

Debian Jigdo mini-HOWTO

Peter Jay Salzman

[<p@dirac.org>](mailto:p@dirac.org)

Copyright © 2001 Peter Jay Salzman

v0.116, 2002-12-15

Abstract

Getting Debian ISOs has always been a painful, slow and supremely inefficient process. Jigdo is a new tool for obtaining Debian ISOs in an easy, fast and very efficient manner. This HOWTO describes why you should use jigdo, a little bit about how it works and how you use it to get and update Debian ISOs.

Jigdo is a very general tool, and isn't tied specifically to Debian ISOs. The jigdo tools can be used to make any ISO available for download in the same easy, fast and efficient manner they're used for Debian ISOs. This HOWTO will cover this as well, but we'll focus primarily on downloading Debian ISOs.

[<p@dirac.org>](mailto:p@dirac.org) / www.dirac.org/p.

Distributed subject to the GNU General Public License, version 2.

Table of Contents

1. Administrata	1
1.1. Authorship and Copyright	1
1.2. Acknowledgements	1
1.3. Comments and Corrections	1
1.4. Latest Version And Translations	1
2. Why jigdo?	2
2.1. How Does One Get A Debian ISO Image Set?	2
2.2. Why Not Download The Whole ISO Image?	2
2.3. Why Not Use The Pseudo Image Kit (PIK)?	2
2.4. What Is Jigdo?	3
3. How Jigdo Works (optional)	4
3.1. Preparing The ISO For Download	4
3.2. The .template File	4
3.3. The .jigdo File	5
3.4. Downloading The Image	5
4. Downloading Your First Image (In 5 Easy Steps)	6
4.1. Install Jigdo	6
4.2. Download The .template And .jigdo Files	6
4.3. Run jigdo-lite	6
4.4. Specify A Mirror	7
4.5. Downloading Of The ISO	8
5. Updating Your Image	10
6. Frequently Asked Questions	13
6.1. [11 Aug 2002]: Can I have two jigdo-lite sessions at the same time in the same directory?	13
6.2. [11 Aug 2002]: Why aren't the translations of this HOWTO on LDP?	13
6.3. jigdo takes a bit long to download the files because wget keeps disconnecting and then reconnecting to the FTP server for each file. Is there a way to make it faster?	13
6.4. How can I make jigdo use a proxy?	14
6.5. What do I do if my jigdo download gets interrupted?	14
6.6. My jigdo download won't complete because the .jigdo file is broken. When I download a new, fixed .jigdo file, do I need to download all the data over again?	14
6.7. Can I use jigdo to download images for DVD?	14
6.8. Can I burn the .iso.tmp file to CD?	14
6.9. Why doesn't jigdo work? It downloads some packages and deletes them. I know it doesn't write them to the iso.tmp file because the file size doesn't change!	15
6.10. I'm having trouble getting jigdo-easy to work.	15
6.11. I'm having trouble getting jigdo to download Sarge or Sid.	15
6.12. Jigdo-lite is too verbose. How can I supress some or all of its messages?	15
6.13. Can I use jigdo on platforms other than Linux?	15
7. Errata	16
7.1. jigdo-easy	16
7.2. GUI Interface	16

Table of Contents

7. Errata

7.3. jigdo-file-cache.db.....	16
7.4. Resources.....	16

1. Administrata

1.1. Authorship and Copyright

This document is copyright (c) 2001 Peter Jay Salzman, <p@dirac.org>. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1, except for the provisions I list in the next paragraph. I hate HOWTO's that include the license; it's a tree killer. You can read the GNU FDL at <http://www.gnu.org/copyleft/fdl.html>.

If you want to create a derivative work or publish this HOWTO for commercial purposes, contact me first. This will give me a chance to give you the most recent version. I'd also appreciate either a copy of whatever it is you're doing or a spinach, garlic, mushroom, feta cheese and artichoke heart pizza.

1.2. Acknowledgements

Originally, I was going to thank the author of jigdo, [Richard Atterer](#), simply for writing jigdo. Anyone who has obtained Debian ISOs by other means will know why. However, my thanks needs to go further. This HOWTO started out as some webpages I wrote about my experience with jigdo. Richard took the time to email me extensive corrections, clarifications and answers to questions I had about jigdo. Since then, he has read my work many times. Richard is a developer who not only cares about his work, but also about the people who use it. Sadly, this is becoming less common in this busy world we live in. Thanks, Richard!

I'd also like to thank [Conrad Wood](#), [Elcio Mello](#) and [Marcelo Ramos](#) for translating this mini-HOWTO. I feel totally honored that they have found my words worthy of their time and effort. Thanks, guys!

1.3. Comments and Corrections

I care a great deal about the people who use this document. Even mini-HOWTOs take a long time to write, and I wouldn't have invested so much effort into something people don't understand. If you have comments, corrections or suggestions, even in matters like writing style, don't hesitate to email me. As long as I'm not totally swamped by my PhD, I'll do my best to respond to each email I receive about this mini-HOWTO.

1.4. Latest Version And Translations

Conrad Wood <cnw@conradwood.net> has translated this HOWTO to German.

Elcio Mello <mello@ajato.com.br> has translated this HOWTO to Portuguese.

Marcelo Ramos <mramos@montevideo.com.uy> has translated this HOWTO to Spanish.

The translations are available from <http://www.dirac.org/linux/debian/jigdo>. See [Section 6.2](#).

The stable English version can be found at The Linux Documentation Project: <http://tldp.org/docs.html> in the mini-HOWTO section. If you want to see the work in progress, you can get the "bleeding edge" version from <http://www.dirac.org/linux/debian>. If you'd like to translate this mini-HOWTO to another language, please contact me at <p@dirac.org>.

2. Why jigdo?

2.1. How Does One Get A Debian ISO Image Set?

If you want your own set of Debian CDs there are many ways of getting them. One way is to buy them from [vendors](#) who sell Debian CDs. This has some merit since some of the vendors donate money back to the Debian project. Your donations help make sure that Debian is around for a long time.

Another way of getting a set of Debian CDs is to burn your own set. This first entails obtaining an ISO image and then burning that ISO image to a blank CD. Before jigdo, there were two ways of creating Debian CDs:

1. Downloading the entire ISO
2. Using the pseudo-image kit (PIK)

This document is about the new and better way of obtaining Debian ISO images, using a tool called jigdo. In fact, the PIK is now deprecated. The canonical method of getting Debian ISO images is with jigdo.

2.2. Why Not Download The Whole ISO Image?

There are mirrors which offer http and ftp downloads of Debian ISOs. The problem is that there are very few mirror sites, and their bandwidth can't support everyone who wants Debian ISOs. For example, fsn.hu has reportedly saturated the connection of its provider. The outgoing traffic reaches a few terabytes per month!

In addition, Debian testing and unstable get updated often. Your ISOs may become outdated the same day you download them unless you find some sneaky way of updating them like mounting the ISO on a loopback device and using rsync (which is what the PIK does). So if you want up-to-date ISO images, you must download a new set of ISO images every day. Clearly, this is not the way you want to obtain Debian ISOs!

Even if you want to download the stable ISO images, they still get updated every few months. Downloading the ISO images will give you up-to-date images for a few months, but every time a new revision of Debian stable is released, you'll need to go through the painful process of downloading the entire ISO set from scratch. This is not a good use of your time and the mirror's resources.

2.3. Why Not Use The Pseudo Image Kit (PIK)?

The PIK addresses most of the problems of downloading entire ISO images. The downloads are fast, and the PIK uses rsync to update only those portions of an ISO image that need to be updated, so it's an efficient way of keeping your ISO set up-to-date. However, there are some hefty problems with the PIK:

- It's difficult to use and not very user friendly.
- You can't use the PIK to download testing and unstable ISO sets.
- The PIK relies on rsync which is CPU-intensive for the server. If too many people use PIK with the same server, it would go up in smoke. Even if the PIK is made more friendly for the user, it's unacceptably unfriendly for the mirrors.
- The PIK uses rsync, which is blocked by many of the stricter firewalls. So even if you wanted to use that nice fast corporate network at work, you might run into problems using the PIK.
- Each image needs to be stored on the server. That was OK in the good old Potato days, when the 28 CD images "only" took 17 GB. Starting with Woody, the 96 CDs need 57 GB or so. Now imagine

that we also want to offer DVDs and this figure doubles.

2.4. What Is Jigdo?

Jigdo (which stands for "Jigsaw Download") was written by [Richard Atterer](#) and is released under the GNU GPL. It's a tool that allows efficient downloading and updating of an ISO image. Any ISO image. Jigdo is not Debian specific, however Debian has chosen it to be the preferred method of downloading ISO images.

The jigdo tool comes with two utilities:

- `jigdo-file` is used by the person offering the ISO image. It enables anyone to download that image by creating a `.jigdo` and `.template` file for the image.
- `jigdo-lite` is used by people who want to download the ISO image. It downloads the image using the image's `.jigdo` and `.template` files which were created by `jigdo-file`. If your main concern is simply downloading Debian ISO images, you'll just be using `jigdo-lite`.

A common misconception is that jigdo creates ISO images; it doesn't. `Jigdo-file` simply allows people to download an ISO image by creating a `.jigdo` and `.template` file. The people who want to download the ISO image will get these two files and use `jigdo-lite` to download the image. The ISO image needs to be made in advance, before `jigdo-file` is used, and that's usually done with a utility like `mkisofs` or `debian-cd`.

Jigdo addresses all the problems with the other two methods of obtaining Debian ISO images:

- It's much faster than downloading the entire ISO image.
- Unlike downloading the entire ISO image, it can take an outdated CD (or a loop mounted outdated ISO image), download *only* the files that have changed since the CD (or ISO image) was created and create a new updated ISO. Very similar to how you use `cvs` to update source code.
- `jigdo-lite` is much easier to use than the PIK.
- `jigdo-lite` uses `wget` which, by default, uses `http` to transfer files. The PIK uses `rsync`. While `rsync` may be blocked by some firewalls, the only firewalls that block `http` are the ones from which you shouldn't be using `jigdo` to begin with. You'll almost never run into firewall problems with `jigdo-lite`.
- `jigdo-lite` is much more user friendly than the PIK.
- `jigdo-lite` is much more server friendly than the PIK.

Clearly, `jigdo` is the best method of obtaining Debian ISO images.

3. How Jigdo Works (optional)

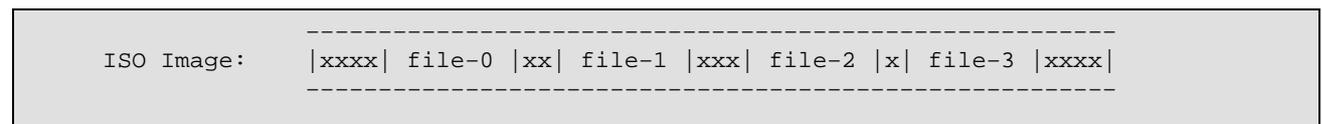
You don't need to know this material to use jigdo, but it may help demystify what jigdo does. If you're not interested in the details, simply fast forward to [Section 4](#), "How Do I Use Jigdo".

There are two components to jigdo:

- `jigdo-file`: Prepares an ISO for download (used by the person offering the ISO)
- `jigdo-lite`: Downloads the ISO (used by the person downloading the ISO)

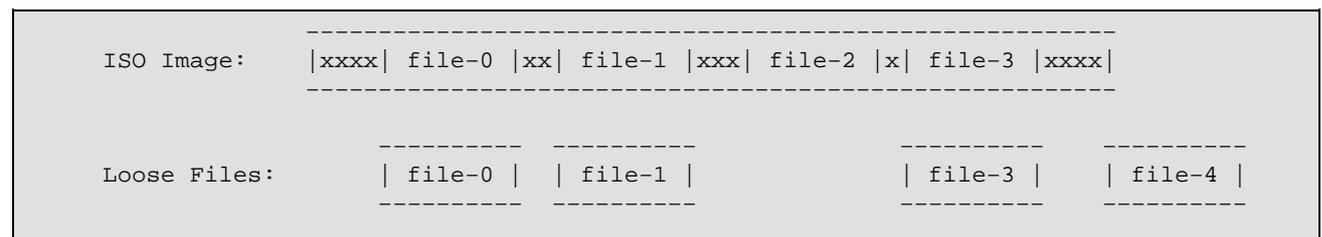
3.1. Preparing The ISO For Download

A CD image is a filesystem called iso9660, but for this discussion, we can safely talk about a CD image as being a big file called an "ISO image" (about 650MB) that contains files at various offsets. For instance, if a CD contains a 567 byte file named README, the ISO image might contain the README file's contents between offsets 20480000 and 20480567. You can visualize a CD image as:



The "x" areas of the image contain things like directory information, zero padding, disk name, boot block, etc.

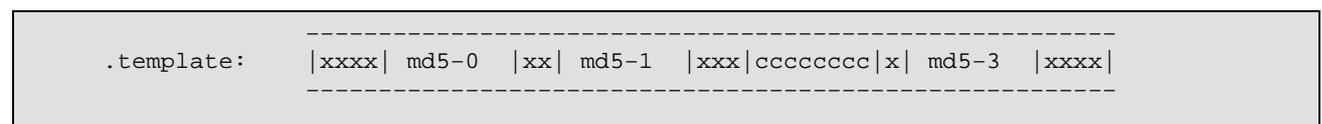
`jigdo-file` takes two things as input: the complete CD image (so the ISO already needs to have been made) and a set of files which may or may not be in the image. Here's a visualization of `jigdo-file`'s input:



Through magic, `jigdo-file` finds out which of the loose files are contained in the ISO image and their offsets within the ISO file. It outputs two files: a ".template" file and a ".jigdo" file.

3.2. The .template File

Given an input of an ISO image and a set of files which may or may not be in the ISO image, `jigdo-file` outputs a .template file for that ISO image. Here's what the .template file looks like:



`jigdo-file` found that the files `file-0`, `file-1` and `file-3` were contained in the ISO image. It removed the contents of these files and replaced them with each file's md5 checksum (the `md5-0`, `md5-1`, etc).

The "x" data (directory information, zero padding, etc) within the ISO image is compressed and written to the .template file. Finally, any files within the ISO image that weren't supplied as loose files (like `file-2`) are also compressed and written to the .template file. This is shown as "c" data in the .template file visualization.

Loose files which were supplied to `jigdo-file` that aren't found in the ISO image (like `file-4`) are ignored.

3.3. The .jigdo File

Given an input of an ISO image and a set of loose files which may or may not be in the ISO image, `jigdo-file` outputs a .jigdo file for that ISO image. The Debian .jigdo files are gzipped, so you need to use `zcat` or `zless` to view them. Here's what a .jigdo file looks like when you gunzip it:

```
md5-0=http://somemirror.org/file-0
md5-1=http://somemirror.org/file-1
md5-2=http://somemirror.org/file-2
md5-3=http://somemirror.org/file-3
```

The .jigdo file simply provides a mapping between the md5sum of a file within the ISO image and the download URL of that file. There are some other things within the .jigdo file, and if you look through it, you'll see the .jigdo file has the same format as a ".ini" file. It should be self explanatory, but if you want the nitty-gritty details, see the `jigdo` documentation.

The format shown above is not quite what you'd see in a typical .jigdo file, but it's very similar. If you look at the [Servers] section at the bottom of the .jigdo file, you'll see exactly what the difference is between what I showed above and an actual .jigdo file.

3.4. Downloading The Image

Once you use `jigdo-file` to generate a .jigdo and .template file for an ISO image, anyone can use `jigdo-lite` to download that image. `jigdo-lite` downloads all the files of a Debian ISO using `wget`, assembles them and forms a copy of the original ISO image on the fly.

4. Downloading Your First Image (In 5 Easy Steps)

We'll assume that you're starting from scratch and don't have any Debian ISOs on hand. Once you burn your set of ISOs, you can use `jigdo-lite` later to update them. We'll cover updating your ISOs in the next section.

4.1. Install Jigdo

First install the `jigdo-file` package:

```
# apt-get install jigdo-file
```

`Jigdo` is under aggressive development. Bug fixes and improvements are constant, so if you're using stable or testing, download `jigdo-file` from unstable at <http://packages.debian.org/unstable/utils/jigdo-file.html>. As of 19 July 2002 it's at version 0.6.8. This is the version used for the examples of this HOWTO.

Note to Woody users: The version of `jigdo-lite` which comes with Woody (rev 0) is not capable of downloading Sarge or Sid. See [Section 6.11](#).

4.2. Download The .template And .jigdo Files

For each ISO image you want to download, you'll need two files:

- The `.jigdo` file for the disk image you want to download.
- The `.template` file for the disk image you want to download.

Example: Woody has 8 images, so you need to download 8 `.jigdo` files and 8 `.template` files. They can be downloaded from <http://www.debian.org/CD/jigdo-cd/> and are named `woody-i386-1.iso.jigdo`, `woody-i386-1.iso.template`, . . . , and `woody-i386-8.iso.template`.

Alternatively, instead of downloading a `.jigdo` and `.template` file for each ISO image, you can give `jigdo-lite` an URL that points to the `.jigdo` file that you need, like `http://a.mirror/woody-i386-1.jigdo`. `jigdo-lite` will download the relevant `.jigdo` and `.template` files on the fly. However, if you're a bit nervous about using `jigdo-lite` for the first time, just download the files and I'll walk you through the process.

4.3. Run jigdo-lite

Run `jigdo-lite` and give it the `.jigdo` file of the image you want to download. Using Woody as an example:

```
$ jigdo-lite woody-i386-1.jigdo
```

Alternatively, if you're working with an URL to the `.jigdo` file rather than a copy of the `.jigdo` file on your hard drive, run `jigdo-lite` with an argument of the URL:

```
$ jigdo-lite http://a.mirror/woody-i386-1.jigdo
```

Debian Jigdo mini-HOWTO

You'll see something like:

```
-----  
Jigsaw Download "lite"  
Copyright 2001-2002 by Richard Atterer <jigdo@atterer.net>  
Getting mirror information from /etc/apt/sources.list  
-----  
Images offered by `woody-i386-1.iso.jigdo':  
  1: woody-i386-1.iso  
-----  
If you already have a previous version of the CD you are  
downloading, jigdo can re-use files on the old CD that are also  
present on the new image, and you do not need to download them  
again. Mount the old CD ROM and enter the path it is mounted under  
(e.g. `/mnt/cdrom'). Alternatively, just press enter if you want  
to start the download of any remaining files.  
Files to scan:
```

If you forget to pass `jigdo-lite` a `.jigdo` file, it will prompt you for one (or an URL that points to one).

If you suspended `jigdo-lite` with `cntrl-z` (don't do this; I'll tell you what you'd see) and did an `ls`, you'd find a new file in the directory named `woody-i386-1.iso.jigdo.unpacked`. This file is simply a gunzip'ed version of the `.jigdo` file.

Right now, `jigdo-lite` is telling us that if we did have an outdated version of the CD, to give the pathname to the CD. Since we're assuming that you're starting from scratch and have no Debian ISOs yet, we have nothing to scan (we'll cover this in [Section 5](#)). So just press `<ENTER>`.

4.4. Specify A Mirror

You'll see:

```
-----  
The jigdo file refers to files stored on Debian mirrors. Please  
choose a Debian mirror as follows: Either enter a complete URL  
pointing to a mirror (in the form  
`ftp://ftp.debian.org/debian/'), or enter any regular expression  
for searching through the list of mirrors (try a two-letter  
country code such as `de', or a country name like `United  
States', or a server name like `sunsite'):  
Debian mirror [http://linux.csua.berkeley.edu/debian//]:
```

`jigdo-lite` is smart enough to use the mirror that you use for your Debian updates by pulling it from `/etc/apt/sources.list` (you can see that I use `linux.csua.berkeley.edu` for my updates). If you wanted to use a different mirror, you would specify a different mirror here. If this is the mirror you want to use, press `<ENTER>`. `Jigdo-lite` will then write a `.jigdo-lite` file in your home directory.

Next, if the `.jigdo` file you're using references a package which needs to be downloaded from a Non-US server (software encumbered by US export restrictions), `jigdo-lite` will prompt you for a Non-US server. The message displayed (and your response) will be very similar to the mirror dialog described in the previous paragraph. The only difference is that you need to specify (or accept the default value for) a Non-US server.

Debian Jigdo mini-HOWTO

If the ISO image you're about to download contains Non-US software, you'll see:

```
-----  
The jigdo file also refers to the Non-US section of the Debian  
archive. Please repeat the mirror selection for Non-US. Do not  
simply copy the URL you entered above; this does not work because  
the path on the servers differs!  
Debian non-US mirror [http://linux.csua.berkeley.edu/debian-non-US//]:
```

and jigdo-lite will write your choice to your `$HOME/.jigdo-lite` file. However, if the image you're about to download doesn't contain Non-US software you won't see this dialog.

Note that you can reset the default mirrors that jigdo uses for your future downloads in `$HOME/.jigdo-lite` file with the following lines:

```
debianMirror='http://some-mirror-to-use/debian/'  
nonusMirror='http://some-other-mirror/debian-non-US/'
```

4.5. Downloading Of The ISO

After specifying the mirror(s), jigdo-lite will look for the `.template` file. If it can't find one, it'll download a `.template` file from a mirror. After it finds the `.template` file (or after it downloads one), you'll see:

```
-----  
Merging parts from `file:' URIs, if any...  
Found 0 of the 1224 files required by the template  
Will not create image or temporary file - try again with different input files  
--13:38:08-- http://linux.csua.berkeley.edu/debian/pool/main/b/bind9/  
lwresd_9.2.1-2.woody.1_i386.deb => `lwresd_9.2.1-2.woody.1_i386.deb'  
Resolving linux.csua.berkeley.edu... done.  
Connecting to linux.csua.berkeley.edu[128.32.247.238]:80... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 157,318 [application/x-debian-package]  
  
30% [=====> ] 47,418 113.22K/s ETA 00:00  
...
```

After the `.template` file was found (or after it was downloaded), jigdo-lite begins pulling packages onto your hard drive. There will be a lot of messages flying across your screen. If this is confusing to you, see [Section 6.12](#). While jigdo-lite is downloading the packages, switch to another console (or open another xterm) and do an `ls` in the directory you're running jigdo-lite in. Now there should be 6 files in the directory:

- `debian-30r0-i386-binary-1.iso.list`
- `debian-30r0-i386-binary-1.iso.tmp`
- `jigdo-file-cache.db`
- `tmp/`
- `woody-i386-1.jigdo`
- `woody-i386-1.jigdo.unpacked`
- `woody-i386-1.template`

Debian Jigdo mini-HOWTO

`woody-i386-1.iso.tmp` won't appear right away. It's a temporary version of the ISO file that gets written to every so often.

`jigdo-file-cache.db` is a Berkeley DB file containing md5sums of any files read in when you specify a directory at the `Files to scan:` prompt. It's more fully described in [Section 7.3](#).

`tmp/` is a directory containing Debian package files that get downloaded. For instance, as I write this, it contains:

```
$ ls tmp/
alsa-headers-0.5_0.5.12a-2_all.deb   tkdiff_3.08-3_all.deb
alsa-utils-0.4_0.4.1-9.1_i386.deb   xfonts-intl-chinese-big_1.2-2.1_all.deb
gnuserv_3.12.4-3_i386.deb           xmanpages-ja_4.1.0.20011224-1_all.deb
pilot-link_0.9.5.0-8_i386.deb       xscreensaver_3.34-3_i386.deb
smpeg-plaympeg_0.4.4-8_i386.deb
```

Every so often, the directory gets flushed and the files get added to `woody-i386-1.iso.tmp`.

At this point, go play some Quake III because this will take some time (you may want to play on a different machine because jigdo is very disk intensive when it assembles the ISO file). At some point, the download will finish and you'll be staring at:

```
FINISHED --13:32:58--
Downloaded: 7,469,872 bytes in 9 files
Found 9 of the 9 files required by the template
Successfully created `woody-i386-3.raw'

-----
Finished!
The fact that you got this far is a strong indication that `woody-i386-3.raw'
was generated correctly. I will perform an additional, final check,
which you can interrupt safely with Ctrl-C if you do not want to wait.

OK: Checksums match, image is good!
$
```

5. Updating Your Image

Presumably, you've read the last section, followed the instructions, burned your newly created ISO files onto CD and are feeling warm and fuzzy. Sooner or later, some packages will get updated and now you want to donate your old CDs to some newbie at your local LUG's installfest and burn yourself a set of updated CDs. Since you're well on the way to becoming a jigdo-guru, we won't go into as much painful detail as we did in the last section.

The first step is to download the .jigdo and .template files, again, for the images you want to update. You may wonder why you need to download them a second time. The reason is because the updated image you want to download has changed. Files may have been added or deleted, but even if not, any updated packages or files will have a different checksum from the checksum listed in the .jigdo and .template files you used when you first downloaded the images.

At this point, you're either holding an outdated Debian CD in your hand or you have the CD's outdated ISO image on your hard drive. Let's go through the steps of getting an updated ISO file. If you have a CD, put it in your CD drive and mount it:

```
$ mount /cdrom
```

On the other hand, if you have an ISO file you'd like to update, mount it as a loop device (you may need to be root to do this). I'll be updating my Woody image, since I noticed that Woody just got a few security updates:

```
# mount -o loop woody-i386-1.iso /mnt
```

Now run jigdo-lite with the .jigdo file as an argument.

```
$ jigdo-lite woody-i386-1.jigdo

-----
Jigsaw Download "lite"
Copyright 2001-2002 by Richard Atterer <jigdo@atterer.net>
Loading settings from `/home/p/.jigdo-lite'

-----
Images offered by `woody-i386-1.jigdo':
  1: Debian GNU/Linux 3.0 r0 Woody - Official i386 Binary-1 CD (debian-30r0-i386-binary-1.i

Further information about `debian-30r0-i386-binary-1.iso':
Generated on Thu, 18 Jul 2002 14:34:12 +0100

-----
If you already have a previous version of the CD you are
downloading, jigdo can re-use files on the old CD that are also
present on the new image, and you do not need to download them
again. You found the secret message; you're a very careful
reader. Mount the old CD ROM and enter the path it is mounted
under (e.g. `/mnt/cdrom'). Alternatively, just press enter if you
want to start the download of any remaining files.

You can also enter a single digit from the list below to
select the respective entry for scanning:
  1: /mnt
```

Debian Jigdo mini-HOWTO

```
Files to scan:
```

jigdo-lite is asking us to give it the location of your mounted CD (if you're updating a CD) or your loop mounted ISO file (if you're using the ISO file). I'm using an ISO file loop mounted on /mnt, so I'll enter /mnt. If you're updating a CD, enter the mount directory of your CD, which is most likely /cdrom. In either case, jigdo-lite will scan the directory of your mounted media, determine which files need updating and re-use the files which don't need updating. You may see something like:

```
Files to scan: /mnt/other

Not downloading .template file - `woody-i386-1.template' already present
jigdo-file: Output file `debian-30r0-i386-binary-1.iso' already exists - delete
it or use --force
jigdo-file failed with code 3 - aborting.
```

What happened? Actually, I wanted to show you this because you'll bump into it sooner or later. I'm updating an ISO file, but the outdated image file is in the same directory I'm working in. Jigdo-lite wants to generate a file called woody-i386-1.iso but there's already a file by that name in the current directory (the outdated image). Jigdo-lite doesn't want to destroy that file, so it bails and lets me know that I can either delete that file or use --force to overwrite the file. You could also rename or move the file too, but I guess jigdo-lite assumes we already know this. :-)

Don't be timid about moving or renaming the image file just because it's loop mounted. The filesystem uses inodes under the hood, and even if you move or rename the file, the inode stays the same. You won't hurt the filesystem mounted under /mnt. As for deleting the ISO file, that won't hurt the mounted filesystem either. A file's inode gets deallocated only when the inode's reference count drops to zero. Mounting the ISO image bumps the reference count up, so the file really gets deleted only after you **rm** the file *and* unmount the loop device. All you people who are updating the CD don't have to worry about any of this. :-)

I'll rename the ISO file to woody-i386-1.iso.old and run jigdo-lite again. Let's try again:

```
$ jigdo-lite woody-i386-1.jigdo

-----
Jigsaw Download "lite"
Copyright 2001-2002 by Richard Atterer <jigdo@atterer.net>
Loading settings from `/home/p/.jigdo-lite'

-----
Images offered by `woody-i386-1.jigdo':
  1: Debian GNU/Linux 3.0 r0 Woody - Official i386 Binary-1 CD
    (debian-30r0-i386-binary-1.iso)

Further information about `debian-30r0-i386-binary-1.iso':
Generated on Thu, 18 Jul 2002 14:34:12 +0100

-----
If you already have a previous version of the image you are
downloading, jigdo can re-use files on the old image that are also
present on the new image, and you do not need to download them
again. Mount the old CD ROM and enter the path it is mounted under
(e.g. `/mnt/cdrom'). Alternatively, just press enter if you want
to start the download of any remaining files.
You can also enter a single digit from the list below to
```

Debian Jigdo mini-HOWTO

```
select the respective entry for scanning:
  1: /mnt
Files to scan: /mnt
Not downloading .template file - `woody-i386-1.template' already present
...
Found 1200 of the 1224 files required by the template
...
```

jigdo-lite remembers that I wanted to scan /mnt and tells me I can either type 1 to scan that directory or type the directory in again. Since I'm a perverse person, I type the name of the directory again.

The ellipsis represent some text that changes rapidly. The first ellipsis is a dynamic list of what files jigdo-lite is scanning. The second ellipses denotes progress in writing woody-i386-1.iso.tmp. Once jigdo-lite finishes scanning the files and writing the temporary ISO file, it prints:

```
Copied input files to temporary file `woody-i386-1.iso.tmp'
- repeat command and supply more files to continue

-----

If you already have a previous version of the image you are
downloading, jigdo can re-use files on the old image that are also
present on the new image, and you do not need to download them
again. Mount the old CD ROM and enter the path it is mounted under
(e.g. `/mnt/cdrom'). Alternatively, just press enter if you want
to start the download of any remaining files.
You can also enter a single digit from the list below to
select the respective entry for scanning:
  1: /mnt
Files to scan:
```

Since you normally don't have another source of files to scan other than your loop mounted ISO file (or your CD), press <ENTER>. Jigdo-lite will then ask you about which mirrors you want to use, just like it did when you downloaded your ISO for the first time. You've already answered these questions before, but if you truly don't remember, you might want to re-read [Section 4.4](#).

At this point, you'll see jigdo-lite working its magic. Now wasn't that easy?

6. Frequently Asked Questions

Questions prepended with a date indicate a time sensitive question, in other words, a question that relates to a temporary situation. If you see one of these questions and know that the temporary situation has changed, please [contact me](#) and let me know so I can remove the question from the mini-HOWTO.

6.1. [11 Aug 2002]: Can I have two jigdo-lite sessions at the same time in the same directory?

Not at the moment. The two sessions will clash over the `tmp/` and `jigdo-file-cache.db` files. This will be worked on. If you want to run concurrent jigdo-lite sessions, use different working directories.

6.2. [11 Aug 2002]: Why aren't the translations of this HOWTO on LDP?

I've been having trouble getting the translations of this HOWTO submitted to the non-English LDP editors.

The German LDP editor, Marco Budde <Budde@tu-harburg.de> refuses to accept the German translation because it was written in Docbook and not Linuxdoc, even though Docbook is the preferred SGML language for the LDP.

The Portuguese LDP editor, Alfredo Carvalho <ajpc@poli.org>, has completely ignored my submission of the Portuguese translation.

If you care about having LDP documents in these languages, I urge you to write to these editors and ask them to please be more responsible about accepting translated documents. For the time being, you can download these translations from my personal website, <http://www.dirac.org/linux/debian/jigdo>.

6.3. jigdo takes a bit long to download the files because wget keeps disconnecting and then reconnecting to the FTP server for each file. Is there a way to make it faster?

The download speed can be increased by using an HTTP instead of an FTP server – FTP is not a very efficient protocol for downloading lots of small files. Additionally, you may want to upgrade to the latest version of wget, because that version supports persistent HTTP connections, which results in another slight speed increase.

Unfortunately, even with persistent HTTP connections, the download speed will not be as high as that of a single-file ISO download. Such speeds can only be achieved with HTTP pipelining – the jigdo GUI application will support pipelining.

6.4. How can I make jigdo use a proxy?

This is described on <http://debian.org/CD/jigdo-cd/> as well as the README of the jigdo-lite tarball.

6.5. What do I do if my jigdo download gets interrupted?

If your download gets interrupted, all you need to do is restart jigdo-lite and hit <ENTER> at all the question prompts. Jigdo-lite will pick up where it left off.

6.6. My jigdo download won't complete because the .jigdo file is broken. When I download a new, fixed .jigdo file, do I need to download all the data over again?

You may find that the .jigdo file you downloaded is broken. It's very uncommon, but it does happen from time to time with moving targets like Debian testing or unstable.

If you find that your .jigdo file is broken, you'll need to download a new .jigdo file (when a fixed one becomes available), but you *won't* need to download all the ISO data again.

You can use the same loop mounting trick we use when updating an ISO image. The difference is that there's no finished .iso file to start with, but the .iso.tmp file is an ISO image too and can be used to finish the download without having to re-download all the data that was downloaded before the broken .jigdo file caused jigdo-file to halt. Simply loop mount the .iso.tmp file on /mnt and when you re-run jigdo-lite with the fixed .jigdo file, tell jigdo-lite to scan /mnt. Don't forget to rename or move the .iso.tmp file so it doesn't interfere with jigdo-lite which will want to create a new .iso.tmp file.

6.7. Can I use jigdo to download images for DVD?

Absolutely; the process is identical to downloading CD images. The only thing you need to do differently is to download the .jigdo and .template files for DVDs instead of CDs. You can find the DVD .jigdo and .template files at <http://www.debian.org/CD/jigdo-cd/>.

Note that you need Linux 2.4 or later to create DVD-sized files. Under Windows, DVD-sized images can't be created at all at the moment because the C++ library of the mingw gcc port doesn't have big file support yet.

6.8. Can I burn the .iso.tmp file to CD?

We haven't tried yet, but it should be possible. You'd probably find some files are filled with "0"s. If someone tries it, please contact me at <p@dirac.org> and let me know what happened.

But more importantly, why would you WANT to do this? :-)

6.9. Why doesn't jigdo work? It downloads some packages and deletes them. I know it doesn't write them to the `iso.tmp` file because the file size doesn't change!

Jigdo works just fine – the `iso.tmp` file is created at the beginning with its final size, but filled with zero bytes. Later, parts of it are overwritten with the downloaded data.

You can tell that jigdo is making progress by looking at the messages "Found X of the Y files required by the template" that are printed from time to time. The second value "Y" should decrease. When it reaches zero, the download is finished.

6.10. I'm having trouble getting jigdo–easy to work.

See [Section 7.1](#).

6.11. I'm having trouble getting jigdo to download Sarge or Sid.

If you're using Potato or Woody, upgrade to jigdo–lite 0.6.8. Because of a change in Jigdo, the version of jigdo–lite that comes with Woody (rev 0) cannot download Sarge and Sid images. The version that comes with Sarge is sufficient ([Section 7.4](#)).

If you're using Sarge or Sid, then you may need some help. Search the archives of the `debian-cd` mailing list, and if that doesn't solve your problem, you should send them a request for help ([Section 7.4](#)).

6.12. Jigdo–lite is too verbose. How can I suppress some or all of its messages?

Jigdo–lite uses `wget`, and `wget`'s output can be quite verbose. If this is unsettling, you can make `wget` more quiet by adding `--non-verbose` to the `wgetOpts` switch in your `~/ .jigdo-lite` file. If you want `wget` to print no messages at all, use `--quiet` in the `wgetOpts` switch.

6.13. Can I use jigdo on platforms other than Linux?

Certainly. If you're interested in Potato or Woody under Microsoft Windows, old SunOS, HP–UX and IRIX you can use jigdo–easy. See [Section 7.1](#) and [Section 7.4](#).

If you want to download Potato, Woody, Sarge or Sid under Microsoft Windows, jigdo–lite has been ported to that platform and can be downloaded from the main jigdo site ([Section 7.4](#)).

7. Errata

7.1. jigdo-easy

Jigdo-easy, by Anne Bezemer, is a fork from jigdo-lite which is portable to a wider range of systems, including Microsoft Windows, old SunOS, HP-UX and IRIX). It's also easier to use than jigdo-lite but because of changes made to Jigdo, will only work with Potato and Woody. Jigdo-easy will not be able to download Sarge and Sid. See [Section 7.4](#) and [Section 6.13](#).

7.2. GUI Interface

A GTK+ interface to jigdo is currently being worked on. It's not fully functional yet, but will be available at some point. There will be a Linux as well as a Windows GUI client.

7.3. jigdo-file-cache.db

The cache contains the md5sums of files read when you supply a directory at the `Files to scan:` prompt. If you have jigdo-file scan the same directory a second time, the scan will be very fast.

This could be useful in the following case: rev0 gets updated to rev1. With the rev1 CD images, some packages may have been pushed from CD n to CD $n+1$, or vice versa. If you had a particularly slow link (e.g. modem), you'd try to avoid downloading these packages again. For this reason, when downloading the new version of CD n , you'd let jigdo-lite scan the three CDs $n-1$, n and $n+1$ (or even all 8 CDs if you want to be 100% sure).

If you have jigdo-lite scan the same CDs over and over again while updating each of the 8 CD images, the cache will prevent all the data on the CDs from being read multiple times.

The cache is much more important when *generating* jigdo files, because you don't want jigdo-file to read in your whole 50GB Debian mirror for every generated jigdo file.

7.4. Resources

This HOWTO is winding down to a close, but I thought I'd leave you with a few links and references to learn more about the jigdo tools and how they work.

<http://atterer.net/jigdo>

This is the jigdo home site. You should definitely browse this site; lots of information about ports, GUI clients and everything under the sun relating to jigdo.

<http://cdimage.debian.org/~costar/jigdo>

The Debian page for jigdo-easy ([Section 7.1](#)).

<http://www.debian.org/CD/jigdo-cd>

The main Debian page for jigdo.

<http://packages.debian.org/testing/utils/jigdo-file.html>

If you're using Potato or Woody, please upgrade jigdo-file to version 0.6.8, which comes with Sarge ([Section 6.11](#)).

<http://lists.debian.org/search.html>

Debian Jigdo mini-HOWTO

You can use this page to search the debian-cd mailing list archives.

<http://www.debian.org/MailingLists/subscribe>

The subscription page for the debian-cd mailing list.