Debian Jigdo mini−HOWTO

Peter Jay Salzman

<p@dirac.orgZZZ>

2005−12−05 ver 1.8

Abstract

Getting Debian ISOs has always been a painful, slow and supremely inefficient process. Jigdo is a tool for distributing and 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.

Table of Contents

Table of Contents

1. Administrata

1.1. Authorship and Copyright

If you want to create a derivative work or publish this HOWTO for commercial purposes, I'd appreciate it if you 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

I would like to thank the author of jigdo, Richard Atterer, simply for writing jigdo. Anyone who has obtained Debian ISOs by other means will know why. 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, Marcelo Ramos, Yufeng Wang, Tsukasa Yamaguchi, Yuri Kozlov, and Oguz Yarimtepe for translating this mini−HOWTO into languages other than English. I feel totally honored that they have found my words worthy of their time and effort. Thanks, guys!

Lastly, I'd like to thank Mark van Lent, Gordon Huff, David Anselmi, Thierry Cabuzel, Russell L. Harris, and Jens Seidel for kind words and corrections.

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 dissertation and the book I'm writing on debugging code with GDB/DDD for No Starch Press, I'll do my best to respond to each email I receive about this mini−HOWTO. News flash: I've completed my Ph.D.; now I'm swamped with job hunting. Does anyone need to hire a theoretical physicist?

1.4. Latest Version And Translations

German:
: Conrad Wood `<cnw@conradwood.netZZZ>`.

Portuguese
: Elcio Mello.

Spanish
: Marcelo Ramos `<mramos@montevideo.com.uyZZZ>`.

Chinese

Yufeng Wang

Japanese
: Tsukasa Yamaguchi. Available at http://www.linux.or.jp/JF/JFdocs/Debian−Jigdo.

Russian
: Yuri Kozlov `<kozlov.y@gmail.comZZZ>`. Available at http://alioth.debian.org/project/showfiles.php?group_id=30279.

Turkish
: Oguz Yarimtepe `<oguzy@comu.edu.trZZZ>`. Available at http://docs.comu.edu.tr/howto/debian−jigdo.html.

In addition to the URLs given above, all the translations (as well as the English version) are available at my website: http://www.dirac.org/linux/debian/jigdo. If you'd like to translate this mini−HOWTO to another language, please contact me at `<p@dirac.orgZZZ>`.

The English version of this HOWTO can also be found at The Linux Documentation Project: http://tldp.org/docs.html.

2. Why jigdo?

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

If you want a set of Debian CDs there are many ways of getting them. One way is to buy them from vendors who sell Debian CDs. This definitely has 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 newer and better way of obtaining Debian ISO images, using a tool called jigdo. In fact, the PIK is now officially dead and all further references to it have been removed from this document. 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 did). 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. 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 official method of downloading ISO images.

A common misconception is that jigdo creates ISO images; it doesn't. Let's discuss the overall process of how jigdo allows you to obtain an ISO image. Let Adam (a Debian release manager) be the person offering the ISO image. Let Betty (a Debian user) be the person who wants to download the ISO image.

1. Adam first creates an ISO image suitable for burning a CD. He might use a utility like mkisofs or debian−cd to create the ISO image. He also creates two small files associated with his newly created image: a `.jigdo` file and a `.template` file. He makes these two files available for download to anyone who wants to obtain his ISO image.

2. Betty then downloads the `.jigdo` and `.template` files. She uses jigdo−lite along with these two files to download Adam's ISO image.
3. When Debian gets updated, Adam creates a new version of the ISO and generates new `.jigdo` and `.template` files.
4. When Betty wants to update her CDs, she downloads the new `.jigdo` and `.template` files and uses them with jigdo−light to update her copy of the ISO images. The important thing here is that she only downloads the differences between her old ISO and Adam's new ISO. She does not have to re−download the parts that are unchanged.

Jigdo comes with two utilities: jigdo−file (used by Adam) which creates the `.jigdo` and `.template` files, and jigdo−lite (used by Betty) which uses these two files to download or update the ISO. If all you want to do is obtain/update Debian ISOs, you'll only use jigdo−lite. You can forget that jigdo−file even exists. :−)

Jigdo addresses all the problems with the other 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 uses wget which, by default, uses http to transfer files. Unlike rsync, http is never blocked by firewalls (except the ones behind which you shouldn't be using jigdo to begin with).
- Jigdo is very kind to the bandwidth of the servers offering the Debian images. The Debian mirrors can handle a bigger load of people using jigdo to download Debian images than with other methods of getting them.

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 download Debian ISOs, but it may help demystify how jigdo works. If you're not interested in the details, simply fast forward to Section 4, "How Do I Use Jigdo".

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:

```
                 --------------------------------------------------------
    ISO Image:   |xxxx| file-0 |xx| file-1 |xxx| file-2 |x| file-3 |xxxx|
                 --------------------------------------------------------
```

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:

```
                 --------------------------------------------------------
    ISO Image:   |xxxx| file-0 |xx| file-1 |xxx| file-2 |x| file-3 |xxxx|
                 --------------------------------------------------------

                      ----------   ----------               ----------    ----------
    Loose Files:      | file-0 |   | file-1 |               | file-3 |    | file-4 |
                      ----------   ----------               ----------    ----------
```

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:

```
                 --------------------------------------------------------
    .template:   |xxxx| md5-0  |xx| md5-1  |xxx|cccccccc|x| md5-3  |xxxx|
                 --------------------------------------------------------
```

jigdo−file found that the files `file-0`, `file-1` and `file-3` were contained in the ISO image. It removed the contents of the 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 enhancements 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 28 Nov 2005 it's at version 0.7.2−2.

4.2. Download The .template And .jigdo Files

For each ISO image you want to download, you'll need both the .jigdo and .template file for that image. Both files follow the same naming convention:

```
distro-arch-n.jigdo
distro-arch-n.template
```

where distro is the name of the distro (like "sarge"), arch is the architecture (like "i386") and n is the disk number (like "1").

For example, sarge 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/. The first .jigdo and .template file are named `sarge-i386-1.jigdo` and `sarge-i386-1.template` respectively.

4.3. Run jigdo−lite

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

```
lucifer$ ls
sarge-i386-1.jigdo  sarge-i386-1.template
lucifer$ jigdo-lite sarge-i386-1.jigdo

Jigsaw Download "lite"
Copyright 2001-2003 by Richard Atterer <jigdo@atterer.net>
Getting mirror information from /etc/apt/sources.list

-----------------------------------------------------------------
Images offered by `sarge-i386-1.jigdo':
  1: 'Debian GNU/Linux testing "Sarge"
         - Official Snapshot i386 Binary-1 CD' (sarge-i386-1.iso)

Further information about `sarge-i386-1.iso':
Generated on Fri, 7 Feb 2003 20:31:28 -0700

-----------------------------------------------------------------
```

```
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 in 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 downloading
the remaining files.
Files to scan:
```

If you suspended jigdo−lite with **control+z** (don't do this; I'll tell you what you'd see) and looked at the output of **ls**, you'd find a new file in the directory named `sarge-i386-1.jigdo.unpacked`. It turns out that .jigdo files are gzip'ed. This file is simply a gunzip'ed version of the `.jigdo` file.

Right now, jigdo−lite is telling us that if we have an outdated version of first CD of sarge, we should give the pathname to the CD. This is how you update your ISO images (or complete your incomplete downloads). 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**.

See also Section 7.2, "More About Scan Sources".

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/]:
```

By default, jigdo−lite pulls the mirror from your `/etc/apt/sources.list`. If you want 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, jigdo−lite will prompt you for a Debian Non−US mirror. The message displayed (and your response) will be very similar to the mirror dialog in the previous paragraph.

```
-----------------------------------------------------------------
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//]:
```

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

If you want to change the default mirrors you use with jigdo at any time in the future, you can modify these two lines in `~/.jigdo-lite`:

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

4.5. Downloading Of The ISO

After you specify the mirror(s), jigdo−lite will begin downloading files to assemble the ISO image:

```
Not downloading .template file - `sarge-i386-1.template' already present

-----------------------------------------------------------------
Merging parts from `file:' URIs, if any...
Found 0 of the 826 files required by the template
Will not create image or temporary file - try again with different input files
--09:35:12--  http://mirror/debian/pool/main/p/pack/pack_3.10-1_i386.deb
    => `sarge-i386-1.iso.tmpdir/mirror/debian/pool/main/p/pack/pack_3.10-1_i386.deb
Resolving linux.csua.berkeley.edu... done.
Connecting to linux.csua.berkeley.edu[128.32.112.231]:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1,911,624 [application/x-debian-package]

19% [======>                              ] 378,304      149.87K/s    ETA 00:09
```

There'll be a lot of messages flying across your screen; if this is confusing to you, see Section 6.13. 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:

- `sarge−i386−1.iso.list`
- `sarge−i386−1.iso.tmp`
- `jigdo−file−cache.db`
- `sarge−i386−1.iso.tmpdir/`
- `sarge−i386−1.jigdo`
- `sarge−i386−1.jigdo.unpacked`
- `sarge−i386−1.template`

The `sarge−i386−1.iso.tmpdir/` directory contains all the Debian packages that jigdo−lite downloads. Every so often, the directory gets flushed and the files get written to `sarge−i386−1.iso.tmp`, which is an temporarily incomplete version of the ISO image you want. Note that `sarge−i386−1.iso.tmp` won't appear until the first time `sarge−i386−1.iso.tmpdir/` gets flushed.

`jigdo−file−cache.db` is a Berekeley DB file containing md5sums of any files read in when you specify a directory at the `Files to scan:` prompt. It's described in Section 7.3.

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 flushes files to the `.iso.tmp` 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 `sarge-i386-3.raw'

-----------------------------------------------------------------
Finished!
```

```
The fact that you got this far is a strong indication that `sarge-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). Using Woody as an example:

```
# 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.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 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
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. See also Section 7.2, "More About Scan Sources".

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* umount 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
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 (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. Why does jidgo ask *twice* for scanning for existing files? Is it enough to say yes once ?

It keeps asking this as long as you enter a path to scan. The idea is that you may want to scan several old CDs, so you can insert one after the other into the drive and keep supplying the path "`D:\`" (or whatever). See also Section 7.2, "More About Scan Sources".

6.2. Jigdo Has Problems Downloading Certain Filenames.

When downloading Debian images under Windows, jigdo–lite may appear to have trouble downloading one or more of the following files:

```
      libbusiness-onlinepayment-bankofamerica-perl_xxx_all.deb
      libbusiness-onlinepayment-authorizenet-perl_xxx_all.deb
      libbusiness-onlinepayment-payconnect-perl_xxx_all.deb
      libmasonx-request-withapachesession-perl_xxx_all.deb
      libtemplate-plugin-calendar-simple-perl_xxx_all.deb
```

Move the jigdo download directory up by as many directories as possible, closer to the drives's root directory.

The NTFS filesystem has a 255 character limit on a file's pathname. When jigdo–lite downloads files from the internet, it makes a copy of the server directory structure in its download directory. With their very long names, the above Debian packages may exceed the allowed path length, which leads to error messages like "`Cannot write to `[very long pathname]' (No such file or directory)`".

Some people may now wonder: Why does jigdo–lite use wget's "`--force-directories`" switch, which creates these problematic directory hierarchies?

Early versions of jigdo–lite did not use it, but then some folks requested that jigdo–lite always use the "`--continue`" switch to avoid half–downloaded .deb files being ignored and deleted when you interrupt and restart jigdo–lite.

Unfortunately, it turned out that this led to problems: The Debian servers contained several identically named files (e.g. "`root.bin`") in different directories, and if you interrupted jigdo–lite at roughly the right time during the download, the chances were high that the resumed download would append data to the wrong half–downloaded file, corrupting it and making the entire jigdo download fail.

6.3. How do I make jigdo use my proxy?

Edit `~/.jigdo-lite` (or `jigdo-lite-settings.txt` for the Microsoft Windows version) into a text editor and find the line that starts with "`wgetOpts`". The following switches can be added to that line:

```
-e ftp_proxy=http://LOCAL-PROXY:PORT/
-e http_proxy=http://LOCAL-PROXY:PORT/
--proxy-user=USER
--proxy-passwd=PASSWORD
```

Of course, substitute the correct values for your proxy server. The last two options are only necessary if your proxy uses password authentication. The switches need to be added to the end of the wgetOpts line before the final `'` character. All options must be on one line.

Alternatively, under Linux you can also set up the `ftp_proxy` and `http_proxy` environment variables, for example in the file `/etc/environment` or `~/.bashrc`.

6.4. Jigdo−lite fails with an error − have I downloaded all those MBs in vain?

If jigdo−file aborts after downloading a considerable chunk of the ISO contents, you'll have a large "`.iso.tmp`" file. There are several things to try to salvage your download:

- Restart the download by pressing **RETURN**. Maybe some of the files could not be downloaded because of timeouts or other transient errors. Try to download the missing files again.
- Try a different mirror. Some Debian mirrors are slightly out of sync −− maybe a different mirror still holds files that were deleted from the one you specified, or it has already been updated with files that are not yet present on your mirror. This has happened quite a few times with me.
- Retrieve the missing parts of the image using rsync. First, you need to find out the correct rsync URL of the image you are downloading: Choose a server that offers rsync access to the stable or testing images, then determine the correct path and filename. Directory listings can be obtained with commands like **rsync rsync://cdimage.debian.org/debian−cd/**.

  Next, remove the "`.tmp`" extension from jigdo−lite's temporary file by renaming it, and pass both the remote URL and the local filename to rsync: **rsync rsync://server.org/path/binary−i386−1.iso binary−i386−1.iso** You may want to use rsync's `--verbose` and `--progress` switches to get status messages, and `--block-size=8192` to increase its speed.
- Under Linux, you can loop−mount the `.tmp` file to access the packages that were already downloaded, and reuse them for generating an image from a newer .jigdo file. To do this, first issue the following commands as root in the directory with the broken download: **mkdir mnt; mount −t iso9660 −o loop *.tmp mnt**. Next, start a new download in a different directory, and enter the path of the mnt directory at the "Files to scan" prompt.

  Under Microsoft Windows you can do the same thing by loop mounting the temporary ISO image using "virtual drive" software. Daemon tools and Nero Image Drive are both very popular. See also http://tinyurl.com/c39zr for more options.

6.5. [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. It's a shame that we have people within the open source community who would sabotage our community from the inside.

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.

Shame on you, Marco Budde `<Budde@tu-harburg.de>`.

Shame on you, Alfredo Carvalho `<ajpc@poli.org>`.

6.6. 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.

6.7. 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 uncommon, but it does happen from time to time with moving targets like Debian testing or unstable.

If you find that `.jigdo` 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−lite 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.

6.8. 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/.

On Linux, you need kernel 2.4 or later to create DVD−sized files.

Under MS Windows, you need to use `jigdo-win-0.7.1a` (released 21 July 2004) or later to create DVD−sized images. This is because of a bug in the large file support of Mingw32, the compiler used to create the MS Windows executables. The bug got fixed on this date, and `jigdo-win-0.7.1a` was released.

6.9. Can I burn the `.iso.tmp` file to CD?

Thanks to Gordon Huff and David Anselmi, we now know the answer is "yes you can". But more importantly, Gordon gave a good reason why you'd want to do this in the first place. Paraphrasing Gordon:

> My friend's Win98 has a *nice* cable connection. I arrive in the morning, start jigdo (more than one, actually) and then we go to the store, tie back the kiwi plant, put up the Christmas lights and Christmas tree, trim the tree, order and split a pizza and fire up the satellite dish.
>
> I leave my friends place with several iso.tmp's on CDRWs. When I get home, I use the iso's that didn't finish to update my jigdo setup at home which is a dial−up.

6.10. Jigdo−lite is broken! It downloads 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 first value "`X`" should increase. When `X` equals `Y`, the download is finished.

6.11. I'm having trouble getting jigdo−easy to work.

See Section 7.1.

6.12. For image updates, I want jigdo−lite to scan 14 loop−mounted images in one go. How can I do this?

When updating CD images, it's tiresome to keep loop−mounting and unmounting images. However, by default the Linux kernel only supports eight loop devices, and jigdo−lite's menu of previously entered paths only has five entries.

To scan many loop−mounted images, you must first tell the Linux kernel to support more than the default eight devices. This is done by giving the "`max_loop`" parameter to the module when loading it, e.g. with "`modprobe loop max_loop=16`" on the command line or by adding the line "`options loop max_loop=16`" to `/etc/modules.conf`. In Debian, you must put this line into a file named e.g. `/etc/modutils/local-loop` and then run **update−modules** because direct changes to `/etc/modules.conf` will be overwritten.

Having mounted the individual images, you can pass the parent directory of their mount points to jigdo−lite for scanning. For example, if the images are mounted under `/mnt/myloopmounts/image1/` etc., pass "`/mnt/myloopmounts`" as the path to scan. If passing the parent directory is inconvenient, you can also create a directory and fill it with symlinks to the mount points.

6.13. Jigdo−lite is too verbose. How can I supress 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.

6.14. 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).

6.15. On MS Windows, why do I get a "`No such file or directory`" error message?

You might find that under MS Windows, jigdo−lite will download some files but then fail to read their contents, which will produce a "`No such file or directory`" error message.

It seems that this occurs if the length of the filenames that jigdo processes exceeds a certain limit. The solution is to move the half−finished download up in the directory hierarchy, closer to the top−level directory of the drive.

6.16. On MS Windows, why won't my image grow larger than 2GB?

You're using an old version of jigdo. Please upgrade to `jigdo-win-0.7.1a` or newer. See Section 6.8.

6.17. On MS Windows, `jigdo-lite.bat` fails with an error message saying "sh" was not found.

This means that the `PATH` command in the `.bat` file failed. For some reason, this is the case if you unpacked jigdo on a Windows network share using a path like "`\\SomeServer\Files\jigdo`". Solution: Use "**Map network drive**" (in the explorer "tools" menu) to assign a drive letter like "`Z:`", then double−click on the `.bat` file inside "`Z:\jigdo`". Alternatively, a workaround is to move everything in the `jigdo-bin` subdirectory up to where the `.bat` file is.

6.18. Can I run multiple instances of jigdo−lite to download images in parallel?

Absolutely. However, to avoid filename clashing, you should run each jigdo−lite instance in its own separate directory. You can start as many instances as you want, go to bed, and when you wake up, all the ISO images will be waiting for you on your hard drive. Be aware that jigdo−lite is bandwidth and CPU intensive, so you won't want to use your computer with multiple instances running in tandem.

6.19. Is there a GUI interface available?

A GTK+ interface to jigdo is being worked on. Both Linux and Microsoft Windows GUI clients are planned. Unfortunately, it's been 80% done for over 1.5 years, so don't hold your breath for its release.

7. Errata

7.1. jigdo−easy

Jigdo−easy, by Anne Bezemer, is a fork of 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 6.14.

7.2. More About Scan Sources

By now you know that when jigdo−lite asks for files to scan, you can use 3 sources:

- A mounted copy of an outdated CD or DVD that you wish to update.
- A loop−mounted copy of an outdated ISO image file on your hard drive.
- A loop−mounted copy of the temporary `.iso.tmp` file, when a previous jigdo−lite run aborted.

As Jens Seidel points out, there is another, rather crafty, source you should use for a scanning source: your apt cache. Apt uses the directory `/var/cache/apt/archives` for cache. There will be many Debian packages sitting in this directory, and they can be used for a scan source for jigdo−lite! So when you're asked for a directory to scan, by all means, use this directory too.

If you're editing the `~/.jigdo-lite` file by hand, be aware that multiple scan directories are space separated, for example:

```
scanMenu='/var/cache/apt/archives/ /cdrom/'
```

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
: The official webpage for the Debian jigdo−file package.

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

https://lists.berlios.de/mailman/listinfo/jigdo−user
: The subscription page for the official Jigdo mailing list.