

MINI-HOWTO

backup and/or restore device or partition using zsplit/unzsplit

Jurij Ivastsuk-Kienbaum
jurij [at] device-image.de

March 14, 2006

Revision History

First draft

This document describes a setup that allows a local backup/restore operation of whole devices or separate partitions using zsplit/unzsplit tools.

Contents

1	Legal Notices	2
1.1	Disclaimer	2
1.2	Copyright	2
1.3	Feedback	2
2	Introduction	2
2.1	Why backup/restore devices and/or partitions?	2
2.2	Why zsplit/unzsplit ?	3
3	Backup device	3
3.1	Preparing target partition	3
3.2	Creating file-system on the target partition	4
3.3	Mounting the target partition	4
3.4	Creating a backup image	5
4	Restore device	6
4.1	Working device is broken, but we have the backup image of it!	6
4.2	Preparing devices	6
4.3	Mounting the source partition	7
4.4	Restoring the device	7

1 Legal Notices

1.1 Disclaimer

This article assumes that the reader using the described setup is familiar with the technical concepts and commands. If you are not comfortable issuing the commands in this document, don't try these instructions.

Additionally, if you do not know the difference between `/dev/hda` and `/dev/hda1`, please do not use this HOWTO. This may result in losing all your data. Please note that I take no responsibility for anything that may go wrong, even if the error resulting from any incorrectness in this article.

1.2 Copyright

Copyright © 2006 DeviceImage-Project.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License (GFDL), Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections, with no Front-Cover Texts, and with no Back-Cover Texts. `zsplit` and `unzsplit` tools are licensed in the terms of the GNU General Public License (GPL).

1.3 Feedback

Comments on this article will be highly appreciated. Direct your comments to: `jurij [at] device-image.de`

2 Introduction

2.1 Why backup/restore devices and/or partitions?

Saving regularly the most important data is a very useful habit. Imagine, you have a data-corruption through virus, direct manipulation or mechanical issues, and you have lost all your data. You were lucky if you would have at least an old backup of your data. In this situation the convention procedure to restore your data is the following: first reinstall your system then you should reinstall all applications you normally use and only then you can restore your data.

Making an image of your hard-disk or of separate partition(s) will essentially simplify all these recover steps to only one step: backup and/or restore the whole device and/or partition. Doing this you need not take care about restoring of your system and applications. If the imaging back up method is used, it saves the entire structure of device and/or partition. After restoring of such device image we get fully functional bootable disk which has an exactly the same state as just before the last backup procedure. It will include your bootable system, your applications and all your data.

2.2 Why zsplit/unzsplit ?

These tools combine features of different tools (dd, gzip and split), they are easier to use in scripts as the combination of different tools. zsplit/unzsplit can deal with Very Large Files. With zsplit/unzsplit you have better software control over the process of compressing and splitting.

3 Backup device

3.1 Preparing target partition

Suppose we have a working 120GiB device /dev/hda which is bootable; from this device we will make a backup image.

We should prepare a partition on the other device which would be a target for this backup and is at least as big as 60-80GiB to contain all 120GiB data of the source device. Although the zsplit tool compresses the image during imaging, we still do not know the end-size of the image file. Therefore it is better to have some spare space for any case.

As target device we have an empty 80GiB device /dev/hdb. First, we should make a partition on this empty device. A handy tool for this is "cfdisk". Boot from the *DeviceImage-LiveCD*, start root console, or in the user console get root writes: su - root. Issue the following command on the prompt:

```
cfdisk /dev/hdb
```

Select "New" option, you will be asked what partition you are going to create (primary or logical). If you would like to create only one partition select "primary" otherwise select "logical".

Next, you will be asked to define a size of partition, I would suggest you use the max. size shown as suggestion. After you set the size you can select "Write" option. Only now all your partition settings will be written to your

disk. Before it you can manipulate settings and see if these are appropriate for you, if not, you can "Quit" and no change will be done on your disk. After this procedure you should have a system-visible partition /dev/hdb1. To check it, issue:

```
fdisk -l
```

3.2 Creating file-system on the target partition

To be able to write to the newly created partition we should create a file-system on it. The tool to achieve it is "mkfs". Suppose we have still *DeviceImage-LiveCD* running, open the root console or get root rights as described above. Check if the newly created target partition is still visible: **fdisk -l** if so, you can create a default file-system. Issue following command:

```
mkfs /dev/hdb1
```

With this command the default linux-filesystem ext2 will be created. If you wish to create another file-system, say ext3 you should use -t option, for example:

```
mkfs -t ext3 /dev/hdb1
```

3.3 Mounting the target partition

File system alone is still not sufficient for writing some files on it. This partition should be mounted to the directory tree. Using mount we connect the newly created partition to the meaningful access point in the directory structure.

Check if you already have a directory /mnt/hdb1 (we still suppose you have a running *DeviceImage-LiveCD* and you are in the root console). Issue the following command:

```
ls -al /mnt
```

if you see something like this:

```
drwxr-xr-x 2 root root 4096 Feb 6 23:41 hdb1
```

the directory we need, is existing and you can skip the next step. If you do not see this directory you should create it:

```
mkdir /mnt/hdb1
```

Now, to connect the target partition to a newly created (or existing) directory (mnt/hdb1), issue the following command:

```
mount -t ext2 /dev/hdb1 /mnt/hdb1
```

If you have created partition with the other file-system (for example ext3) the command for mounting will be the following:

```
mount -t ext3 /dev/hdb1 /mnt/hdb1
```

3.4 Creating a backup image

At this step I would suggest we check again if the source and the target devices for our backup are still the same as we think, because the logical name for recognized device depends on the hardware plug-in place. Issue the command:

```
fdisk -l
```

Check if device names and its sizes are corresponding to physical devices which were defined as source and target. /dev/hda is as big as 120GiB and /dev/hdb is as big as 80GiB.

If it is so, check if our target partition is mounted, issue the following command:

```
mount
```

if you see something like that:

```
/dev/hdb1 on /mnt/hdb1 type ext2
```

it means your target partition is mounted and is ready to be used. Using following command you can also check how many free space you have on this partition:

```
df -H
```

The size of free space should be near 80GiB because the only space we have used till now, was for creating the one partition.

Results of your check should be: /dev/hda is the source device, /dev/hdb1 is visible and is mounted to /mnt/hdb1 and you have free space something near the 80GiB.

The amount of free space is only a guess-estimation. May be your image will need something about 10% of all this free space, may be some more... It depends on the file-system installed onto the /dev/hda and on the compression level of imaging operation. If using the default compression rate 6, approximate estimation for the free space needed for image is the following: if we make an image from MSWindows (NTFS filesystem) you should reserve between 60-80% (of source size) free space on the target partition, for msdos, Fat32, Linux, Unix the reserved free space could be distinctly under 50-40%.

If all seems to be ok, you can start creating of image. As example we will use the name of our image: ***first-backup***. Before we start, I would like to mention: we still have a *DeviceImage-LiveCD* running and you are in the root console. Root rights at this point *are absolutely necessary*.

To create the image, issue the following command:

```
zsplrit -N first-backup -o /mnt/hdb1 -d /dev/hda
```

this command will result in creating one image chunk in the directory /mnt/hdb1

If you would like to monitor the image creating process, you should start another root console cd into /mnt/hdb1 directory and issue the command:

```
tail -f debug.log
```

You will see all debug messages during the image creation and the final success message at the end of image creation. After this success message you can be sure that the program has done its job.

If you would like to create an image split into files of defined size, for example 700MiB for the later saving it onto several CD's, you should issue:

```
zsplrit -s 700M -N first-backup -o /mnt/hdb1 -d /dev/hda
```

4 Restore device

4.1 Working device is broken, but we have the backup image of it!

Suppose we have a following situation: our working device /dev/hda is broken, it does not boot anymore. May be virus or some mechanical issue was a cause for it. Fortunately we have a backup-image of this device on our backup device /dev/hdb (partition /dev/hdb1). So we can in a very short time restore our broken device: system, applications and all data. Your working environment will be in a short time available again.

4.2 Preparing devices

Please plug-in your new /dev/hda device on exactly the same place were the broken device was. Notice, that the size of new device should be exactly the same or bigger then the size of the previous (broken) device.

Please plug-in the backup-containing device on exactly the same place were this device was during the backup procedure.

Boot from *DeviceImage-LiveCD*, and start the root console.

4.3 Mounting the source partition

First we should check if all our devices, target device `/dev/hda` (which should be restored) and source device `/dev/hdb` (which contains the backup image) are recognized and visible. Issue already known command:

fdisk -l

If both devices are visible, check sizes of them to be sure that we have exactly those devices (source and target) as we think.

Mount the `/dev/hdb1` partition to the directory `/mnt/hdb1` as described above (see section 3.3 Mounting the target partition for more information).

4.4 Restoring the device

Please notice, if you have a new device `/dev/hda`, which should be restored, and this device has not been partitioned and formatted yet: You don't need all these preparation steps (partitioning and creating a filesystem). Sufficient is only the size of the target device, it should be the same or bigger as the size of original device from which this backup image has been done. Check if our source partition is mounted (see section 3.4 Creating a backup image for more information). If yes, you can start now with restoring. We suppose that the name of our image is still ***first-backup***.

cd into `/mnt/hdb1`

then issue the following command:

unzsplit -D /dev/hda -d first-backup

To monitor the restore process you can start another root console, cd into `/mnt/hdb1` and issue:

tail -f debug.log

If you issue the above command without `-d` option you will see all messages which are created during the restore procedure on the console, in this case the `debug.log` file will not be created.

The success message indicates that the restoring procedure has been successfully done. As next you can try to boot your restored device.

At this step you should remember the following: if the restored device will not boot, you should check the hardware place where the restored device is plugged in. To be able to boot, the restored device should be plugged in to

the same place as the original (broken) one. If the original device has been known as `/dev/hda` than the restored device, and it *is absolutely necessary*, should be also recognized as `/dev/hda`.