Image for Linux

User's Guide Supplement

TERABYTE UNLIMITED
10220 Lazy Bear Street
Las Vegas, NV 89131
http://www.TeraByteUnlimited.com
ASP Member

Trademarks

BootIt is a trademark of TeraByte Unlimited.

Microsoft, MS, MS-DOS and Windows are registered trademarks of Microsoft Corporation.

IBM is a registered trademark of the International Business Machines Corporation.

Linux is a registered trademark of Linus Torvalds

All other trademarks and copyrights referred to are the property of their respective owners.

Notices

Image for Linux was compiled using an unmodified version of Open Watcom, which can be found at www.openwatcom.org.

Technical Support Policy

Technical support is provided online. The most current versions of software and documentation (including updates to this manual) are available at www.terabyteunlimited.com.

The Image for Linux home page, with software and documentation update information, and support resources, can be found at http://www.terabyteunlimited.com/imagel.html.

A support knowledge base for all TeraByte Unlimited products, including Image for Linux, can be found at www.terabyteunlimited.com/kb.

Registered users can email their questions to support@terabyteunlimited.com, if no suitable resolution is found via the aforementioned support resources. If the issue is not resolved via email, telephone support may be provided. Unregistered users will be provided technical support and product information through email only. In all cases, TeraByte Unlimited reserves the right to refuse any communication method that would incur a cost.

Ombudsman Statement

This program is produced by a member of the Association of Shareware Professionals (ASP). ASP wants to make sure that the shareware principle works for you. If you are unable to resolve a shareware-related problem with an ASP member by contacting the member directly, ASP may be able to help. The ASP Ombudsman can help you resolve a dispute or problem with an ASP member, but does not provide technical support for members' products. Please contact the ASP Ombudsman online at www.aspshareware.com/omb.

Table of Contents

Introduction and General Information

Image for Linux (IFL) is a backup and restore program that can create, validate, and restore images of your partitions. The typical usage is to create a compressed image of a partition and save it to a file (or set of files) on a hard drive, a network drive, or to optical media such as a CD or DVD. The image created contains an exact copy (or snapshot) of all data on the partition. If it becomes necessary at a later time to restore the image, the resulting restored partition will be an exact copy of the partition as it was when the image was created.

Image for Linux is essentially an adaptation of Image for DOS (IFD) that is designed to run on the Linux operating system. If you are familiar with Image for DOS, you will find Image for Linux to be quite similar in appearance and function. The primary **differences** that you will encounter are summarized below, and will be covered in detail later in this manual.

- The hard drives and CD/DVD devices are listed somewhat differently in the menus
- The mounting and unmounting of partitions and network drives may be required to save, restore, and validate images

The images created by Image for Linux are 100% compatible with, and can be restored by, the other TeraByte Unlimited imaging products; Image for Windows, Image for DOS, and BootIt NG. The reverse is also true. Image for Linux can restore images created by the other TeraByte Unlimited products.

What You Can Do With Image for Linux

- Create, restore, and validate images on unmounted FAT, FAT32, or NTFS partitions
- Create, restore, and validate images on any mounted file system, including network drives
- Create, restore, and validate images on ATAPI and SCSI CD/DVD devices
- Create bootable CD/DVD restore discs.
- Perform imaging operations interactively using the menus, or from the command line

System Requirements

Hardware • IBM-compatible personal computer (i386 or newer)
- 32-MB RAM (minimum)
- Writable CD or DVD drive (optional)

Software • Linux based operating system - Linux kernel 2.6 recommended
- Linux kernel edd module enabled and loaded for restore operations (see page 18)

Hardware Support

Image for Linux (IFL) relies on the Linux kernel to provide access to mass storage devices, such as hard drives, optical drives, and network drives. The newer the kernel version, the better, as far as hardware support is concerned. For the most part, modern Linux distributions running on a 2.6 series kernel are likely to support all hardware devices you may need to access. Assuming the kernel version and configuration supports it, you will be able to create, restore, and validate images on all of the following:

- IDE, SATA, SCSI, USB 1.1/2.0, and IEEE 1394 hard drives
- ATAPI and SCSI CD/DVD devices
- Mounted network drives (SMB, NFS etc.)

Installing Image for Linux

1. Download the Image for Linux archive. The archive is provided in the zip format.

2. Create a directory with a name of your choosing, and extract the contents of the zip archive in that directory. Note that in Linux, you can use the unzip utility for this (e.g. unzip filename.zip).

3. Change to the directory created in step 2 (if you haven't already), and become **root**.

4. Run the command **'chmod 755 setup'** from the command prompt. Doing this will ensure that the setup script is an executable file.

5. Run the command **./setup**. This command will do the following:
   - Ensure that the Image for Linux program **(imagel)** is an executable file
   - Ensure that the **makedisk** script is executable
   - Build CDBOOT.F35. This file is needed to create bootable restore discs (CD/DVD)
   - Check the status of the edd module in the Linux kernel. If you get a message saying there is a problem with the edd module, please refer to the section on edd later in this manual.

6. At this point you can run Image for Linux by typing **./imagel** at the prompt. Note that Image for Linux should always be run **as root** to ensure that you have read/write access to the devices needed to create, restore, or validate images.

7. If desired, you also have the option of running the **makedisk** script to create an Image for Linux boot disk. Please refer to the section entitled **The Image for Linux Boot Disk** for more information on this subject.

You can run Image for Linux from a terminal within the X environment (such as an xterm), or you can also run it from a text console outside of the X environment. As previously mentioned, you will (in most cases) **need to be root** when running the program in order to have permissions to access the devices you need to work with.

When running Image for Linux from a distribution, you will typically not be able to create a reliable image of any partition that is currently mounted. The evidence of this is that the byte-for-byte validation will fail because the partition's contents have changed during the process of creating the image. While some partitions can be temporarily unmounted to solve this issue, this will not be the case with the root partition especially. To create images of partitions that cannot be conveniently unmounted from the running system, a good alternative is to run Image for Linux from a boot disk.

The Image for Linux Boot Disk

This section will cover how to create and use the Image for Linux boot disk that comes with the product. The boot disk image included in the zip archive can be used to create either a floppy disk, or an equivalent CD/DVD. The boot disk (both floppy and CD/DVD) currently supports the following hardware:

IDE, SATA, and USB 1.1/2.0 hard drives
ATAPI CD and DVD devices

Creating the Image for Linux Boot Disk

You have the option of creating the disk from either Windows or Linux.

From Windows:

1. Extract the contents of the downloaded archive into a directory of your choice.

2. Run the MakeDisk program included in the archive (makedisk.exe).

3. Follow the prompts to create your choice of a bootable diskette or a bootable CD/DVD.

4. See the included readme.txt file for additional information.

From Linux:

1. If you haven't already, follow steps 1 through 5 in the **Installing Image for Linux** section.

2. You have a choice of creating either a bootable diskette, or an ISO file that can be burned to CD/DVD.

3. To create a bootable diskette, type **./makedisk /dev/fd0** at the prompt.

4. To create an ISO file, type **./makedisk /iso** at the prompt. This step will create the file IFL.ISO, which can then be burned to a CD/DVD using your choice of burner software.

5. See the included readme.txt file for additional information, including how to create the bootable diskette if you only have a USB floppy drive.

Using the Image for Linux Boot Disk

The floppy diskette and CD/DVD boot disks are functionally identical in terms of capability and hardware support. The difference between them is in how the CDBOOT.F35 file is handled at boot time. CDBOOT.F35 is the file required to be able to create **bootable** restore discs when imaging to a CD/DVD device. Due to space restrictions and other considerations, this file must be built after the disk is booted from. With the CD/DVD version, this happens automatically without user intervention.

With the floppy disk version, the process of building CDBOOT.F35 takes considerably longer because files need to be copied from the diskette after boot. Because this process requires a noticeable amount of time, you will be able to configure the diskette to best suit your anticipated need for CDBOOT.F35. On the first boot from a freshly created diskette, you will be asked whether or not to build the file. If you choose yes, it will retrieve the two required files (vmlinuz and imagel) from the diskette, build CDBOOT.F35, and leave you at the same introductory screen as the CD/DVD does. If you choose no, it will retrieve just imagel from the diskette (which takes much less time), and then go to the introductory screen.

The introductory screen presents some basic information, and prompts you to press <Enter> to start up Image for Linux. From there, you can use Image for Linux in the same manner as if running from a full Linux distribution. The Image for Linux boot disk has 5 virtual consoles available, and you have the option of switching among them by pressing **<Alt+F1..F5>**. The primary reason you may want to do this would be to mount or unmount a hard drive partition, without exiting the Image for Linux program first.

When you do exit Image for Linux by pressing <Esc> from the main menu, you will be left at a Linux command prompt, from which you can start the program again by simply typing 'imagel'. The screen you will see when first exiting Image for Linux will also display some additional information above the prompt. This information will explain how to start up Image for Linux again (by typing 'imagel'), and will summarize how to

switch among the 5 virtual consoles. With the floppy disk version, there will be some additional information displayed as follows:

- The current status of CDBOOT.F35 (built or not built)
- If not built, how to build it now using 'buildcdboot'
- The current CDBOOT.F35 build option on boot (Yes, No, or Ask)
- How to change that option using 'config'

If CDBOOT.F35 is not built at boot, it can subsequently be built at any time by typing '**buildcdboot'** at the prompt. This will require that the Image for Linux boot disk be inserted in the drive. The buildcdboot script will retrieve the files needed from the floppy disk, and then build CDBOOT.F35.

If you wish to change the CDBOOT.F35 build option at boot, you can run **'config'** from the prompt. This script will present you with 3 choices as follows:

- YES (y) – CDBOOT.F35 is always built automatically at boot
- NO (n) – CDBOOT.F35 is built manually (if needed) by running 'buildcdboot'
- ASK (a) – The user will be asked YES or NO at boot time

The choice you make will be saved to the floppy disk, and will determine what is done with CDBOOT.F35 the next time the diskette is booted from. Please note that CDBOOT.F35 is only required if you need to create **bootable** CD/DVD restore disks during the session. If not, the best option will be NO, since that will save time when booting from the diskette.

Note that you can type 'exit' to see the initially displayed information again, after it scrolls off the screen.

Creating a Customized Image for Linux Boot Disk

It is possible to create your own customized Image for Linux boot disks. This will not be covered in detail, but if you do attempt to do this, there are a few key points to be aware of:

- Image for Linux requires that a **/tmp directory** exist in the root file system. Without this directory present, the program will not start up.

- Image for Linux requires the **/proc directory** to gather information about what drives and devices are connected to the system. In order to activate /proc, you need to enable the /proc file system in the kernel configuration under File systems -> Psuedo filesystems -> /proc file system support. To mount and activate /proc on boot, you can use the command 'mount –t proc proc /proc', or you can put an equivalent line in /etc/fstab.

- Image for Linux requires the Linux **edd module** in order to ensure that images are restored using the correct CHS values. **Linux kernel 2.6** is required for this. The edd module is enabled under Processor type and features -> Firmware Drivers -> BIOS Enhanced Disk Drive calls.

- In order for Image for Linux to make use of the edd module, the **/sys directory** must exist, and the sysfs file system must be enabled in the kernel. The sysfs file system option appears in the 2.6 kernel configuration under File Systems -> Psuedo filesystems. In the kernel .config file, you should see the line "CONFIG_SYSFS=y" in the Psuedo filesystems section. To mount and activate /sys on boot, you can use the command 'mount –t sysfs sysfs /sys', or you can put an equivalent line in /etc/fstab.

Using Image for Linux Interactively

This section will cover using Image for Linux interactively through the menus to create, restore, and validate images. Please note that you may use the Esc key to undo menu selections, and move back to the previous menu.

Creating an Image

1. Start up Image for Linux to get to the main menu.

2. Use the arrow keys to select the "Create Image" menu item, and press Enter. The "Save From" window will appear listing all available hard drives. The hard drives listed will be of two general types:

   - Those listed as **/dev/hdx,** such as /dev/hdb and /dev/hdc. These are hard drives connected to an IDE controller.

   - Those listed as **/dev/sdx**, such as /dev/sda and /dev/sdb. These are SATA, SCSI, USB, or IEEE 1394 (FireWire) drives, and are all accessed in Linux under the general category of SCSI disks

   The order of the drives listed will depend on how they are connected to the system, and also the BIOS boot order. If all drives that you expect to see don't appear on the list, try pressing the ESC key, waiting a few seconds, and then selecting "Create Image" again.

3. Select the hard drive that contains the partition you wish to backup, and press Enter. The "Hard Drive n" window appears (where "n" is the applicable hard drive number). The Linux hard drive designation corresponding to Hard Drive n will appear in parenthesis.

4. Select the partition you wish to backup, and press Enter. The "Save To" window appears.

5. From the "Save To" window, select the destination for the backup file(s). Before pressing Enter to proceed:

   - Select **File** if you would like to save the image to file(s) on a mounted partition or network drive. If you are saving the image to a file, do not save it to the same partition you are backing up. If you do, the restored partition will be in an inconsistent state, which can compromise reliability. You do not have to supply a file extension—just the path and file name itself. The extension will be added automatically.

   - Select the **CD/DVD** option if you would like to save the backup file(s) to CD or DVD. Image for Linux can automatically overwrite CD-RW, and DVD+RW media. However, if you wish to use DVD-RW media, it must be either brand new, fully blanked, or fully formatted before being used. To fully blank the DVD-RW media, use your burning software's "full erase" function. (The "quick erase" function will not work for this purpose.)

   - Select the **Partition** option if you would like to save the backup file(s) directly to a partition. Image for Linux supports this option for FAT, FAT32, and NTFS partitions. For other file systems, you will need to mount the partition, and select "File" from the Save To menu.

   - Note that regardless of which backup destination you select, Image for Linux will automatically create backup files with the .IMG extension first, and then creates additional numbered files extensions as necessary. For example, if your backup results in three image files, they will be named BACKUP.IMG, BACKUP.001, and BACKUP.002, in that order of creation. How many image files are created depends

on how large the source data is, whether compression is used, and what "Maximum File Size" setting you use, as described in a later step.

6. The screen that appears next depends on what "Save To" option you selected above:

   - If you selected the **File** option, you will be presented with a dialog box where you can enter (or navigate to) a path and file name to save the image to. The file name used is required to follow the DOS 8.3 file name convention.

   - If you selected the **CD/DVD** option, you will be presented with 2 options on the next screen; ATAPI/SCSI and SG. In general, all of your CD/DVD devices will be listed under one of the two choices, and the other choice will result in a "No usable CD/DVD drive found!" message. Which choice works will depend on your kernel version and configuration. Typically, systems using a 2.4 series kernel will have the devices listed on the SG menu, while systems using a 2.6 series kernel will show the devices under the ATAPI/SCSI choice. Locate the CD/DVD device you wish to save the image to, and select it from the list.

   - If you selected the **Partition** option, you will be prompted with "Save To" options that mirror the "Save From" options that were outlined in step 2 above. After selecting a drive and partition to save to, the final step in specifying the destination will involve providing a path and file name, using the MS-DOS 8.3 naming convention.

   - Note that you may save the image to a directory other than root. In order to do so, you must create the directory structure beforehand, in an environment that supports the file system in use. Since the MS-DOS 8.3 naming convention will be used by Image for Linux, you may want to restrict each folder name to 8 characters or less. If you are saving to a directory other than root, you can specify the path by using either the "\dirname1\dirname2\filename" format, or the "/dirname1/dirname2/filename" format. Both the forward and backward slash characters are acceptable in the path you specify.

7. The screen that appears next will again depend on which "Save To" option you selected above.

   - If you chose to save to a **file** or a **partition**, you will next come to the "Maximum File Size" screen. Your selection on the "Maximum File Size" screen tells Image for Linux how large the backup file(s) that are created can be. For example, if the backup ends up being 3.5 GB in size overall, and you select the "2 GB" option, Image for Linux will create one 2.0-GB file, and one 1.5-GB file. The "698 MB" and "648 MB" selections are intended to create backup files that fit on 700-MB and 650-MB CD-R/RW discs, respectively. You can use this option, for example, if you are saving the backup file(s) to a hard drive initially, and will later burn them to a CD.

   - If you chose to save to **CD/DVD**, you will now be asked to select the CD Write Speed. The default option is "Optimal", and it is recommended that you select this option unless/until you encounter problems. After the CD Write Speed selection, you will be asked if you would like to **validate each disc**. This option is not the same as validating the image itself (see step 8). Instead, it ensures that each CD/DVD disc itself is readable after being written to, and verifies that the data on the disc appears to be the same as the data that was sent to the drive. Per-disc validation has the ability to detect media errors that may have occurred during the disc writing process. If an error is detected, you will be prompted to redo the failed disc. (Without this option enabled, you will only be notified of errors after the backup process is complete.)

8. The next step in all cases will be to choose your **image validation** options. If you answer "Y" for the first validation prompt, you will be asked if you would like the validation to be performed with a full byte-for-byte comparison of the source and backup data. If you perform only a basic validation (i.e. you respond "Y" to the first validation prompt, and "N" to the second), Image for Linux will perform internal consistency checks on the backup file(s), once they are created. Enabling this option increases the overall processing time, but can help ensure that the backup is reliable. If you choose to have a full byte-for-byte validation performed (i.e. you respond "Y" to both validation prompts), Image for Linux will verify that every byte in the source data was backed up correctly, ensuring 100% accuracy. This option generally doubles the processing time of the overall backup operation, but is advisable to use where maximum reliability is required.

9. Once you respond to the image validation prompts, Image for Linux proceeds with the backup operation, and will provide an indication of the backup progress throughout (and also for the validation progress, if any). In the case of saving to CD/DVD, it will prompt you to insert discs when needed, and ask if you would like to erase a non-blank disc when applicable.

The backup and validation operations may be interrupted at any point by pressing the Esc key. Image for Linux will ask you to confirm that you want to cancel before it interrupts the current operation. When all operations are done, the completion screen will appear.

Restoring an Image

1. Start up Image for Linux to get to the main menu.

2. Use the arrow keys to select the "Restore Image" menu item, and press Enter. The "Restore From" window will appear.

3. Select the source location of the backup file, before pressing Enter to proceed:

   - Select **File** if you would like to restore from a backup stored on a mounted partition or network drive.

   - Select the **CD/DVD** option if you would like to restore from a backup on CD or DVD disc.

   - Select the **Partition** option if you would like to restore from a backup on a partition that is not currently mounted. Image for Linux supports this option for FAT, FAT32, and NTFS partitions.

4. The screen that appears next depends on what "Restore From" option you selected above:

   - If you selected the **File** option, navigate to and/or select the desired backup file, noting that the MS-DOS 8.3 naming convention is used.

   - If you selected the **CD/DVD** option, you will be prompted to select either ATAPI/SCSI or SG. These are the same menu choices as when selecting a CD/DVD device to save an image to. Again, the most common choice will be ATAPI/SCSI, but your device may be listed under the SG menu if you are using a 2.4 series kernel. Select the option that lists the device you want to restore from, and then select that device from the list. At that point, you will be prompted to insert the first disc of the backup.

   - If you selected the **Partition** option, you will be presented with the same list of hard drives as when saving an image. Select the hard drive from the list, and then the partition on that drive where the image

to restore is located. Finally, navigate to and/or select the desired backup file, noting that the MSDOS 8.3 naming convention is used.

- Whichever initial "Restore From" option you selected, you should now be at the "Restore To" screen.

5. On the "Restore To" screen, you will again be presented with the list of available hard drives. Select the drive that you want to restore the image to. Once you make a selection, press Enter. The second "Restore To" screen appears.

6. Select the partition or free space entry you wish to restore to, and press Enter. You will be prompted to enter any additional options desired.

7. If you want to use any additional options for the restore, enter them as prompted. Otherwise, leave the text box empty and press Enter.

   - The options that can be used here are A, Z, J, T, and X[mb]. These options can be used in combination, and may be listed in any order. For example: JX10Z. For further explanation of the A, Z, J, T, and X[mb] options, please refer to the "Using Image for Linux from the Command Line" section later in this manual.

   - If the destination you selected in step 6 is a free space entry, the restore operation will begin immediately after you press Enter on the options window.

   - If the destination you selected in step 6 is an existing partition, you will first be warned that the existing data will be overwritten, and will be asked to confirm before proceeding. Then you will be asked if you would like to validate the image before the restore operation.

   - If you choose to have the image validated, Image for Linux will perform internal consistency checks on the backup file(s), prior to the restore operation. Enabling this option increases the overall processing time, but can help ensure that the restore will be reliable.

   - The restore and validation operations may be interrupted at any point by pressing the Esc key. Image for Linux will ask you to confirm that you want to cancel before it interrupts the current operation.

   - When all operations are done, the completion screen will appear.

8. Once the restore process completes, reboot the computer if you are prompted to do so.

Note: If you do not reboot when asked, the operating system will think the partition and file system is as it was before the restore. This could cause data corruption. You can override the reboot prompt by using the /RN switch during a command line restore, but only do this if you are an advanced user, and understand the potential ramifications of not rebooting.

Validating an Image

1. Start up Image for Linux to get to the main menu.

2. Use the arrow keys to select the “Validate Image” menu item, and press Enter. The “Validate” window will appear.

3. Select the location of the backup file you wish to validate, before pressing Enter to proceed:

   - Select **File** if you would like to validate a backup stored on a mounted partition or network drive.

   - Select the **CD/DVD** option if you would like to validate a backup on CD or DVD disc.

   - Select the **Partition** option if you would like to validate a backup on a partition that is currently not mounted. Image for Linux supports this option for FAT, FAT32, and NTFS partitions.

4. The screen that appears next depends on what “Validate” option you selected above:

   - If you selected the **File** option, navigate to and/or select the desired image file to validate.

   - If you selected the **CD/DVD** option, you will be prompted to select either ATAPI/SCSI or SG. These are the same menu choices as when selecting a CD/DVD device to save or restore an image. Select the option that lists the device where you want to validate an image, and then select the specific device from the list. At that point, you will be prompted to insert the first disc of the backup.

   - If you selected the **Partition** option, you will be presented with the same list of hard drives as when saving or restoring an image. Select the hard drive from the list, and then the partition on that drive where the image to be validated is located. Finally, navigate to and/or select the desired image file to validate on the selected partition.

5. Once you point Image for Linux to the image file to be validated, the validation process begins automatically. The process may be cancelled at any time by pressing the Esc key. Image for Linux will confirm before canceling.

Using Image for Linux from the Command Line

- A summary of the Image for Linux command line options can be viewed by running the following command at the prompt: **./imagel /?**

- When running Image for Linux from the command line, you will often need to include references to hard drive numbers, and/or partition IDs. To determine the correct hard drive number or partition ID, carry out steps 1 through 3 under the section titled **Creating an Image**, being sure to select the hard drive whose number and/or partition IDs you need to obtain. The hard drive number will then appear in the format "Hard Drive n" (e.g. "Hard Drive 0" or "Hard Drive 1").

- For each hard drive, the partition ID for each partition is shown in parenthesis in the middle of each partition description, and will consist of either two or four characters. The partition ID is entered on the command line in the format 0xID, where ID is the two or four character partition ID that you see when looking at the hard drive in Image for Linux. For example, if the partition ID is shown as (8C) in Image for Linux, you would enter that as 0x8C on the command line. A four character partition ID such as (103E) would be entered as 0x103E.

- For partition IDs of 9 or below, a single digit may be used in place of hexadecimal notation (e.g. 1 is equivalent to 0x1, and 5 is equivalent to 0x5).

- Per-disc validation [R] is only applicable when saving images to a CD or DVD drive. This option ensures that discs are readable after being written to, and verifies that the data on the disc appears to be the same as the data that was sent to the drive. Per-disc validation has the ability to detect media errors that may have occurred during the disc writing process. If an error is detected, you will be prompted to redo the failed disc. (Without this option enabled, you will only be notified of errors after the backup process is complete.)

- As long as the file CDBOOT.F35 is in the current directory, a bootable CD/DVD will be created automatically when burning directly to a CD/DVD drive.

- Note that under certain configurations, hard drive numbers may be different in Linux than they are in DOS, Windows, or other environments.

Creating an Image from the Command Line

imagel /Chx[VBJRW] [/U] /CD[S]n|d:\file.img[;s]

Source options: /Chx[VBJRW] [/U]

/Chx	create image from hard drive **h** (0-9), partition ID **x**
V	validate image after creating
VB	do a byte for byte validation
J	don't eject CD/DVD disc after operation completes
R	perform per disc validation (only valid when saving to CD/DVD)
W	use raw mode (backup all sectors rather than just used sectors)
/U	don't use compression when creating image

Destination when saving to a CD/DVD device: /CD[S]n

/CDn	save image to CD/DVD device n (omit n to get a list of devices)
S	specify that the CD/DVD device is an SG device

Destination when saving to a file: /path/filename[;s]

s	limit file size on creation: s=0=2GB, 1=698MB, 2=648MB

Destination when saving to a partition: d:/path/filename[;s]

d	hard drive number + partition ID
s	limit file size on creation: s=0=2GB, 1=698MB, 2=648MB

Command Line Examples for Creating an Image

imagel /C10x03 /mnt/redhat	Create an image from hard drive 1, partition ID 03 and save it to the file /mnt/redhat.img.
imagel /C10x03VB /mnt/redhat;1	Same as above, but do a byte-for-byte validation and limit the file size to 698 MB
imagel /C00x01VB /CD0	Create an image from hard drive 0, partition ID 01 and save it to CD/DVD device 0. Do a byte-for-byte validation of the image.
imagel /C00x01VBR /CDS0	Same as above, but also do a per disc validation [R], and specify that CD/DVD device is an SG device [S]
imagel /C00x02 00x01:/redhat	Create an image from hard drive 0, partition ID 02, and save it the file redhat.img on hard drive 0, partition ID 01

Restoring an Image from the Command Line

imagel /R[h][x][NVAXMJT] [/O|/C[R]] /CD[S]n[file]|d:\file.img

Target options: /R[h][x][NVAXMJT] [/O|/C[R]]

/R[h][x]	restore image [to hard drive **h** (0 – 9)] [to partition ID **x**]
N	don't prompt for reboot after restore
V	validate the image before restoring
A	set the restored partition active
M	restore to the first free block of space where the image will fit
J	don't eject CD/DVD disc
T	install a standard MBR on the target drive
X[mb]	expand restored partition to fill free space [fill free space minus mb MB]
/O	overwrite the target partition without warning
/C[R]	clear MBR and EMBR before restore [also restore NT signature]

Source when restoring from a CD/DVD device: /CD[S]n

/CDn	restore image from CD/DVD device n (omit n to get a list of devices)
S	specify that the CD/DVD device is an SG device

Source when restoring from a file: /path/filename

Source when restoring from a partition: d:/path/filename

d	hard drive number + partition ID

Command Line Examples for Restoring an Image

imagel /R0 /mnt/redhat.img	Restore the image file at /mnt/redhat.img to hard drive 0
imagel /R0VAT /O /mnt/redhat.img	Same as above but also validate before restoring [V], set the restored partition active [A], install a standard MBR on hard drive 0 [T], and overwrite the target partition without confirmation [/O]
imagel /R1VM /CD1	Restore image from CD1, and restore it to the first free space area on hard drive 1 where the image will fit [M]. Validate the image before restoring [V]
imagel /R0VN 10x02:/redhat.img	Restore image file redhat.img from hard drive 1, partition ID 02 to hard drive 0. Validate the image before restoring [V], and don't prompt for reboot after the restore [N].
imagel /R0VNX /O /C 10x02:/redhat.img	Same as above, but also expand the partition to fill available free space [X], overwrite the target partition without confirmation [/O], and clear the MBR on hard drive 0 before restoring [/C].

Additional Notes on Image for Linux Command Line Restores:

- The restored partition will go to the same hard drive number (unless overridden) and physical location on the drive as it was backed up from.
- If the source partition was a volume and there is now no extended partition at the original location, Image for Linux will attempt to create the original extended partition. If Image for Linux cannot create the extended partition, it will be restored as a primary partition.
- If the source partition was originally a primary partition, and an extended partition now encompasses that location, it will be restored as a volume.
- If an existing partition or volume occupies the same starting location as the partition to be restored, a warning message will be issued before overwriting that partition or volume. (Unless this warning message is suppressed, as described in the table above.)

Validating an Image from the Command Line

imagel /V /CD[S]n|d:\file.img

Validate an image on a CD/DVD device: /V /CD[S]n
/CDn validate image on device n (omit n for a list)
S specify that device is an SG device

Validate an image accessed as a file: /V /path/filename

Validate an image on a partition: /V d:/path/filename
d hard drive number + partition ID

Command Line Examples for Validating an Image

imagel /V /mnt/redhat.img	Validate the image file at /mnt/redhat.img
imagel /V /CD0	Validate the image on CD/DVD device 0
imagel /V /CDS0	Same as above, but specify that CD/DVD device 0 is an SG device [S].
imagel /V 10x02:/redhat.img	Validate the image file redhat.img on hard drive 1, partition ID 02

Using IFL.INI and Environment Variables

Image for Linux offers several advanced configuration options. These options may be specified in a user-created IFL.INI file, or by using environment variables. Some options may be specified using either method. A table summarizing the available IFL.INI and environment variable options is included at the end of this section. Note that if a particular setting is specified in both the IFL.INI file and in an environment variable, the setting specified in the environment variable will take precedence.

The IFL.INI File

The IFL.INI file (the filename IFL.INI must be upper case) can be created with any text editor, and can contain one or more settings. Image for Linux will look for IFL.INI in the current directory only. The following is an example of what the contents of an IFL.INI can look like:

CONSOLE=1
CDRS=16

Image for Linux Environment Variables

To set an environment variable in Linux, you need to use the **'export'** command, either from the command prompt prior to running Image for Linux, or in a script. The variable names, such as IFL and IMSG, must be upper case. The following are some examples:

export IFL=CONSOLE (use text mode interface)
export IFL=CDWS:4 (set CD write speed to 4)

To assign more than one IFL option on a line, you need to separate the options with a semicolon, and enclose the entire option string in single quotes. Here is an example:

export IFL='CDWS:4;CONSOLE'

If an option can contain spaces, such as the IMSG option allows for, it must also be enclosed in quotes:

export IMSG='Validating the Redhat image file'

To see a list of all currently assigned environment variables, you can use the **'env'** command. To see one particular variable, you can use the **'echo'** command – for example 'echo $IFL'. To unassign an environment variable, you can use the export command with the –n option – for example 'export –n IFL.

Option	Description	IFL.INI	Environment Variable
CDCMZ	If you experience errors or problems when a CD is being closed, then you may need to use this option.	CDCMZ=0 CDCMZ=1	IFL=CDCMZ
CDRS	Sets the CD read speed. For example: CDRS:8 sets the read speed to 8X. These speeds are CD-based; multiply by 8 to approximate DVD speeds (e.g. to read a DVD disc at 2X, use CDRS:16).	CDRS=4 CDRS=8	IFL=CDRS:8
CDWS	Sets the CD write speed. For example: CDWS:8 sets the write speed to 8X. These speeds are CD-based; multiply by 8 to approximate DVD speeds (e.g. to burn a DVD disc at 1X, use CDWS:8). **Note:** If you are having problems burning a reliable CD/DVD, you may need to slow the write speed down using this option.	CDWS=4 CDWS=8	IFL=CDWS:4
CONSOLE	Disables the Image for Linux graphical interface, and forces text-only mode. This option is only valid when Image for Linux is run from the command line (or from a script).	CONSOLE=0 CONSOLE=1	IFL=CONSOLE
IAR	Image Auto Response value. Set this to Y or N to auto respond to 'Y'es or 'N'o prompts and error messages. You can use errorlevel in a script to determine if the operation was successful or not.	IAR=Y IAR=YA	IFL=IAR:Y
IBXT	When using the Burn Extra Track option, you will not be prompted for the last CD during a restore. This option may not work with all CD or DVD drives.	IBXT=0 IBXT=1	IFL=IBXT
IGNVLAB	Ignores the use of volume labels for partition names. This option has no affect if there is no EMBR on the drive.	IGNVLAB=0 IGNVLAB=1	IFL=IGNVLAB
IMSG	The contents of this variable are displayed below the progress bar during the image or restore process. The message must be less than 80 characters.	N/A	IMSG=Creating the image…
SEQVOLID	This option only applies to volumes within an extended partition. When enabled, the SEQVOLID option allows you to specify the volume to be backed up to—or restored over—using sequence numbers, rather than the actual volume ID. The sequence numbers must be specified in hexadecimal form, and are determined by the order of the volumes within the extended partition, beginning at 1. To determine what volume ID to use with this option, first enable the option, and then either: (1) Use the interactive method as described in this manual ; or (2) Run the command line: imagel /Ln - (n = hard drive #)	SEQVOLID=0 SEQVOLID=1	IFL=SEQVOLID
TZ	This option sets the time zone to be used by Image for Linux. When you are saving images to NTFS partitions or CD/DVD discs, using the correct time zone will ensure that the date/time stamps of the image files will be correct when they are viewed within Windows. Please refer to http://terabyteunlimited.com/kb/article.php?id=260 for more information.	TZ=AAA[+/-]HBBB	TZ=AAA[+/-]HBBB

Determining Whether EDD is Enabled and Loaded

When you run the **setup** script, as described in the **Installing Image for Linux** section, it will check the status of the edd module on your system. Before exiting, the setup script will report the status of edd, and refer you to this section of the manual if it detects a problem.

EDD (or edd) is a piece of code in the Linux kernel that can be either enabled or disabled in the kernel's configuration. **Linux kernel version 2.6 is required** in order for Image for Linux to recognize the edd module. If enabled and loaded, edd will cause the **/sys/firmware/edd** directory to be created, and that directory will contain data that Image for Linux uses to determine the correct CHS geometry for hard drives. The absence of edd will not affect the creation and validation of images, but it can affect restore operations, especially when **restoring to an empty hard drive** (empty meaning the drive currently has no partitions). Without edd, Image for Linux will attempt to use other means to determine the geometry, but the result may not be correct.

Some distributions will already have edd enabled and in use, while others may not. It is suggested that users follow the steps listed below to determine the status of edd. Again, keep in mind that while Image for Linux can otherwise be used with older kernels, it will **only** be able to make use of the edd module if you are running **kernel version 2.6.** If unsure, you can determine your kernel version with the command **'uname –r'.** The following steps assume that you are using kernel version 2.6.

Step 1: Determine whether edd is already loaded and in use.

If the directory **/sys/firmware/edd** exists, that is all you need to know. This indicates that edd is both enabled in the kernel, and currently in use. If not, move on to Step 2.

Step 2: Determine if edd is enabled in the kernel.

It is possible for the edd module to be enabled in the kernel, but not be currently loaded. To determine if this is the case, become **root** and run the command **'/sbin/modprobe edd'.** If the command gives no response and simply returns to the prompt, this usually means that the module does exist, and has now been loaded. You can verify this by again checking for the **/sys/firmware/edd** directory as described in step 1. If that directory now exists, that means that the edd module exists and is now loaded, and you can move on to step 3.

If the edd module does **not** exist, you will have gotten an error message from the modprobe command above, and the **/sys/firmware/edd** directory will still not exist. This means that the Linux kernel was not compiled with edd enabled. Correcting this situation will require that the kernel be reconfigured and recompiled with edd enabled. The subject of configuring and compiling the kernel is beyond the scope of this manual, but if you are familiar with compiling the Linux kernel, the edd option can be found in the kernel configuration under Processor type and features -> Firmware Drivers -> BIOS Enhanced Disk Drive calls.

- Note that all currently loaded kernel modules can be listed with the command **'/sbin/lsmod'**. When edd is loaded, it will show up in that list. This is another way to verify that edd exists and is loaded.

- Also note that it is possible that edd is actually enabled and in use, but is **not** listed by **'/sbin/lsmod'.** This will be evidenced by the existence of the **/sys/firmware/edd** directory described in step 1, despite the fact that lsmod does not list edd. This will be the case if edd is compiled directly into the kernel, rather than as a loadable module. This situation is functionally equivalent to the edd module being loaded, and is not a problem.

Step 3: Configure the system to load edd automatically on boot (if desired).

If you were able to successfully load the edd module manually in step 2, you may now want to configure your system so that this happens automatically on boot. Doing this will prevent the situation where you forget to load the module manually before performing a restore operation.

There are several ways to get edd loaded on boot, but the following two methods should cover the majority of situations. The first method can be used for Debian and Debian-based distributions, while the second method can be used for Redhat and Redhat-based distributions.

1. Debian and Debian-based distributions (Debian, Ubuntu, Mepis, Linspire, Xandros etc.)

Become root, and add 'edd' as a line in the file /etc/modules. The file /etc/modules should already exist, although it may currently be empty. This file contains a list of modules to be loaded automatically when the system boots that would not otherwise be loaded.

2. Redhat and Redhat-based distributions (Redhat, Fedora, Mandriva etc.)

Become root, and add the following line to the file /etc/rc.d/rc.local:

/sbin/modprobe edd

The rc.local file contains commands to be executed automatically on boot.

Linux Help Topics

While using Image for Linux is very similar to using Image for DOS, it will be beneficial at times to understand a few key areas of Linux in order to get the most out of the program. This section is intended to provide some introductory information for those unfamiliar with Linux.

Mounting and Unmounting Partitions

In Linux, a partition must be mounted in order to access the files on it. In some cases, partitions are mounted automatically when the system boots, based on a table contained in the text file **/etc/fstab**. That file can be edited to mount additional partitions automatically. However, doing that is not necessary to be able to save an image to a partition, or to restore one from it.

If you want to save an image to a file on a certain hard drive partition (or restore one from it), you have two choices. One is that you can access the partition directly (without mounting it) by choosing to save the image to a **partition** in Image for Linux, rather than to a **file**. In order to do this, however, the partition's file system must be FAT, FAT32, or NTFS. For these file systems, Image for Linux can internally mount the partition and be able to read/write files on it. For others file systems such as Ext2/3 and ReiserFS, the partition being saved to or restored from must be mounted first. There are just a few commands that you need to know when mounting and unmounting partitions.

The first is the **'df'** command, which is the command to list all currently mounted hard drive partitions, as well as any mounted CD/DVD drives, floppy drives etc. This command can be used to see if a partition is already mounted or not, and also as a way to verify that a mount or umount command has worked. Most versions of df

support the –T option, so that you can use the command **'df –T'** to also display the file system in use for each mounted partition.

The second is the **'mount'** command, which is the one used to actually mount partitions. Just typing 'mount' without any parameters is another way to get a list of currently mounted file systems (this will also include other mounted file systems such as proc and sysfs). To actually mount a partition, the mount command has the following basic syntax:

mount [-t fs] [-o options] /dev/xxx mountpoint

The /dev/xxx is the device to be mounted, and the mountpoint is the directory to mount it on. In most cases, the –t and –o options can be omitted, so that the command line needed will typically come down to something like the following basic examples:

1. Mount a FAT32 partition dev/hda1 at the mount point /mnt
   mount /dev/hda1 /mnt OR mount -t vfat /dev/hda1 /mnt

2. Mount an Ext2 partition /dev/sdb4 at the mount point /home/user/mount1
   mount /dev/sdb4 /home/user/mount1 OR mount -t ext2 /dev/sdb4 /home/user/mount1

3. Mount a ReiserFS partition /dev/hdb2 at the mount point /home/user/mount2
   mount /dev/hdb2 /home/user/mount2 OR mount -t reiserfs /dev/hdb2 /home/user/mount2

The directory used as the mount point in the command must already exist, and should be a directory that contains no files. Specifying the file system with the –t option is usually not necessary, but is shown above for completeness.

To unmount a partition, the **'umount'** command is used. Note that the spelling of this command i**s** umount, and not unmount. The syntax for umount is '**umount dir | device'.** In other words, you can give either the directory where the partition is mounted, or the partition (device) itself as the command line parameter for umount. Here are some examples:

1. Unmount the partition mounted at /mnt
   umount /mnt

2. Unmount the partition /dev/hda1
   umount /dev/hda1

In the course of mounting and unmounting, it may become necessary to create additional mount points. For this, the **'mkdir'** command can be used to create directories, while the **'rmdir'** command can be used to remove them. These two commands are very simple to use:

mkdir /mydir -> create the directory /mydir

rmdir /mydir -> remove the directory /mydir

You can also refer to the following TeraByte Unlimited KB article, which covers some additional information on working with partitions in Linux:

http://terabyteunlimited.com/kb/article.php?id=131

Working with Network Drives

Two of the more common network file systems available in Linux are smbfs (Samba), and nfs (network file system). These filesystems can both be mounted over a network connection, and then accessed by Image for Linux as a network drive from a mount point in the local filesystem.

1. Samba shares (smbfs)

In order to access a Samba share from Image for Linux, you will first need to mount that share in Linux by using either the **mount** command or the **smbmount** command as follows:

mount –t smbfs [-o options] //server/share mountpoint

smbmount //server/share mountpoint [-o options]

The two commands above are the same, because using mount with the –t smbfs option is equivalent to using smbmount directly. As an example, if you had a Windows host named **winxp** with a shared partition named **driveh**, and you wanted to mount it on a directory named **/home/user/driveh**, the command to mount it from Linux would be:

smbmount //winxp/driveh /home/user/driveh

Typically, you will be prompted for a password to gain access to the share, although that information can also be included on the command line. The man page for smbmount (man smbmount) contains details for all options that can be used with smbmount. You can verify that the share is mounted by using the same commands as would apply to a local hard drive partition; ‘df’ and ‘mount’. The mounting of Samba shares can also be accomplished with one of several GUI programs that are available for this purpose.

Once mounted, Image for Linux can access the share through the file system, just as it would for a local file system. You will be able to create, restore, and validate images on the mounted network share, provided that you have read/write access to the share.

2. NFS shares (network file system)

NFS shares can be mounted and accessed in much the same way as Samba shares. The mount command to do that is as follows:

mount -t nfs server:share mountpoint

As an example, if the server was named **debian,** and had an nfs share named **/storage**, and you wanted to mount that share at **/home/user/images**, the command line to mount that share would be:

mount -t nfs debian:/storage /home/user/images

Using nfs requires that the Linux kernel be configured to support nfs, and also that the correct nfs software packages are installed. It also requires that the nfs shares be properly configured. Although most leading distributions support using nfs, the details of how to go about setting it up will differ. It is suggested that you refer to your distribution’s documentation for more information. The following web site is also a good source of information on nfs: http://nfs.sourceforge.net/