile.sys

Updatetrk fetches external third party, non GPL software (fetches and updates all AV engines for virusscan) and creates a new TRK isofile or updates your TRK USB stick. It also saves any changes you made to your running copy of TRK. It allows you to install tarballs and even rpms and have it saved for the next time you run TRK. An example of such software is IBM Tivoli Storage Manager client. Adding this software to TRK allows you to do offline backup/restores from a TSM server.
Updatetrk can create a new isofile which you can burn or directly update your USB bootmedium. Given that you run TRK on a computer that normally runs Windows, it will use the pagefile.sys as extra swapspace on which to create its new image.

USAGE

updatetrk -b [BUILDTARGET] -i [TARGETISOFILE] (or) -u [TARGETUSBDEVICE] -s -a avs,bde,clam,fprot,va -f

-b [BUILDTARGET]
: Build location where intermediate files can be stored. Should be at least 1100mb. Make sure no other files reside in that location, as they will be included as well. If no build target is given, a script called 'getswap' will be invoked which will search for swap partitions and Windows pagefile.sys files. This pagefile will be added as swapspace and extend youir working memory. This has no consequences for your Windows system afterwards. It will just be reused.

-i [TARGETISOFILE]
: Target dir on which to create the isofile. There should be at least 350Mb free on the target. The filename will be automatically given. If this option is omitted, the file will be created in the same location as the buildtarget. If the buildtarget is omitted, it will be created in the temp dir of the drive containing the pagefile. In general this will be c:\temp under Windows.

-u [TARGETUSBDEVICE]
: Target USB filesystem to update. This performs a simple copy to the location you specified. In general this is the partition from which you booted TRK.

: Example 1: 'updatetrk -b /hda1/temp/trkbuilding/ -i /hda1/Docs/' This will use /hda1/temp/trkbuilding/ as construction site and create /hda1/Docs/trinity-rescue-kit-3.4-363u.iso

Example 2: 'updatetrk -u /dev/sda4' will search for a local pagefile as construction site and copy back the files to your USB device (or fixed harddisc)
If you specify no options at all, TRK will always be created on a local pagefile as an ISO in c:\temp. If no pagefile is found, the script will exit. If you specified a specific buildtarget, the files will remain there afterwards at your convenience.
Remark: option '-i' and '-u' are not useable together.

-s avs,bde,clam,fprot,va
Skip the inclusion of specific AV engines+updates. The syntax speaks for itself. If you skip all AVs, updatetrk will only apply what is currently already modified on your running copy. -f Force updatetrk to build, even if not enough tempspace is available. It is possible to assign more of the swapspace to /dev/shm (the default volatile temporary storage location of Linux). Running 'getswap -s 80' f.i. will assign 80% of the total memory (ram+swap) available to /dev/shm. With 'df -h' /dev/shm you can verify if you arrive at 1.1Gb free space on this location.

-a
Copy contents of / (root) completely and “as is” instead of cleaning up and omitting certain directories. This will include logs and bash_history as well.

EXAMPLES

Example 1: 'updatetrk -b /hda1/temp/trkbuilding/ -i /hda1/Docs/'

This will use /hda1/temp/trkbuilding/ as construction site and create /hda1/Docs/trinity-rescue-kit-3.4-363u.iso

Example 2: 'updatetrk -u /dev/sda4'

This will search for a local pagefile as construction site and copy back the files to your USB device (or fixed harddisc)

If you specify no options at all, TRK will always be created on a local pagefile as an ISO in c:\temp. If no pagefile is found, the script will exit.
If you specified a specific buildtarget, the files will remain there afterwards at your convenience.

REMARKS

-Option '-i' and '-u' are not useable together.

-If you're behind a proxy server, run '. setproxy' first

-If you're updating your running USB stick, hard **RESET** your computer as soon as updatetrk has finished, since the underlying filesystems will have changed without having been remounted.

-TRK has an empty rpm database (except for 1 package that does some basic “provides”), meaning that almost any RPM you install will protest about missing dependencies.
In many cases, the binaries would work anyway.
The easiest way to test this is by installing the rpm in this way:

Type the command: 'rpm -ivh --nodeps <package.rpm>'

Then test it by executing the binaries. If it fails because certain libraries are missing, run ldd /usr/bin/mybinary and check what library is missing. Go and look for that library on rpmfind.net and install it.

Many commercial rpms are compiled in such a way that they are compatible with most Linux distros. Most of them require glibc 2.2 or higher. TRK is glibc 2.3.4.

-If your CD/RW drive is currently available (booted from RAM or usb stick), you can directly record the isofile.
For a blank cd, run f.i. 'cdrecord trinity-rescue-kit.3.4-build-363u.iso'
If you still need to erase your rewritable CD, first run 'cdrecord -blank=fast'

What will 'updatetrk' do?

-First ask for license agreements to install commercial software

-Then, it downloads the latest engine + virus signatures for ClamAV

-Next, it downloads F-Prot + updates

-As third antivirus, it fetches Avast Antivirus + updates. You will need a valid license key handy (free registration on site)

-4th antivirusscanner is BitDefender and starts the installation procedure. This is BitDefender 's own install procedure, just follow instructions on the screen, type "accept" for the license agreement and leave everything to the default propositions.

-The last virusscanner it fetches is Vexira. No intervention is needed here.

-Finally, it copies the complete contents of /bin, /sbin /etc and /lib to your new TRK. This means that anything you modified or added in these directories will be included in your new TRK. This gives you an easy way to make small modifications/additions to TRK.

After that, updatetrk recreates its /usr filesystem squashfs image. Since this image is orignally mounted with a pseudo read/write aufs over it, volatile changes will be submitted in the new squashfs image.

Also the initrd is recreated and everything from /bin /sbin /etc /lib and /var/lib is copied to it. Logfiles and session-based changes are discarded, except if the option -a was added to the commandline. Then everything will be copied back “as is”, including bash history f.i.

This script can be run time after time to keep your antivirus signatures up-to-date.

When no destination parameter was given to TRK and you 're running TRK from CD, it will move the newly created isofile to the drive where it found your pagefile.sys in directory 'temp'. Most likely this will be C:\temp.

To get complete and up to date info, please check out the online version of the manpage:

http://trinityhome.org/manpages/man8/updatetrk.8.html

2.7 Trk2usb

This utility puts your **Trinity Rescue Kit to a USB stick/disk** or even to a fixed harddisk. This is one of the two methods you can use to get TRK running from USB. This is also the recommended and easiest method of doing it. Only downside here is that you first have to burn the iso version of TRK to CD, boot from it and run it from there.

When given without the noformat option, it will destroy all data on the destination device, create a fat filesystem in the 4th partition id of the disk. This gives the best compatibility with most BIOSes.
It is also possible to do a non-destructive transfer to your medium given the -n option.
The destination medium has to be at least the size of the TRK isofile. Recommended is 256Mb minimum.

USAGE

trk2usb -d [DEVICE] -s [SIZE] -n

-d [DEVICE]
: Specify the destination device. In case you use it without the noformat option, you need to give the complete disk as argument, not a partition. Your disk will be erased, zeroed out and formatted with one partition as FAT16 of maximum 1Gb as the 4th primary partition. This is for maximum compatibility.

  Example: trk2usb -d /dev/sdc

-n
: Noformat. Use this option if your device is already correctly formatted and you don't want to lose your data on it. This requires you to specify the destination as a partition.

  Example: trk2usb -n -d /dev/sdc1

-s
: Optionally, you can specify the size of the destination partition in Mb. This allows you to add more partitions later. This option is not combinable with -n

Read the full and updated manpage online here:

http://trinityhome.org/manpages/man8/trk2usb.8.html

2.8 Trk2iso

On special demand, I 've created this small utility that creates an ISO file of your currently running TRK. It doesn 't update trk, it just creates an isofile in the current directory from where the command was launched. Here 's how to use it best:
'mountallfs -g' => suppose /dev/hda1 is your ntfs c:-drive
'mkdir /hda1/temp'
'cd /hda1/temp'
'trk2iso'

Once the isofile created, you can reburn TRK to CD.

2.9 Fileserver

In short: **share your drives like a windows fileserver**
Fileserver is a bash script that calls mountallfs and creates Windows fileshares from your mountallfs mounted volumes. It can run in guest-mode or user secured mode. It can optionally also run read-only.
The default netbios name will be "TRKSRV" and the workgroup will be "WORKGROUP". The server will not register in any WINS or DNS servers, so for it to appear in your network neighbourhood might take some time because it will use broadcast.
For quick access to it from an external PC running Windows, it's recommended to go to start => run (or "Start Search" in Vista or Seven) and type \\1.2.3.4 where 1.2.3.4 is the ip-address of your TRK. Fileserver mentions the ip-address in green when it starts. Fileserver calls mountallfs to search and mount all local filesystems.

Fileserver can be called in two modes: in **secured** or in **guest** mode (+ optionally read-only mode)

-**Secured mode** is invoked with the command 'fileserver -s'
This will prompt your for adding a user and a password that will have access to your files remotely. Just follow instructions. Here 's how the output would be if you ran it

```
Starting a username/password secured Samba fileserver and
sharing all local filesystems
Enter a username which will be created to have access to your
local files:
New SMB password:
Retype new SMB password:
Added user testuser.
Mounting all your local filesystems using mountallfs -g
These are the IP-addresses your fileserver will listen to:
192.168.81.5
Starting SMB services: [ OK ]
Starting NMB services: [ OK ]
```

-**Guest mode** is invoked with the command 'fileserver -g'
Use with caution, it allows everyone with network access to the computer to go on the local harddiscs. Use only in a trusted environment with a firewall or not connected to the Internet.

USAGE

-s

Start a secured samba server, i.e. one that does not allow guest access. 'fileserver' will prompt you for a username and a password. This is the default if no arguments are given.

-g
Run a guest enabled samba server. This will give anyone full access to the disks of your TRK running computer. Use only in an environment you can trust!

-r
Shares are read-only.

stop
Stops the samba server and unmounts the local filesystems.

EXAMPLE 1

TRK to TRK use of fileserver.
Suppose that we have two PCs both running TRK on the same network.
We run fileserver on one PC which tells us that its ip-address is, for example, 1.2.3.4 in green characters.
Then, from the other PC, we give the command: mount //1.2.3.4/hda1 /mnt1 (fileserver started as guest)

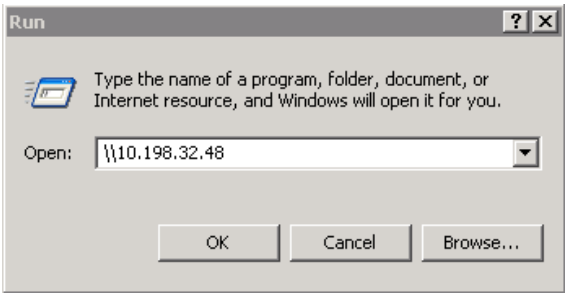
or

mount //1.2.3.4/hda1 /mnt1 -o user=your_user_name (secured) A password will be requested.
In guest mode, just hit the Enter key
In secured mode, enter the applicable password and hit Enter.
The selected partition (in this case hda1) of the remote PC (the fileserver) can now be accessed locally as /mnt1.

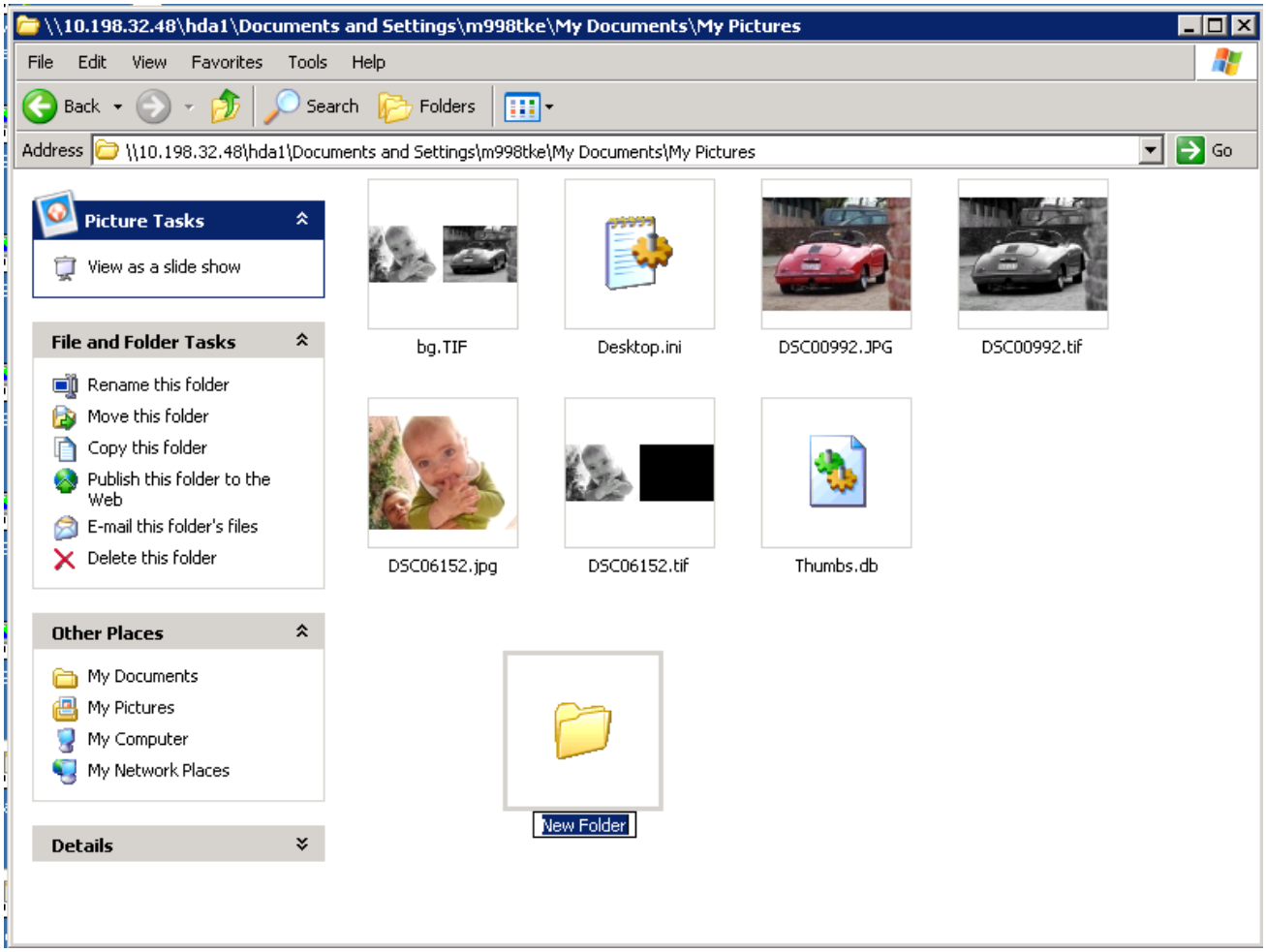
EXAMPLE 2

TRK as seen from a Windows machine
Run the fileserver from the TRK 3.4 simple menu

On your windows machine, click Start => Run (Vista and 7 "Start" => "Search" or winkey +r)



Browse the TRK fileserver, read and write access.



2.10 Bridge

Suppose you have a PC of which of which you like to know what traffic it generates. Or suppose you have a router on which you like to know what traffic passes by. Well, "bridge" will create you a **transparent connection between ethernet cards** that passes all traffic and on which you can start sniffing what passes by. You could use an old fashion hub that replicates all traffic to all of its ports, but nowadays that is almost inexistent, almost any cheap hub is in fact a switch that only replicates traffic to the port it is needed. In that case, put a computer with TRK in between your connection.

What you need

is a computer with at least 2 network cards and the latest TRK to boot from.

-Once the network cards are connected on both sides (on on the PC, one on the network), **run 'bridge up'**. This will setup a bridge and traffic will transparently pass from one side to the other. One note here which is quite important: make sure only 1 network card has its own ip-address, let the other one without (if they both have one, do an 'ifconfig eth1 down' f.i.). I will explain this later.

-Now, start sniffing with tcpdump. Run it like this: 'tcpdump -i eth0' In this way, you see ALL traffic headers that pass by. If you need to filter something specific, use "grep" to get it out. F.i. you need to see all http traffic going to 10.0.1.20, run this: 'tcpdump -i eth0 | grep 10.0.1.20 | grep http'

About the single ip-address: I have noticed that if you connect two network cards to the same network and let them both get an ip-address through dhcp (which will subsequently be addresses in the same network range) and afterwards connect one to a pc to start sniffing, tcpdump will go berzerk and no output will be shown but your TRK will give the impression of hanging and will not respond to a ctrl+c signal very quickly. After about 10 seconds, it will stop. But you will have noticed that sniffing doesn 't work. Well, if one card doesn 't have an address, this problem doesn 't occur. So leave at least one nic down. You can even leave both down, you don 't need an ip-address to sniff the network, "bridge" will bring up both cards in promiscuous mode and make a bridge interface "bro".

To bring the bridge down, just type 'bridge down'

Also read the online manpage on this:

http://trinityhome.org/manpages/man8/bridge.8.html

2.11 Setip

Use this whenever you want to enter a fixed ip address for your network cards

Usage

setip <interface> | -h
This command sets the ip information of your network card(s)
Without an argument, it assumes your network card is eth0
'setip -h' shows the help output
Here 's what the command does:

```
'setip eth1
Enter the IP-address for eth1: 10.0.0.1
Enter the subnet mask for eth1: 255.255.255.0
Enter the default gateway, <enter> for none: 10.0.0.254
Enter the IP-address of the DNS server you 'd like to use,
<enter> for none: 10.0.0.3

Setting ipaddress 10.0.0.1/255.255.255.0 on eth1
Setting default gateway to 10.0.0.254
Setting nameserver 10.0.0.3 in /etc/resolv.conf'
```

Also read the online manpage on this:

http://trinityhome.org/manpages/man8/setip.8.html

2.12 Setproxy

This script prompts you for proxy server settings, should you be unable to go directly onto the Internet. Use it without any arguments, it will prompt you for proxy ip-

address or hostname, proxyport (usually 8080) and optionally username and password (invisible). If no username/password is required, just leave it blank and hit enter.

This script can also be called upon startup in the bootmenu.

Usage: 'setproxy'

If you want your proxyserver to always be included, I recommend you set up your LAN to be "TRK-aware", explained later in this documentation.

Setproxy sets the proxy settings for these applications:

-wget
-links
-ClamAv
-F-prot
-BitDefender
-general http_proxy environment variable

Also available for reading is the online manpage:

http://trinityhome.org/manpages/man8/setproxy.8.html

2.19 Ntfsundeleteall

This script is a simple wrapper for the utility 'ntfsundelete'

What it does is **find all deleted files on an ntfs volume and tries to recover them**. It does this by looking for traces in the ntfs Master File Table and sees whether the file is still available on the disk (or partially overwritten). It 's a good utility that recovers your files with the original filename but without the relevant path in front. So to avoid double recovered filenames, I 've written the script so that it prepends the original inode (=internal number for the file on the filesystem) before the filename, so you always have a unique filename. It 's up to you to rename the files afterwards, but at least you have the original name in the recovered filename.

Here 's how the script works

Suppose you have the volume /dev/hda1 in ntfs from which you wish to recover files.
Make sure this volume is NOT mounted
Prepare a place to which you can store your recovered files. Recovered files will never be written to the original partition from which you recover them because the risk exists you will overwrite other files that still need recovery. Let 's say you have space on a network volume that you 've mounted on /mnt0 (for this procedure, see the section "Getting around with commands"), with a subdir called "recovered"
Here 's the syntax of ntfsundeleteall:
ntfsundeleteall Device Savedir [minimum percentage] --force

Now here 's how we do it in our example:
'ntfsundeleteall /dev/hda1 /mnt0/recovered'

This is very simple, this will recover any file it finds, even if only 1% of the file is recoverable. You 're probably not interested in that, because these files are most likely to be corrupt and unusable, so you add a parameter on the percentage you wish the file is recoverable. If we only want 100% recoverability, make the command like this:

'ntfsundeleteall /dev/hda1 /mnt0/recovered 100'

You will see the directory /mnt0/recovered filling up with recovered files, hopefully your file is in there.

You might have noticed the parameter "--force". This will only skip the check that your filesystem is ntfs, you can use this if you think your filesystem is so badly damaged that it won 't even be recognized anymore, but I doubt you will be even capable of recovering any files then.

Another recommendation I can give you is Photorec. Photorec is a third party utility that is able to recover files in a unique way: it is independent of the filesystem, it just does raw read of the disk (tries to use as much info as it can from whatever filesystem there was on it) and tries to recognize known document formats to recover whatever it can find. Although the name implies it might only serve to recover pictures, it can in fact recover any kind of important document, including word and excel, but also file archives such as zip and tar.gz.

The only thing you won 't get from this utility is the original filename, so you 'll have to sort out your recovered documents one by one.

2.13 Getswap

Find swapspace in Windows pagefiles and swap partitions and use it as tempspace. This script is called upon by virusscan and updatetrk to find more working memory. In normal circumstances your available ram should suffise.

Getswap will remount /dev/shm and make more shared memory available (in e.g. /tmp/) Sometimes you have too little RAM, so a swapdisk is a reasonable alternative to gain memory.

USAGE

getswap -s <SIZE in % of total memory> -d -f

-s SIZE: this parameter allows you to tweak how much of the ram will be made available for /dev/shm (=/tmp in TRK).
Sometimes you need a huge amount of tempspace. With this parameter you can specify the size in percentage of the total amount of memory (RAM+swap). Default figure is 50%.

-d: disable swap usage and return to original memory situation if possible

-f: force disabling even if too much tempspace is used. Will not work if the tempspace is more than 100% of the amount of RAM.
Will fail at even less.

2.14 Trinisup

Trinisup is a tiny utility that connects to a remote server and sets up a tunnel to your local TRK. It allows the TRK support team to connect to your TRK and share a local console, allowing full access to your computer. Trinisup uses ssh with a private key to connect as a user to the remote trinityhome.org support server. This user is not active until the support team has activated it. So in order to get help, you will first have to contact the TRK guys, who can then activate this user and provide assistance.

USAGE

Usage is very simple. Just run 'trinisup'. By default it maps port 30000 to your local port 22 (which runs an ssh server). If port 30000 is already occupied, you can specify an alternate port as an argument to trinisup.
For example: 'trinisup 30001'
Once the connection is established, you will get a non-interactive message displaying that you are connected.
Hitting <enter> will disconnect this session as long as no one is connected to your computer. If a session is still in progress, this connection will only exit as soon as the session is terminated or you press ctrl+c
You need tcp port 443 to be open to the outside world. All other connections happen inside the tunnel.

A shared local console can be accessed by tapping alt+F6

2.15 Pi - automated backup wrapper script originally for Partition Image

Pi is a third party tool contributed by a few great guys (also responsible for the TRK menu) who wish to remain anonymous. Below is the (elaborate) manpage.

NAME

pi - automated backup wrapper script originally for Partition Image

SYNOPSIS

pi -c -i [CONFIG FILE] -a -k [NUMBER OF COPIES] -d

DESCRIPTION

The pi script was designed for:

make possible a fully automated backup system.
facilitate the use of partimage.

OPTIONS

Type pi then <RETURN> to launch interactive mode.

-c :
search a pi.cfg configuration file and loads it if found. If this file contains the parameter AUTO=1, the backup will run automatically.
-a :

Starts the backup automatically (same as AUTO=1 in pi.cfg)

-k # :

This option (keep) sets the number of copies of the image to be kept. See details in Section *Old backups* below

-d :

debug mode. The backup will not be actually launched and the command that would have launched partimage is only displayed on the screen

Trick :

If you execute pi with -c -d, you will see which partition(s) and destination are selected from the found pi.cfg file.

FILES

pi.cfg :

pi can use a configuration file. This file is not needed in interactive mode. This file has two purposes :

Add menu entries

pi automatically creates the list of your local drives. If you want to backup to a network drive, you must add it in pi.cfg

Automatic mode

All selections can be set in pi.cfg and then pi can run without any user intervention A pi.cfg file corresponding to your backup is created in the destination directory each time a backup is successful.

partition_info.txt :

Each time you make a backup, a file named *partition_info.txt* is created in the destination directory. This file contains information on your partitions (all partitions of all local drives, even USB disks or USB keys). This may be useful in case you have to repartition your disk after a crash.
WARNING : partimage is unable to restore an image on a partition smaller than the original partition. See the documentation of partimage for details. Therefore it is very important to know the size of the partition that an image contains. unpi, the complementary script of pi, will give this information but you can also find it in partition_info.txt.

MOUNT POINT

pi mounts the backup unit you have chosen on /sysbackup

RELATED SCRIPTS

Well, you have make an image not only for the fun, you may want to restore it one day. You can use partimage or **unpi** the little brother of pi which will help you for restore in the same way that pi helps you for backup.

INTERACTIVE MODE

If the pi script is run with the **-c** option, and there is a pi.cfg file in the root directory of any disk in the machine, or in /trk/trk3/pi or in /etc/pi and all options necessary

are set, then pi is launched automatically, using the pi.cfg parameters. If pi.cfg is not found or uncomplete, pi exits immediatly.

* If the pi script is run without the **-c** option then pi is launched interactively and three successive menus ask the user to select:
* The unit or units to save.
* The destination disk that store the backup (local drive or network)
* The destination disk list may be customized using MENU_UNIT[], MENU_LOGIN[] and MENU_PASS[] in pi.cfg file.
* The destination directory

a) The partitions to save
Here you can select a partition in the list of partitions found by the system, or select "information on partitions" for more information on your partitions, or select "select several partitionsâ€□. In this case you must type the names of the partitions to save as they appear in the list, separated by spaces. For example: hda1 sda1

b) The destination drive (the one on which the backup will be saved).
A list presents successively the units defined in pi.cfg and the units that the system finds on local disks. After selecting the unit, the script will attempt to mount the disk. If it fails, an error message is sent and the script stops. Possible causes include, among others:

1. You have chosen an NTFS partition and Windows has been put in hibernate state on the disk. In this case, NTFS partitions are not closed and they cannot be mounted rread/write, you can only read. It is possible to make a backup of such partition, but you cannot use it as the destination unit. To avoid this error, completely shut down Windows before starting TRK Autobackup. Since TRK revision 3.4 : hibernated partition cannot be written, but the state of other not closed partitions is normally solved by ntfs-3g which is able to successfully mount them for writing. However it is always better to properly shut down Windows before making a backup.

2. The system also put in the list extended partitions, which can disturb. An extended partition is a mere container and cannot receive files. For information on how your partitions are made, use the "information about partitions" option of the main menu.

3. You have selected a network drive and the network connection is not good, or your LOGIN and PASSWORD parameters are incorrect.

c) The destination directory
In the destination device previously defined, the script can either use the existing directories that are presented here, or a new directory: in this case select the last option and type a directory name **without spaces in its name.** We are in Linux which does not accept spaces in the names of directories and

files.
Then the script will present you a summary of your choices that you can confirm or reject by **y**(es). or **n**(o). In the last case the script closes. Finally before running the backup a last message information is presented. You can again interrupt the process by pressing any key other than SPACE or ENTER, or type on ENTER to immediately start the backup, or let the script look after himself. After 15 seconds, if you do nothing, it will launch the backup.

AUTOMATIC MODE

To enable automatic operation, you must file a pi.cfg in the root directory of a partition of the machine (all mountable partitions are scanned) or in /trk/trk3/pi (for TRK users) or in /etc/pi and the script must be launched with option **-c**. The first file found will be used.
If the file is found, the parameters are loaded and the backup is started. This backup will be excuted without any user intervention if all necessary parameters are set in the file and if AUTO has been set to 1. but there are two confirmation messages which are displayed for 10 or 15 seconds, and during this time you may stop the process before the backup starts. If AUTO=0, then a confirmation is requested from the user. If you add the command pi **-c** to your trklocscript file in the trk3 directory, then you can have a CD or USB key which will backup the partitions you have defined without any intervention of the operator : just boot on the CD or the USB key, and you are done. This is handy if you want users who don't know anything about computers to be able to save their system disk from time to time.

To setup automatic mode, the recommended steps are :

*
Use pi in normal interactive mode to make your first backup.
*
Once you are satisfied and anything works as you want, locate the the pi.cfg file which has been created in your successful backup directory
*
Copy this file in the root of your first drive (any other drive can be chosen, but it is more easy to use the first one, generally sda1 or c:)
*
You can edit this file in Linux or Windows if you want to change something, Windows end of lines are supported. Complete description of this file format is explained below in the section *CONFIGURATION FILE*.
*
At this point, if you run **pi -c** your backup will start automatically. You will have two occasions offered to you to stop it.

If you add the command pi -c in the trklocscript file in the trk3 directory of your TRK disk, then it is possible to start a backup by just booting on this disk, without any further intervention. Complementary details are found in the FAQ.
$$$

FILES CREATED

Pi creates three files (or file set) in the destination directory :

*
The partimage files with extension .000, .001 .002 etc.
*

a pi.cfg file with the configuration you have just used and which can be used for automatic process.

* a partition-info.txt file, which contains the details of your hard disks configuration. It may be very useful in case you must reformat your disk

BACKUP SCREEN

During the backup, partimage shows a screen that displays the progress of the operation. Despite the message that is displayed at the bottom of the screen, it is unfortunately not possible to interrupt a backup because the * option does not work. This is a bug in partimage. If you want to stop anyway, you can do a ctrl+alt+del. You will just need to think about delete the created files.

OLD BACKUPS

The script handles the number of copies of your partitions you want to keep. The default is 2 (the copy you just made, and one older). This value can be changed with the option -k. 4 will mean : the image just made + 3 older copies (total : 4 copies). Examples :

-k1 : One image : all older images are deleted
-k2 : The last image + 1 older image (default)
-k3 : The last image + 2 older images

-kn : The last image + (n-1) older images

-k0 : Special mode (see below : IN CASE OF LOW SPACE IN THE DESTINATION) The digit does not mean the number of older copies but the total number : -k2 will not keep two older copies but one.

The behaviour is the following :

a) the current backup is performed into temporary files.

b) these temporary backup files are renamed with their final backup filenames, like 20YY-MM-DD-X-hda1.000

c) a sub-directory is created with the following name: 20YY-MM-DD-saved_partition, for example: 2010-06-22-hda2 and the current backup files of this partition are transferred under this sub-directory.

d) depending of the number specified with the k option, old backups are deleted: for example, if user has specified pi -c -k 4 and there are already 3 existing 20YY-MM-DD-saved_partition sub-directories, the oldest one will be automatically removed once the current backup will have successfully completed.

CONFIGURATION FILE (pi.cfg)

CAUTION: The name of the configuration file is pi.cfg or PI.CFG or Pi.cfg. ALL OTHER SPELLINGS WILL BE IGNORED.

Spaces are not allowed in file names or directory (replace them by _ or -)

When you have completed a backup, the destination directory contains a pi.cfg file corresponding to the options you had choosed. Here is a possible example :

```
# No space before and after the = sign
# The names are case-sensitive
# No backslashs: \, only slashs/


# MENU_UNIT[x]=//192.168.1.4/sysbackup specify a network disc to be displayed
in the interactive choice of BACKUPDRIVE
# MENU_LOGIN[x]=//192.168.1.4/sysbackup specify login of the network disc[x]
displayed in the interactive choice of BACKUPDRIVE
# MENU_PASS[x]=//192.168.1.4/sysbackup specify password of a network disc[x]
displayed in the interactive choice of BACKUPDRIVE
# x index start at number 1. Several network discs may be specified using different
indexes.
# Destination automatic backup unit. It may be a network drive, for example:
# BACKUPDRIVE =//192.168.1.4/sysbackup
# In this case the parameters LOGIN and PASSWORD are required
# Or a local unit such as BACKUPDRIVE=/sdb1
BACKUPDRIVE=//192.168.1.4/sysbackup
LOGIN=mylogin
PASSWORD=mypass

# Backup directory in automatic mode
DEST=Lawrence

# Beginning of backup files' names;
# pi will generate 20YY-MM-DD-X- prefix in the filename of the unit to backup
# (X=A for automatic and I for Interactive) and partimage will add the extension
000, 001 and so on.
# Example : in Automatic mode, the resulting file should be :
# â€œ20YY-MM-DD-A-hda1.000â€□
# Example : in Intercative mode, the resulting file should be :
# â€œ20YY-MM-DD-I-hda1.000â€□

# Partitions to backup in automatic mode
# If there are multiple partitions, separate them by spaces,
# and put it between quotation marks
# Example PARTITION="sda1 sda5"
PARTITION="sda1"

# If AUTO=1 and a pi.cfg configuration file was found,
# the backup is done without any user intervention
AUTO=1
KEEP=2
```

Remarks:

Always use the forward slash and not the backslash: we are in Linux and not in Windows. Long before Microsoft existed, the backslash was an escape character on Unix. Only Microsoft's commitment to be "like no other" has led to ignore this meaning and use the backslash in directories' path. Linux meets Unix standards. There are never spaces before or after the equal sign under penalty of non-functioning of the script. The parameters' names are always written in uppercase.

BACKUPDRIVE:

You may indicate a network drive, and in this case it will be also mandatory to specify LOGIN and PASSWORD. The format will be: //xxx.xxx.xxx.xxx/shared_directory_name. This can also be a local drive and in this case the format will be, for instance: /dev/sda1. It is prudent to use the complete device names returned by info on partitions. /sda1 should work too, but /dev/sda1 is safer.

DEST:

This is the destination directory to be used (and possibly created) in the unit selected by BACKUPDRIVE. It is possible to select a subdirectory, but in this case it is mandatory that the directory already exists. Example:

<FILE> - <partition> .000 DEST=backup/system/unite_c

If the saved partitions are hda1 and hda5, the following files are created (X=A or I):

20YY-MM-DD-X-hda1.000 20YY-MM-DD-X-hda5.000

PARTITION:

This parameter contains the names of the partitions to backup. If there is one partition, you can simply specify it as follows:
PARTITION=hda1
If there are multiple partitions, they must be separated by spaces and put between quotation marks:
PARTITION="hda1 hda5"
It is allowed to have spaces between quotation marks but it is prohibited outside. PARTITION= "hda1 hda5" will not work because you should never have a space before and after the = sign
Note the different syntax from BACKUPDRIVE. There's no slash or /dev/.

AUTO:

AUTO=1 enables automatic backup and a delay is placed on the confirmation messages. It is possible to interrupt the process by pressing any key other than SPACE or ENTER during the period. After the delay and without user intervention, backup is started.

KEEP:

This parameter has the same meaning as the parameter -k (see the section "Old backups" above). The default value is 2.

IN CASE OF LOW SPACE ON THE DESTINATION

using -k0 option

In the normal process described above, there is a moment where you have two images of your partition : one that is currently on temporary files and one which is in the normal directory. If you do not have enough space on your disk for this, you can use the **-k** option with a value of 1 or 0

-k1

If you have enough disc space for two copies of the image but want to keep only the last, this parameter will cause the following behaviour:

a) creation of the new image in temporary files
b) if the image was successful, the old image is erased
c) the new image files are renamed.
If the image had failed, temporary files remain on the disk. They will be overwritten by the new image when you will try again to make one. You can also delete them yourself.

-k0

If you have just the place for an image, this parameter will give the following behavior:

a) erase the existing image
b) creation of the new image
Warning: this mode is not secure because if the creation of the new image fails, the old image has already been erased.

FAQ

Q: How can I make a fully automatic image system for my hard disk ?

A: The first step is to setup a pi.cfg file which will do what you want : save your system partition or all your partitions to a given destination.
Once this is done and you can run pi -c and get the result you want, the hardest part is done. See above for all informations on this step.
Now you have to run this automatically when TRK starts. This is easy :

* Add a file with the name "trklocscript" in the trk3 directory of your TRK USB key or your TRK CD.

*
In this file add this line :
pi -c
You are done.

Q: My computer does not boot on an USB device and I must boot on CD. I cannot copy a file to a CD.

A: When you burn your TRK CD, be careful to let the session open. This can be configured if you are using CdBurnerXP on windows. Some versions of Nero automatically close the session and don't let you add anything.
- Open your CD in a CD burner program
- Choose the option to continue the CD
- add trklocscript as described above in the trk3 directory
- burn your cd
- Run it. You are done.

Q: And what will happen with this CD/USB key ?

A: Just boot a computer which has a valid pi.cfg file in the root directory of one of its disks, and the backup defined in that pi.cfg file will be performed. If you had set AUTO=1, just sit back and relax the job will be done automatically. If you are suspicious and want to know what happens before it is performed, then set AUTO=0 and you will have a confirmation screen.

Q: I am lazy and I want a simpler solution

A: OK. You must first create a TRK USB key with trk2usb.
Add trklocsript to the trk3 directory of this key
Ask a friend who has a computer which starts on USB key to start his computer on your TRK USB key and to run the command :
trk2iso
N.B. He must first change to a directory with at least 150 Mo of free space
Then ask him to give you the iso file created.
Burn it and you are done.

Q: This is not so simple

A: At first, yes. But once you have your USB key it is very easy to modify it, and then ask your friend to create the new iso. If he is patient enough, it is a good solution.

Q: I don't want to run always pi on startup. Is it possible to have this feature enabled on a computer and not on another ?

A: Yes, it is easy : with the pi -c command in the trklocscript file, if a computer has a pi.cfg file, pi will run when TRK starts. If a computer does not have this file, then pi will appear for 3 seconds, then disappear and the TRK menu will be shown.

LICENSE

Public domain

AUTHORS

Frank Michel (for the menus), Gaston and Averell

2.20 Clonexp (obsoleted by mclone)

Clonexp clones an ntfs partition to another computer over the network. Both PCs must be running the same version of TRK.. The one who is receiving the data must run with sshd enabled (triggerable from the simple TRK menu)

Basically what it does is run ntfsclone in save mode to stdout at one computer and restore mode from stdin on the other, piping over ssh between eachother.

When running clonexp, make sure your source partition is smaller or equal to what can fit on the destination. Ntfsclone (and subsequently clonexp) cannot dynamically resize a partition, so it 's a good thing to make the destination partition equal or bigger in size than the source. Do this easily with parted, supplied on TRK. If your maximum possible destination size is smaller than your source, try making your source smaller with parted. At least, if the data on it is less than the destination and the volume is not too fragmented.

Also make sure neither of the local nor remote filesystems are mounted, or the script will exit.

Clonexp can run in two modes: without any arguments, it runs in interactive mode and prompts you for source device, destination host and destination device. It will each time test whether these things exist.

You can also run it in a single commandline, in this way (drop the <>):
clonexp <sourcedevice> <destinationhost>:<destinationdevice>
e.g. 'clonexp /dev/hda1 192.168.0.7:/dev/hda1'

It will prompt you once for the password you have entered on the remote TRK machine and then start copying the data. There is also the option '--bkupbr', which will first backup your partition bootrecord and later save that as a file on your newly created partition. Although it may never really serve, I give this opportunity should you ever be unable to boot from the partition and you want to play around a bit with your old bootrecord. This might as well completely obliterate your partition, so use with caution (backups 'n stuff ready...)
I was once able to make a cloned unbootable partition bootable this way by afterwards running from the Windows XP recovery console and do a chkdsk.
It 's recommended you use clonexp only between computers which have the same disk layout, or even better, are exactly the same hardware.
Clonexp is obsoleted by mclone, which is more efficient, more bugfree and multiclient capable

3. Procedures

Let 's talk a little on what actions you should take in time of need, how to best use TRK to perform troubleshooting, or how you even might work with tools other than TRK.

3.1 Rescueing files of dying harddiscs (mounting network => cp, ddrescue)

Cool it!

When you have a harddrive that is starting to die on you (bad clusters, physical damage), it is imperative that you "upset" the disk as little as possible. In most cases the cause of your dieing disk is heat: hard drives can get very hot, especially when heavily used and poorly cooled. A good consideration is to make sure that your disks are placed freely floating so heat can get away easily. Also make sure that all of your fans in your computer blow in the same direction. A standard here is that fans in the front "inhale" air, while fans in the back of your computer "exhale". If you are considering placing a fan inside your computer directly on your disk, it is best practice to let it blow away from the disk. This way you create a vacuum and no air anywhere, not even in the smallest holes, can sustain in a vacuum.

Another good practice is to attach your disks firmly to the computer, so the iron from the chassis can guide away heat from the disk. Note: Do this only by attaching the sides of the harddisks to the chassis.

Now, in case it is already too late and your disk is dieing, it may be that it works fine initially and then starts to make funny noises afterwards: probably the bearings have worn out or the platters are starting to distort and the heads are grating on the surface of the disk.

The most important thing here is to hurry up and get those files off that disk.

One suggestion I might give here might sound weird, but it worked for me already in many cases:
put the disk in the fridge for a while and then boot it up! Really, I'm not kidding. However, don't freeze it, that will only make it worse. Put it in a normal refrigerator and let it cool down in there. Also put a towel around it to avoid condensation.
Another possibility here is to connect your disk externally through a disk-to-USB cable that you can buy for 20$ or less at your local computer shop. (or connect the disk to an IDE cable that hangs out of your PC like guts)
Now, should you have that, go and look for that wine cooler pack you got for your birthday, you know, those plastic things you put around a bottle to keep it cool (or go and buy it at your local grocery store). A gelpack to ease the muscles can also do the trick, but don't get it out of the freezer where it's minus 18° Celsius, that is far too cold and might damage your disk even more. Always wrap a towel around your disk first.

Now, once this is in place, you will have a better time window to start recovering files.

What you can try, from good to bad to worst case scenario.

In the best case you will be still able to mount the filesystem on the disk. Use mountallfs to search for all local filesystems.

Make sure you have a location to which you can copy your recovered files. This can

be another local disk or a remote network filesystem such as a Windows share. How you should reach this safe haven is explained earlier in this document (Getting around with commands).

So let's assume a few things:

-mountallfs manages to mount your filesystems on your dieing disk
-you mount a remote network share:

'mount -o username=administrator,password=mypassword //10.0.0.5/c$ /mnt0'
-'cd /hda1' where hda1 is your dieing disk
-do an 'ls' to see that you can still see files there.
-go to the directory from which you would like to recover documents. Don't start to try and copy all files, your time here is limited and it is very likely that your disk is damaged in places where it has read and written many times. In the case of a Windows system, this is the WINDOWS system directory. Stay out of it, there are normally no documents there.

More likely you will do something like this:

'cd /hda1/Documents\ and\ Settings/John\ Doe/My\ Documents'

-If you don't know how many documents you are going to save, do an 'ls -l'. If you see there are many subfolders, do an 'ls -lR' or 'du -h'. 'ls -lR' is the equivalent of a dos 'dir /s', but it is better to do 'du -h'. This will tell you how much data there is in this folder, whether there are big files, small files, many files, etc...
-Once you know this, estimate what you have to rescue. Listen to what the disk has to say, if it starts making funny noises already, try to save important data first.
-Let's say all is well, start a copy (first make a remote dir to save your files to):
'mkdir /mnt0/recovered'
'cp -rvf * /mnt0/recovered'
This command starts a copy of all files (*) to /mnt0/recovered, all subdirs (r), shows the progress file by file (v) and forces overwriting files on the remote location (f).

If this works out: good for you.

Otherwise...

If your disk starts to make funny noises and you see error messages such as

"Disk Seek Complete Error", this means we are running into bad clusters and your copy operation will probably slow down dramatically.
In this case, try to start copying files or folders more separately. Skip the files first on which it runs into errors. There are other means to (maybe) recover them later.
If that works out, but you still need the other files, let's try something else.

I recommend you let the disk rest for a while by turning it off for about 20 minutes and put it in the fridge wrapped in a towel.
Once you've started it up again, you might have another go at copying the bad files again, but cancel it as soon as you run into the same problems again.

In this case go over to using ddrescue.

ddrescue takes an input file and copies it to an output file, just as a normal dd would do. In this case, however, there are a few differences: it tries to copy raw data from a file, but it will not try for too long when it runs into bad, completely unreadable clusters; it will fill up the unreadable spaces with zeroes.
Example: 'ddrescue /hda1/Documents\ and\ Settings/John\ Doe/My\ Documents\outlook.pst /mnt0/recovered/outlook.pst'

You can see here that the rescue is file by file. Here is a quick oneliner to rescue a complete directory.
'cd /hda1/Documents\ and\ Settings/John\ Doe/My\ Documents\; for i in *; do ddrescue "$i" "/mnt0/recovered/$i; done"'

Another way of trying to rescue your data is to go and copy complete disks or partitions. Just so you can be able to use Windows' chkdsk later on the rescued filesystem, ddrescue the contents to another partition on another disk. It is also possible to rescue the filesystem to a file and try to mount the file as a loopback device from Linux/TRK, but you won't be able to let Windows access it.
So here goes an example (make sure nothing is mounted, neither source nor destination): 'ddrescue /dev/hda1 /dev/hdb1'
In this case you will raw copy the complete contents of your partition /dev/hda1 to /dev/hdb1. Make sure /dev/hdb1 is equal or bigger than /dev/hda1
Boot into Windows and let it run a chkdsk on it (or use the Windows install CD in recovery mode).
If your filesystem is native Linux, such as an ext2 filesystem, you can rescue it to a normal file somewhere and run e2fsck on it like that.
Example: 'ddrescue /dev/hda1 /mnt0/rootfilesystem.img logfile-hda1'

If ddrescue was unable to rescue all of your hda1 because it had to skip certain sectors, try going into raw mode, which will bypass your kernel disk i/o layer and read sector per sector. In raw mode it is much slower, but ddrescue keeps its already completed work in a logfile and only does the sectors that were skipped.
Use it then like this:
'modprobe raw'
'raw /dev/raw/raw1 /dev/hda1'
'ddrescue -r3 /dev/raw/raw1 /mnt0/recovered-hda1.img logfile-hda1'
Once you have this, restore the image to another disk that boots into a normal PC into Windows and try running chkdsk against it.

If it was a Linux filesystem (ext2 for example), there is no need to restore it:

'e2fsck -v -f /mnt0/ rootfilesystem.img'
'mount -o loop /mnt0/rootfilesystem.img /mnt1'
<= check your files...

NOTE: it is not recommended to try and perform a filesystem check that starts having physical I/O errors. It will probably make things even more unrecoverable.

3.2 Recovering deleted files or files from formatted drives (ntfsundeleteall, photorec)

In TRK, there are several ways you can use to undelete files. What you should use depends on the state of your harddisk/filesystem and the success of recovery can vary between various methods.

Ntfsundeleteall

The first method already explained earlier is ntfsundeleteall (section 2.8), so I'm not going to re-explain this. What I should mention about this method is that it has a high recoverability ratio on recently deleted files (deleted in a normal way, not

because of a crash or filesystem corruption). It does however a poor job on drives that have already had more write I/O over them. It is completely unusable on formatted drives.

Photorec

In all other cases, you can try the photorec utility, which tries to recover any important kind of document from any filesystem. Be sure to run it by first changing to a place where you can write your stuff, because it saves its recovered output to a subfolder relative from where it was launched. Example:
'mount //10.0.0.1/share /mnt0'
'cd /mnt0'
'photorec'

If you suspect that your filesystem is corrupt, I would first try to run a chkdsk on it and see what it recovers. Then afterwards, see what shows up and, depending on what has become visible, use the ntfs capabilities of TRK or use photorec... But do not run a chkdsk if your disk is physically dieing!!! Try to do as little I/O possible on that disk. Running a chkdsk will probably render it even more unusable in that case.

Maybe ddrescue first

If photorec is unable to recover because of too many corrupt sectors, you can try ddrescue.
This util tries to recover files and filesystems that are on locations with bad sectors. If it is really unable to read the bad sectors, it will pad them with zeroes. Your files will possibly be corrupted, but at least you will have saved whatever there is to save. If you ddrescue a complete filesystem (e.g. 'ddrescue if=/dev/hda1 of=/mnt0/rescued-image.img), you can afterwards mount that image as a loopback filesystem ('mount -o loop /mnt0/rescued-image.img /mnt1') and rescue/undelete files like that.

3.3 Recovering lost partitions (testdisk, gpart, fdisk)

It can happen that you did something stupid or something inexplicable happened and your partition table has gone or is corrupted.

To recover your partition tables, there are a few methods.

Fdisk

I'm not referring to the Microsoft dos version of fdisk, but to the fdisk on Linux. Actually what you do with this is to manually redesign your partition table. Of course, the risk here is that you have absolutely no idea how your partition table looked before or even if you had more than one partition. So this is a good strategy in the case of a disk having only a single partition.
Let us say, in this case, that the partition contains an NTFS filesystem on /dev/hda1
Make sure nothing is mounted (run umountallfs)
Run fdisk:
-'fdisk /dev/hda'
-In fdisk: press 'p' to print your actual partition table
If your actual partition table is a bad one, erase by removing each partition with 'd' and answer with the right partition number (in case of logical drives, first remove

everything equal and above 5).
-'n' for new partition
Choose primary partition, press enter for start and end sector. Fdisk automatically assigns it as a Linux native partition.
-'t' to toggle the partition type. Ntfs is '7'
-'a' to make it active (choose '1'), since it's probably the bootpartition
-'w' to write your changes

If this was your previous partition layout, you should now already be able to mount the ntfs partition. If not, you probably won't be able to boot from it either. But it should be, since this is the way to do it when the problem was simply a lost partition table.

Testdisk

This is the more advanced method of recovering partitions. Run 'testdisk' at the command line, the rest speaks for itself (Intel partition table => Analyze => etc.).
Read the testdisk documentation atwww.cgsecurity.org
Testdisk is also very nice to repair bootsectors. Just play with it a little on a test machine to see what it all does. I have played around with it myself, and although there is no guarantee that you will be able to repair the bootsector every time, it still did a pretty good job where others failed.
Another thing I can recommend to you about testdisk: if it does not find all of your partitions in the first minute, there is probably not much left but some free disk space without a valid filesystem. It is safe to cancel already if you think it found all you need.

Cfdisk

A menu based fdisk replacement, easier to use than the standard fdisk

Gpart

Guess partitions: a utility that might recover your partitions too. However, it has not been maintained for a few years and I have not yet tested it myself.

3.4 Bootsector repair

About bootrecords

Several utilities can be applied to repair a bootsector and there is a lot that can be said about it.
I will only describe a few procedures for Windows and Linux bootsectors.

In a PC style (x86) environment there are 2 ways a machine can boot from a harddisk: Either from the MBR (Master Boot Record) OR from the bootsector on the active partition. Historically, Windows has always booted from the active partition bootsector, but it also writes some stuff to the MBR of the disk. I don't know all the internals of the different bootprocesses, so please correct me if this documentation has it wrong.
Continuing: Linux can boot in the two different ways: Either it writes its code to the MBR (e.g. /dev/hda), OR it writes it to the active partition (e.g. /dev/hda1).

Check partitioning

Sometimes the failure to boot can simply be a partitioning problem. In that case, refer to the previous section. If you rebuild your partitioning as it was before, it most likely will boot again.

Never forget to set the correct active partition!

Windows boot

Let's take a quick look at procedures on how to repair Windows bootrecords.
As I already said, Windows booting depends on both the MBR and the partition bootrecord code.

-ms-sys
This utility can write new MBR code to your disk. Type 'ms-sys -h' to obtain some help on the different parameters, but in most cases you run it first to analyse your disk. 'ms-sys /dev/hda', will tell you about your MBR.
'ms-sys -m /dev/hda' will write an XP style MBR to your disk.
Remember that this utility deals only with the MBR.

For the windows bootsectors you must refer to other utils.

-testdisk
I've tried testdisk to write new MBR code and it worked like a charm for me. What is also nice about it is that, when you go into advanced mode, you can also analyse the bootsector of the partitions and repair them if needed.

-the Windows bootcd way
It might happen that none of these methods get you back into booting Windows, then there is one last method that you can try, which actually has nothing to do with TRK. Boot from the Windows XP cd and go to the recovery console. Make sure you know the password of the Windows system on the local disk. Otherwise, reset it first with 'winpass' on TRK.
-Now, once booted into the recovery console, it is a good thing to run checkdisk first: 'chkdsk c: /p'
-Next run 'fixboot'
-And finally run 'fixmbr'
-'exit' to reboot

-if really nothing worked: FAT32
Finally, if nothing did the trick, it means something is wrong in some way, I don't know, your disk geometry was laid out, filesystem were created, etc, etc. Sometimes it is even a mystery to me why a Windows will not boot when everything else says that it should. However, it often happens after I have cloned a PC with clonexp or Partition Image. Sometimes it works, sometimes it does not.

In that case there is one last resort: **recreate your Windows on a fat32 partition**.
For documentation on that, see the section:**3.5 Manually cloning a Windows installation.**

-ntfsreloc

Very handy tool if your bootsector geometry does not correspond to what the bios claims it to be (the bios is actually wrong). Yet Windows bootsector finds it important it corresponds to the bios specifications.
This utility is invoked by mclone after cloning to make sure the bootsectors are exactly the same as they were (Linux adjusts them to LBA).

Here 's the help output of ntfsreloc 0.7:
adjust filesystem geometry for a NTFS partition
Usage: ntfsreloc [-h # -t #] [-s start [-b]] [-w] [-f] [-p] device
where device points to an NTFS partition

Options:
-w: Write new start sector to the partition.
-h # -t #: Specify number of heads and number of sectors per track
If omitted, determined via ioctl.
-s start: New start sector to write.
If omitted, determined via ioctl.
-b: Proceed even if the specified device is not a
partition (e.g. a regular file)
-f: Force the operation to occur even if device does not look
like a valid NTFS partition or values are equal.
-p: Print debug information (values read, values requested etc.)

This utility displays the current starting sector as defined by the
the filesystem. No change will actually be made without the -w
option.

Exit status is 2 if an error occured, 1 if a change was made or is needed
or 0 if the filesystem already has the correct values.

3.5 Manually cloning a Windows installation

This method describes how you can migrate a Windows installation to another computer and be sure it will boot as long as the Windows installation itself is not corrupt. Additionally, I will also describe how you could make your system bootable again with only a single computer/disk available.

The way we approach this type of "cloning" is very simple: we create a fat32 partition, copy all data from the NTFS partition to it, let the Windows bootcd repair the MBR and bootsectors and Windows boots. Afterwards you can convert the filesystem back to ntfs from Windows.

The procedure

Let's assume a few things first: you have two harddisks attached to 1 PC one of which might be dieing or chkdsk does not repair it and you can not use any cloning tool because it finds errors and exits (happens with Ghost and PQ Drive Image). In the case of a laptop computer, you do not have the possibility to attach a second disk unless via an external USB enclosure, which is not recognized under dos based utils. In TRK you can either mount a remote disk with two PCs running TRK of which one is running in fileserver or secure shell server mode (see other sections on how to do that). The possibilities are myriad.
But our example here works with two local disks, the source disk being an NTFS partition, the destination disk empty or erasable.
Disks are respectively /dev/hda and /dev/hdb, the NTFS partition is /dev/hda1
-Make a large enough vfat filesystem on /dev/hdb. Use qtparted to do it graphically or use...

'fdisk /dev/hdb',
-'n' to create a new primary partition
-'1'
-'t' to toggle, code 'c'
-'a' set active '1'
-'p' to print all that is about to be done to the screen
-'w' to write and quit the partition table
-'mkdosfs -F 32 /dev/hdb1'
-mountallfs
-cp -a /hda1/* /hdb1
-Shutdown the PC, change the harddisks so your new disk becomes the primary (/dev/hda)
-Now, reboot with the Windows CD so we can make the disk bootable (I have found it to be the best way).
Follow the procedure described in 3.4 Bootsector repair .i.e. fixboot and fixmbr.

Your Windows should now start. Once booted into windows, go to a command line and run 'convert c: /fs:ntfs'
Reboot and let your disk convert to ntfs

I have noticed that the Linux 'cp' makes the windows systemdir (in most cases named WINDOWS) into lowercase. It is good practice to make it uppercase afterwards. Although Windows is supposed to be case independent (but still case aware), I've noticed that some services would not start until I uppercased it
-'cd /hdb1'
-'mv windows/ WINDOWS/'

One note here: **Windows XP will not boot from FAT 32 partitions larger than 32Gb**. So do not create them bigger than this. If you have more data to copy, I suggest you move your big files somewhere into a directory that you then omit copying.

3.6 Hardware testing

Although TRK does not claim to be the best hardware test method there can be, there are still a few utilities and procedures you can use to test the health of your computer.

A lot of information about hardware and resulting errors can already be obtained by reading out the boot procedure and system information, described in section 1.4 Reading information about your PC (dmesg, /proc/partitions). Be sure to read that first.

Here we are going to put ourselves a little more into stress testing.

Harddisk testing: Bonnie

Bonnie is a utility that stress tests the I/O of your harddisks. If you want to measure performance or you just want your disk to run very hot, merely mountallfs your filesystems (for ntfs volumes run 'mountallfs -g') cd to a dir on the filesystem (e.g. 'cd /hda1') and execute 'bonnie'
Here is what it could show you:

```
[root@trk]:(/hda1)# bonnie
File './Bonnie.3078', size: 104857600
Writing with putc()...done
```

```
Rewriting...done
Writing intelligently...done
Reading with getc()...done
Reading intelligently... done
Seeker 1...Seeker 2...Seeker 3...start
'em...done...done...done...
            -------Sequential Output-------- ---Sequential
Input-- --Random--
            -Per Char- --Block--- -Rewrite-- -Per Char- --
Block--- --Seeks---
Machine  MB K/sec %CPU K/sec %CPU K/sec %CPU K/sec %CPU K/sec
%CPU/sec %CPU
        1004500 16.368208.359559.59151 11.699855.1 132.01.5
[root@trk]:(/hda1)#
```

Bonnie tests a number of write operations on your disk.
If you want it to run infinitely, just type 'run-bonnie'

CPU testing: burnP6

This is a tool for overclockers and anyone who likes to know that their computer does not crash under heavy CPU load. It uses 100% cpu and makes bad cpus go mad (probably).

Network testing

Actually, a real network testing utility is not included, but you can put a lot of stress on your network card just by transferring large files.

Memory testing

The latest version of memtest+ is included. Just select it from the start menu of TRK (scroll down all the way to the bottom). It tells you information about your memory speed performance and tests to find any possible errors.

3.7 Virus scanning

See section 2.1 Virusscan for all the information you need on virusscanning your PC with TRK.

3.8 Manual PC cleaning

Use winclean, described in section 2.4 Winclean

4. Boot time options and triggers

This section gives you more info on the internals of TRK, how and why to use the differen boot menu options and the triggers TRK has builtin for customisation.
For advanced users who want to automate more or need alternate ways of booting.

4.1 Boot menu options

The boot options of the latest TRK 3.4 contain a lot of different things, but most of the time the default startup will do.

Nevertheless, here are the options explained

-Default startup:
This will run TRK 3.4 with all the default options enabled. This means it will run partly from CD/usb/network, keeping your CD/usb stick unavailable for other things in the meantime, but will boot faster. Keyboard will be default qwerty and fontsize about 3/4 of a normal font in a screensize of 800x600 in VESA compatible framebuffermode. Some videocards are not so VESA compliant and can therefore not benefit from the graphical possibilities of TRK (like the Intel 815 chipset). They will have to run in textmode, preventing them from using qtparted and links (and my nice background logo). When booting from network, you cannot unplug the network cable in this mode.

1) TRK 3.4 in failsafe mode (No menu, VGA, noacpi, noapic)
In this mode, TRK will run with a few failsafe options by not employing the APIC chip nor the ACPI. Also framebuffer mode will be disabled, resulting in lower graphics and lack of background graphics that "soothen your jengled nerves". Use only in case of normal startup failure.

2) TRK 3.4 running from RAM (best >= 512mb, 256mb min)
This option will run Trinity Rescue Kit completely from RAM, which means it copies all of its files into memory so your CD or usb stick becomes available for disconnection. In the case of PXE booted TRKs, you could disconnect the network at that moment. This requires at least 256Mb RAM, but recommended is 512Mb and

more. Especially updated TRK versions require about 250Mb more because of the included antivirusccanners..

3) TRK 3.4 - Run 'mclone' in client mode (!)
On startup, TRK will mclone in client (i.e. RECEIVER) mode. **Use with caution!** This allows you complete local disk to be overwritten by a standard mclone in SENDER mode. This option is for people wanting to perform massive PC replication

4) TRK 3.4 in simple VGA mode (debugging of kernel output)
If you know your machine won 't boot in framebuffer mode. As I said before: no links and no qtparted. This option is also useful in case your TRK doesn 't initiate any output on the screen and you suspect it to hang on kernel initialisation. Many times it means there 's a problem with the ACPI interface of the machine. I 've noticed this quite regularly on HP/Compaq machines.

5) TRK 3.4 with Belgian keyboard (use menu for other)
Since me, Harakiri, is a Belgian, I include my own keyboard as a menu option. Other keyboards can be selected from the startup menu. Internationalisation is set to UTF-8, which should now really be fixed since TRK 3.4

6) TRK 3.4 - Virusscan all drives (Clamav, non interactive)
Runs virusscan on all your drives. Stupid option actually, but this is about the only thing you can do that is set-and-forget without the need for any interaction. This runs virusscan without any arguments, so with ClamAV. You will get no login as long as this is running. Nice option for sysadmins who want to set and forget it. But you can equllay wait for startup and select a few other virusscan arguments from the TRK menu.

7) TRK 3.4 - Try more pcmcia and usb nics (when not detected)
I found that kudzu is not really good with PCMCIA cards and USB stuff and I found out that the easiest way fix this was to just have support for pcmcia bridges compiled in my kernel, start cardmgr and try every module I find until success. Same with USB nics. This will only try this for PCMCIA and USB ethernet cards/adapters.

8) TRK 3.4 - Try more SCSI drivers (when disks not detected)

Same argument as with the latter: if your local disks are not found, try this option to try and load every other more exotic driver module. Be warned: this might lock up your computer.

9) TRK 3.4 with a secure shell server enabled
Run a ssh server, a very nice feature, giving you the ability to remotely work on a machine that your nexter has started up for you.
For security reasons, you will first get prompted to create/change the password, after which in green will be displayed the IP-addresses your TRK will listen to. However, if you're running an updated TRK on which you have changed the root password (which by default is 'trk'), you will not be prompted anymore for a password change and your TRK will be immediately accessible over ssh. Your user will just have to inform you the ip-address.

10) TRK 3.4 - Execute local scripts on harddrive of PC
Run local scripts from ./trk/trk3local.conf found on any local filesystem. In this way you can make computer specific scripts and put them on the harddisk. More info, see section "Triggers"

11) TRK 3.4 as bootserver to boot other TRK clients
This option is magic: it makes your local copy of TRK run as a server to have itself booted over the network. Read more on trkbootnet ins the section Network booting

12) TRK 3.4 - Fileshare all drives as guest, no security
This is the same as the previous option, but this will share all your drives without any security boundary or username. Be careful with this, use only in a trusted environment.

13) TRK 3.4 - Single user mode (no menu)
Starts TRK in very basic mode, sometimes useful when your PC hangs or crashes on hardware detection or some other error.

14) TRK 3.4 - Acpi=off, noapic PCI=^bios (Alternate boot 1)
Some problematic machines with lousy ACPI controls might boot with this option. Happens regularly on some older Compaq machines. This will disable acpi and apic and will use the bios' PCI routing table. Try this when TRK hangs on early startup.

15) TRK 3.4 - Acpi=off, noapic PCI=any (Alternate boot 2)
This is almost the same option as the previous one, but I let the kernel decide what PCI mode should be used. I needed this mode on the brand new HP DC7700 machines with Intel Duo Core because it hung on kernel initialisation in the PCI part.

16) TRK 3.4 - PCI=conf2 (Alternate boot 3)
Another option you can try on PC's with strange PCI bridgings.

17) TRK 3.4 - Verbose startup for debugging (no menu)
I use this to see what scripts are doing or sometimes your PC might hang on something but you don 't know what: try this, but don 't get frightened by all the output.

18) TRK 3.4 - SSH, boot- and guest fileserver, run from RAM
This is a combination of a few options This is a quite common combination you would need when cloning PCs from 1 TRK to another and you only have one TRK cd handy.

19) TRK 3.4 - Run from RAM, run mclone as client
Another combination that explains itself

20) TRK 3.4 with proxyserver support enabled
Start TRK with default options and prompts you for a proxyserver. In fact, it just calls for "setproxy", which prompts you for the proxy server address (hostname or ip), portnumber and optionally username and password (leave blank for none)

21) TRK 3.4 - All devices set to read-only
This is an option for forensic researchers who don't want to touch anything on the local computer but don't have a hardware write blocking device. This sets all of your disks to read-only on a kernel level.

22) Memory tester memtest+ v1.65
Almost every distribution comes with this on startup, but this way it 's included in this one too. Nice to test your memory banks with, but real life tests are still much more certain. Still, it might do the trick for you and find you a bad module.

4.2 Triggers

TRK contains a number of so called "triggers", things it checks for and reacts upon when existing.
Very handy for regular TRK users who want to customise it a little.

4.2.1 The TRK options server: make your lan TRK aware

TRK has quite a few ways you can customise it or have custom scripts be run. These scripts can run automatically because I 've built in numerous placeholders for them which I call "triggers"

If you are a regular user of Trinity Rescue Kit, you might find it useful to have some default parameters set according to your environment i.e. lan
Ever since TRK 3.1, there 's a feature that makes your lan "TRK-aware" since a system administrator can configure his lan in a way any TRK3 can take advantage of it in terms of proxy configurations and user optional script executing, a much wanted feature from the past, but hard to implement since you 're working with read-only media.
Finally, the solution is kind of revolutionary: by activating an otherwise unused parameter in you DHCP server, you can have a TRK point to a server where it can find its config and scripts.
This option is from an obsolete RFC from 1983 called 'Resource location server', which was a UDP protocol that could tell you where to find things like routes and smtp servers. I 'm using it now to provide TRK with an IP-address where it can find a simple webserver on which resides /trk/trk3options.conf
If you 're running ISC dhcpd, add this line to /etc/dhcpd.conf:
option resource-location-servers your-trk-webserver-ip-address;
On a Windows server, go to Administrative tools=>DHCP => go to your scope => scope options => option 11 Resource Location Servers.
I wouldn 't worry too much about enabling this option to break something in your network. I 've been running this for some time now at my office where there are about 100 Windows PCs on dhcp. Nobody hasn 't complained so far.

Anyway, when a TRK 3.2 boots and finds a '**resource location server**' from DHCP, it will indeed look for resources from this configuration file.
Below is an example of such config file, it more or less speaks for itself.
The file is /var/www/html/trk/trk3options.conf on my local webserver

```
TRKSECTION WGET BEGIN
http_proxy=http://10.34.5.201:8080
proxy_user=johndoe
proxy_passwd=doesinc
timeout = 10
tries = 2
TRKSECTION WGET END

TRKSECTION LINKS BEGIN
http_proxy 10.34.5.201:8080
TRKSECTION LINKS END

TRKSECTION CHECKUPDATESARGS BEGIN
-proxy-server=http://10.34.5.201:8080
-proxy-username=johndoe
-proxy-password=doesinc
TRKSECTION CHECKUPDATESARGS END

TRKSECTION TRKSCRIPT BEGIN
#!/bin/bash
#
if [ -f /etc/proxy.conf ]; then
```

```
echo "Setting proxy params in freshclam.conf"
if [ ! -f /etc/freshclam.conf.bak ]; then
cp -f /etc/freshclam.conf /etc/freshclam.conf.bak
fi

if ! [ "rjohndoe" = "r" ]; then
sed "s/# Proxy settings/\nHTTPProxyServer
10.34.5.201\nHTTPProxyPort 8080\nHTTPProxyUsername
johndoe\nHTTPProxyPassword doesinc/" /etc/freshclam.conf >
/etc/fresclam.conf~
else
sed "s/# Proxy settings/\nHTTPProxyServer
10.34.5.201\nHTTPProxyPort 8080/" /etc/freshclam.conf >
/etc/fresclam.conf~
fi;
mv -f /etc/fresclam.conf~ /etc/freshclam.conf
chmod 700 /etc/freshclam.conf
if [ -f /etc/avg.conf ]; then
echo "Setting proxy params in avg.conf"
if [ -f /etc/avg.conf.bak ]; then
# Take a backup of the original avg.conf if necessary
cp -f /etc/avg.conf /etc/avg.conf.bak
fi;
sed -e s'/proxy = off/proxy = 10.34.5.201:8080/' /etc/avg.conf
> /etc/avg.conf~
mv -f /etc/avg.conf~ /etc/avg.conf
chmod 700 /etc/avg.conf
sed -e s'/proxyLogin = off/proxyLogin = johndoe:doesinc/'
/etc/avg.conf > /etc/avg.conf~
mv -f /etc/avg.conf~ /etc/avg.conf
fi;

if [ -f /opt/BitDefender-scanner/etc/bdscan.conf ]; then
echo "Setting proxy params in bdscan.conf"
if [ ! -f /opt/BitDefender-scanner/etc/bdscan.conf.bak ]; then
cp -f  /opt/BitDefender-scanner/etc/bdscan.conf
/opt/BitDefender-scanner/etc/bdscan.conf.bak 2>/dev/null
sed "s/#HttpProxy =/\nHttpProxy = johndoe:doesinc AT 10.34.5
DOT 201:8080/" /opt/BitDefender-scanner/etc/bdscan.conf >
/opt/BitDefender-scanner/etc/bdscan.conf~
mv /opt/BitDefender-scanner/etc/bdscan.conf~ /opt/BitDefender-
scanner/etc/bdscan.conf

fi;
fi;
fi;
TRKSECTION TRKSCRIPT END
```

-TRKSCRIPT is a section where you can do basically anything you want, so be very carefull with what you put in here because it will have an effect on any machine you boot with TRK3 . It will basically execute as a script in the same environment from the last script run by TRK, /etc/init.d/trklocal
What I put in the example here is to download an adapted version of the ClamAV config file 'freshclam.conf'. Refer to the ClamAV manpages for help on this file, but in here I 've put my proxy parameters too.
I don 't need to tell you what other opportunities this opens up.

If you think this file is too complicated for you to create, don 't worry: just start a TRK, run 'setproxy' and this will generate the file /etc/trk3options.conf
Copy this file to your webserver and you 're all set.

If you run setproxy afterwards, you will override alls settings given by your trk3optionsserver.

4.2.2 Scripts on the computer's local harddisks

Another trigger is to have a script executed from the computer 's local harddisk, allowing you to run commands specifically for the computer. It is in the same format as the trk3options.conf: you have TRKSECTION TRKSCRIPT (only this section) where you write your parameters. This will be generated into a script called /bin/trklocscript-<name of the drive> (f.i. /bin/trklocscript-hda1) and executed at the end of the TRK startup procedure.
The location of this script has to be in the folder .trk (dot trk) on the root of the local partition. In this folder you put the file called trk3local.conf.
A sample trk3local.conf could look like this:

```
TRKSECTION TRKSCRIPT BEGIN

#!/bin/bash
# Script for maintenance and backup on Windows machines
# Clean up this drive a little and do a virusscan on it
cd /hda1/Documents\ and\ Settings
for i in *; do rm -rf "$i"\Local\ Settings\Temp\*; rm -rf
"$i"\Local\ Settings\Temporary Internet Files\*;
done;
virusscan -a fprot
# Now back up the userdata of all local profiles to a network
drive
mount -o username=john,password=doe //10.0.0.5/backups /mnt0
for i in *; do mkdir /mnt0/"$i" 2>/dev/null;  echo "Backing up
documents of $i"; cp -rvf "$i"/My\ Documents /mnt0/"$i"; done
umountallfs

TRKSECTION TRKSCRIPT END
```

4.2.3 Script on the TRK medium

If you want some modifications to TRK and don 't want to fiddle directly into TRK's own scripts, you can add a script to the CD/usb stick/NFS share as /trk3/trklocscript. Unlike the previous trigger, this has to be a normal bash script. The advantage of having scripts outside the TRK files is that you can upgrade your TRK afterwards, without having to redo your work every time: in case of a USB stick or NFS share, you just leave the script where it is, in case of a CD, you re-add the trklocscript to the multisession bootcd.
From this script you can trigger any other script you like, f. i. adding more software to your TRK.
Example trklocscript:

```
#!/bin/bash
# Go to /tmp, which is on virtual shared memory, about half
the size of your RAM
cd /tmp
mkdir /tmp/extrabin
```

```
cd extrabin
# Use the variable $TRKMOUNTDIR so you always copy/execute
from the right location
tar xzf $TRKMOUNTDIR/trk3/extrabinaries.tar.gz
# This will set the PATH globally
export PATH=$PATH:/tmp/extrabin
```

5. Upgrade, update and change of bootmedia procedures

TRK 3.4 is able to boot from more than CD alone. You can **boot it from a USB** stick, USB disk, even a fixed harddisk if you like. And the most nifty feature: **boot it from network over PXE!**
This part of the documentation describes how to change over to different bootmethods, upgrade and customise more of TRK.
Putting TRK to a USb stick shouldn 't take more than 5 minutes.
Ad hoc booting from network is even simpler: make sure your network has a running dhcp server (a standard home router will do), boot TRK on 1 computer and run trkbootnet. Next boot the other computer(s), select the network card as the bootdevice and voilà: a TRK booting from another TRK

Making TRK run from PXE takes a little longer to set up, but it 's worthwhile when you have a big lan to manage. Once set up, upgrading to new versions is piece of cake.

In the next sections you will find instructions on how to set up a bootserver under Linux and there 's also a contributed documentation on how to do it on a Windows machine.

5.1 TRK on CD

TRK is in the first place known as a BOOTCD. So even if burning an isofile to a CD is about the simplest thing to do, I will nevertheless commit a piece on how you can get TRK on a cd properly.

If you really want a no brainer, try the executable, self burning version of TRK from the download section. You don't need any burn software to be installed on your computer, but it does require that you have administrator rights.
And how to do it: download, save locally and double click trinity-rescue-kit.3.4-build-366.exe (or the latest version), answer "Yes" after you made sure you have a blank CD-R in your CD writer and just sit back and relax.

If you want to burn TRK from the isofile, there are several options and programs.
For Windows 7 users it's easy: right click on the file and select the option to burn it to CD.
All other users have several possibilities.
1 free and rather easy possibility is to use MagicIso, which is free for isos up to 300mb. At the end of this document you will find it explained in screenshots.
Another free alternative is CD Burner XP Pro, which is a very good and full fledged CD burning software as a whole by the way.

5.2 How to install/upgrade your USB media to run the latest version of TRK

Installing from Windows

This procedure is also valid for new TRK installations on any bootable media that has:
-been formatted as FAT16 or FAT32 (Syslinux documentation recommends FAT16, but FAT32 is tested and works as well)
-smaller or equal than 1Gb (=max 16k clusters). Bigger (and fat32) is allowed but is subject to less compatibility. Please report me on any PC that refused to boot in that way.
-this partition made active
-**TRK_3-4** as a volume label (exactly like this). Another volume label (e.g. RESCUEDISK is possible, but than you would have to add "vollabel=RESCUEDISK" after each line that says "append initrd=..." in /syslinux.cfg

But this is not guaranteed to boot from any PC that can boot USB. Better is to use trk2usb from TRK itself (when booted from CD of course).

The rest of the installation course is like upgrading, so follow the guidelins below. If you 're unable to bring it to a success using this procedure, I recommend you download the ISO version of TRK, burn and start from that, plug in your USB stick and run trk2usb

Upgrading (and installing continued) from Windows

Prerequisites: 7-zip or any archive application that can open iso files.
-Download trinity-rescue-kit.3.4-build-366.iso (or the latest version)
-Plug in your USB stick with the old version of TRK
-Open the file trinity-rescue-kit.3.4-build-366.iso and extract it to your USB drive root (we will call it the G: drive in this case)
-Open a command prompt 'cmd'
-Go to your USB drive: 'G:'
-'cd trk3'
-'syslinux G:' ('syslinux -f G:' if it refuses because windows reports it to be a fixed drive)
-'exit'
-eject the device and try booting from it (use the 'safely remove hardware' from the system tray)

Installing from Linux

I 'm not describing in detail how you should install TRK from Linux, since it might require you to install special packages and perform complicated operations. Someone who doesn 't fear this will have enough with this short procedure
Here goes:
-prerequisites: mtools, syslinux 3.31
-figure out your USB stick device (most of the time /dev/sda)
-run 'mkdiskimage -4 /dev/sda -s 0 64 32' (from the syslinux package). This will completely erase and format your drive to maximum 1Gb. If you want to pass this limit, use fdisk and mkvfat -f 32 /dev/sda1, but beware this can be less compatible.

-edit /etc/mtools.conf and add the line: `drive c: file="/dev/sda4"`
-'mlabel c:TRK_3-4'
-the next steps are the same as upgrading from Linux, so read below

Upgrading from Linux

-Download trinity-rescue-kit.3.4-build-366.iso (or the latest version)
-Plug in your USB stick with the old version of TRK 3.4
-Check what device id it has been assigned using 'dmesg | tail'
-Assuming your USB stick is /dev/sda and your TRK partition /dev/sda4 and we use /mnt/disk1 as mountpoint: 'mount /dev/sda4 /mnt/disk1'
-'cd /mnt/disk1'
-assuming trinity-rescue-kit.3.4-build-366.iso is downloaded to /home/user and the mount point /mnt/disk2 exists/:
-'mount -o loop /home/user/trinity-rescue-kit.3.4-build-366.iso /mnt/disk2'
-'cd /mnt/disk2'
-'cp -avf * /mnt/disk1/
-'cd trk3'
-'umount /mnt/disk1'
-'./syslinux /dev/sda4'

5.3 Setting up your PXE boot environment

Ever wondered what you could do with that PXE stuff from your network card at boot time? Well you could run a TRK on your network.

1. Prerequisites

Here 's what you need:
-the latest TRK
-A (wired) local area network
-Computers with PXE compliant hardware
-A decent configurable dhcp server
-A Linux machine or any OS that can run an NFS server (not Windows, or else a third party soft)
-A TFTP server (these last 3 things can all run on the same server)

Time to install and configure everything: 15 minutes (if you understand everything directly)

2. Put the TRK files somewhere

Download the latest tar.gz distributed TRK version (version 275 and up) and unpack it somewhere on your Linux nfs/tftp server. In this case we 're going to assume unpacking it to /home/trkfiles. User and group ownership is ok for root, just make sure it 's world readable (is normally ok, but just in case)
commands:
'cd /home/trkfiles'
'tar xzf /tmp/trinity-rescue-kit-3.2-build-275.tar.gz'

You can also copy your files from your TRK CD, usb stick or mount the iso file with a loopback device and copy everything from there.

Keep in mind: This is the location where all of your TRK files will reside. The folder you are about to use will be the same for your TFTP server as for your NFS share. Make sure they are the same, not a subfolder of the other!

There 's another thing you should do, that 's generating a default pxelinux config file (/home/trkfiles/pxelinux.cfg/default). I 've created a small script that does it for you, based on the normal isolinux/syslinux config files. This is something you need to do everytime you upgrade TRK
-Go to /home/trkfiles/trk3 'cd /home/trkfiles/trk3'
-'./mkpxelinux'
The script will prompt you for an nfs path to specify. This is the ip-address of your server combined with the path where TRK resides. This is necessary for the secondary startup phase of TRK. If you omit this, you will get weird results when booting.
An example you could fill in is '192.168.81.5:/home/trkfiles'

3. Setting up DHCP

I 'm going to discuss 2 DHCP servers.

First the ISC dhcp server, which is about the standard in most Linux distributions.
Here 's what the dhcp.conf should more or less look like, depending on your site configuration.
It 's what 's in bold that 's important.
Note that in this situation ALL of your machines will be able to boot from PXE.
If you omit the PXE parameters ('next-server' which is your TFTP server and 'filename') from the general dhcp parameters and put them in the section 'group' (what' s commented out with #), you can specify PXE booting per host based on the host's mac address.

```
allow booting;
allow bootp;
ddns-update-style none;
subnet 192.168.81.0 netmask 255.255.255.0 {
    # default gateway
    option routers 192.168.81.2;
    option subnet-mask 255.255.255.0;

    option domain-name "trinityhome.local";
    option broadcast-address 192.168.81.255;

    # Seting up an ip address is better here
    option domain-name-servers 192.168.81.2;
    option nis-domain "trinityhome.local";

    range dynamic-bootp 192.168.81.128 192.168.81.254;
    default-lease-time 21600;
    max-lease-time 43200;
    # PXE directives
    next-server 192.168.81.5;
    filename "/pxelinux.0";

#    group {
#    host testtrk {
#    hardware ethernet 00:0C:29:A1:E9:E5;
#    fixed-address 192.168.81.253;
#    next-server 192.168.81.5;
```

```
#     filename "/pxelinux.0";
#         }
#     }
```

Second DHCP server is with a Microsoft DHCP.
-Run the DHCP snap-in
-Go to your active scope => scope options => right click in it and select 'Configure options'
-Activate option '066 Bootserver hostname' and option '067 Bootfile Name'.
-Put for hostname the IP-address of your TFTP server alltogether.
-For bootfilename '/pxelinux.0'

4. Setting up a TFTP server (primary bootprocess)

I 'm using the tftp-hpa-0.43a TFTP server, which is recommended by the syslinux developer. Download it from kernel.org:
http://www.kernel.org/pub/software/network/tftp/
Download, unpack, configure and make && make install it.
It 's also possible it was already in your distribution, so you don 't need to compile it.
To have the tftp server run from the commandline as a daemon, using /home/trkfiles as the dir where your TRK resides, type 'in.ftpd -l -s /home/trkfiles'
In fact, this whole procedure so far is also explained on the syslinux homepage.

5. Setting up NFS (secondary bootprocess)

This is for a very basic NFS setup. TRK only needs read only access on NFS, so there 's no big security issue for this simple setup.
NFS relies on 1 configuration file for its shares: /etc/exports
Edit /etc/exports and add the next line, again assuming /home/trkfiles is where your TRK resides. This is the same folder as your TFTP server root!
'/home/trkfiles *(ro,no_root_squash)'
Restart your NFS service.

That 's it, you 're all set. Take a machine in your lan, power it on, press F12 and boot from network with TRK. It should boot even faster than from CD or disk.
When a newer version of TRK appears, all you need to do is download and unpack in the same dir where it resided before and regenerate your PXE config with mkpxelinux

In the event you gave wrong parameters or your NFS has any problem at all to get mounted, TRK will drop to a basic shell allowing you to debug what's happened.

6. Custom security and triggers

Setting it up

In some cases you cannot have the physical console of a machine, but it can be remotely booted from network with TRK on PXE.
You could run a ssh server, but TRK by default will not run one because there would be no security (the default password for root is 'trk').
In this case I 've developed a possibility to have a custom shadow file (=holds encrypted versions of passwords on a Linux machine) or even better the use of RSA public keys.
Furthermore, there 's also a check for a custom, per host trk3options.conf file, should you not have a trk3optionsserver environment (feature since TRK 3.1).

Also it will look for a custom trklocscript, so you can launch anything else you like.
Here 's the way to set it up:
-In the directory where your TRK files reside, create a directory called 'pxeconfig'
-in the directory pxeconfig, create directories named according the mac-address of your custom hosts.
If your hosts network card is 00:0C:29:A1:B9:E5, then create that directory, but use hyphens instead of colons, just to avoid filesystems that don 't accept colons. So in this case create directory '00-0C-29-A1-B9-E5'

Security and other triggers

-Per host shadow file
In the mac-address based directory you can now put a custom shadow file. This will be detected upon startup of TRK on the host with the matching network interface and copied locally.
Once copied, TRK will start a secure shell server allowing remote control of the machine with the matching root password from the custom shadow file.

-Per host authorized_keys file
Another way of having custom security for ssh is the use of rsa key authentication. This is fairly easy to setup. The method of public/private key authentication is standard secure shell and is explained anywhere on the internet. I will therefore not explain how you should generate a public/private keypair.
What 's important is that you copy the public key of the root user to a file called 'authorized_keys'. In general, one key is enough, so it 's ok to copy the file id_rsa.pub or identity.pub to 'authorized_keys' into our /<trkpxefiles>/pxeconfig/<mac-addres>/ directory.
Upon startup, this file will be detected and gets precedence over a custom shadow file. Once this file is copied, TRK will disable password based authentication, so you can only login with your private key

-Per host or general trk3options.conf and trklocscript file
Just like with a trkoptionsserver, you can have a custom trk3options.conf file in which you can specify custom proxy settings or a complete script. Just put the file in /<trkpxefiles>/pxeconfig/<mac-addres>/ for per host trk3options.conf or in /<trkpxefiles>/pxeconfig/ to have this file for general use for all local TRK hosts.
The same principle goes for trklocscript, a script that will get executed when found. Place this script in /<trkpxefiles>/trk3/ and it will get executed on startup.