



Lab – MondoRescue

Lab Contents

This lab purpose is to install and use MondoRescue to produce a golden image of a server and restore it.

Lab Writer and Trainer

[Bruno.Cornec@hp.com](mailto: Bruno.Cornec@hp.com)

Table of content

Objectives.....	2
Reference documents.....	3
Environment setup.....	3
MondoRescue installation.....	3
Bare metal Disaster Recovery with Mondo Rescue.....	4
Disaster Recovery (DR) Set Creation.....	4
Scripted Disaster Recovery (DR) Set Creation with a VM.....	14
Bare Metal Recovery on another VM.....	22
Bare Metal Recovery on another system (with iLO Virtual Media).....	23
Bare Metal Recovery in KVM/QEMU (aka P2V).....	23
Network based Disaster Recovery.....	26
Disaster Recovery (DR) Set Creation.....	26
Network based Bare Metal Recovery (with PXE).....	26
SSSTK usage with Mondo Rescue.....	27
Disaster Recovery (DR) Set Creation.....	27
Bare Metal Recovery (with PXE).....	27



Overview of the MondoRescue Lab

Objectives

At the end of the Lab students should be able to install MondoRescue, use mondoarchive to create a golden image of a server and boot from it to restore with mondorestore on a target..

Expected duration : 100 minutes



Reference documents

When dealing with the installation and configuration of MondoRescue, the first approach is to look at some reference documents:

The **HOWTO** at <http://www.mondorescue.org/docs.shtml> as well as **man pages**

The **Wiki** at <http://trac.mondorescue.org/wiki> and especially the **FAQ** at <http://trac.mondorescue.org/wiki/FAQ>

Most of the documentation is provided currently under a wiki format, so contributors are welcome and expected, including those finding issues in that training material. Trac also provides a bug report/enhancement request feature ;-)

A mailing-list dealing with MondoRescue is available at http://sourceforge.net/mailarchive/forum.php?forum_name=mondo-devel

Estimated time for the lab is placed in front of each part.

Environment setup

Estimated time: 10 minutes

MondoRescue installation

Mondorescue is available externally from <ftp://ftp.mondorescue.org/> and internally on the HP LAN from <http://eurolinux.grenoble.hp.com/pub/linux/mondorescue/> or using ftp as well from the same place.

Version 2.2.9.5 is the current stable release. Version 2.2.10 is the development version

Here is the table of corresponding versions:

<i>Tools</i>	<i>MondoRescue 2.2.9.5</i>	<i>MondoRescue 2.2.10</i>
mondo	2.2.9.5	2.2.10.0
mind	2.0.7.6	2.0.8
mind-busybox	1.18.3	-

As the Lab has been prepared on a RHEL6 system, you may want to use yum to do the installation. For that, download the file under `rhel/6/x86_64` named `mondorescue.repo` and place it under your `/etc/yum.repos.d` directory. Adapt to your underlying architecture if needed (point to `i386` instead). Of course, you can use rpm and deal with the dependencies yourself ;-)

```
# cd /etc/yum.repos.d/  
# wget ftp://ftp.mondorescue.org/rhel/6/x86_64/mondorescue.repo
```

Then use `yum` to install the application with all its dependencies and check that the correct version is installed and operational:

```
# yum install mondo  
# man mondoarchive
```



From now on, the Lab is “à la carte” ;-) Which means you can take one of the following part at will and try to do it, once you’ve done at least the Disaster Recovery Set Creation. Estimated time is placed in front of each part.

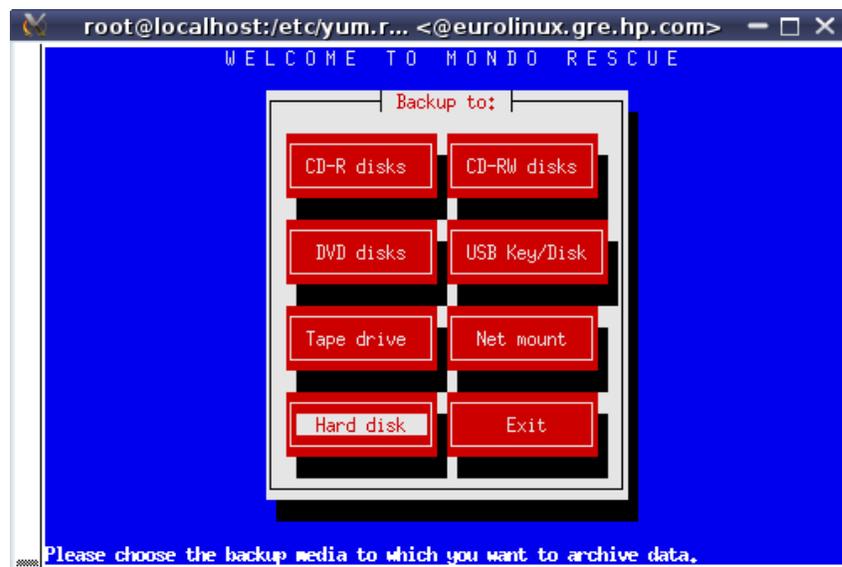
I recommend that if you don't already know mondorescue you do at least the PXE restore as it's the simplest and quickest. The iLO is simple as well but longer and requires adequate hardware. Then look at P2V, SSSTK. Those of you feeling more adventurous, or already knowing the tool, may want to help debug 2.2.10 and use it in their Lab. Bug reports welcome ;-)

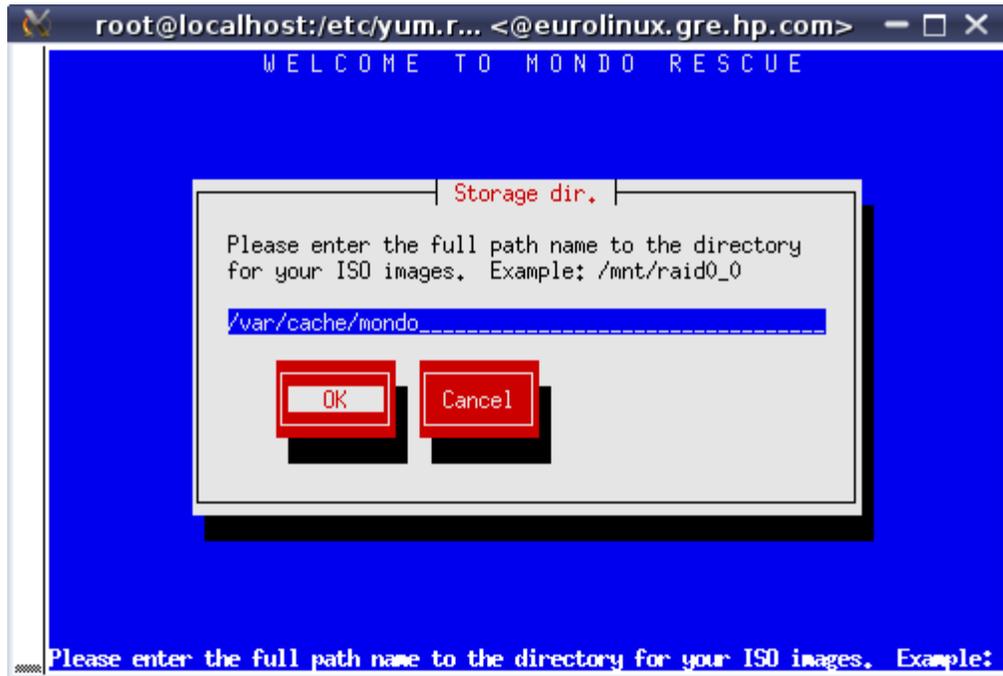
Bare metal Disaster Recovery with Mondo Rescue

Estimated time: 20 minutes – Based on mondo 2.2.9.5, current version.

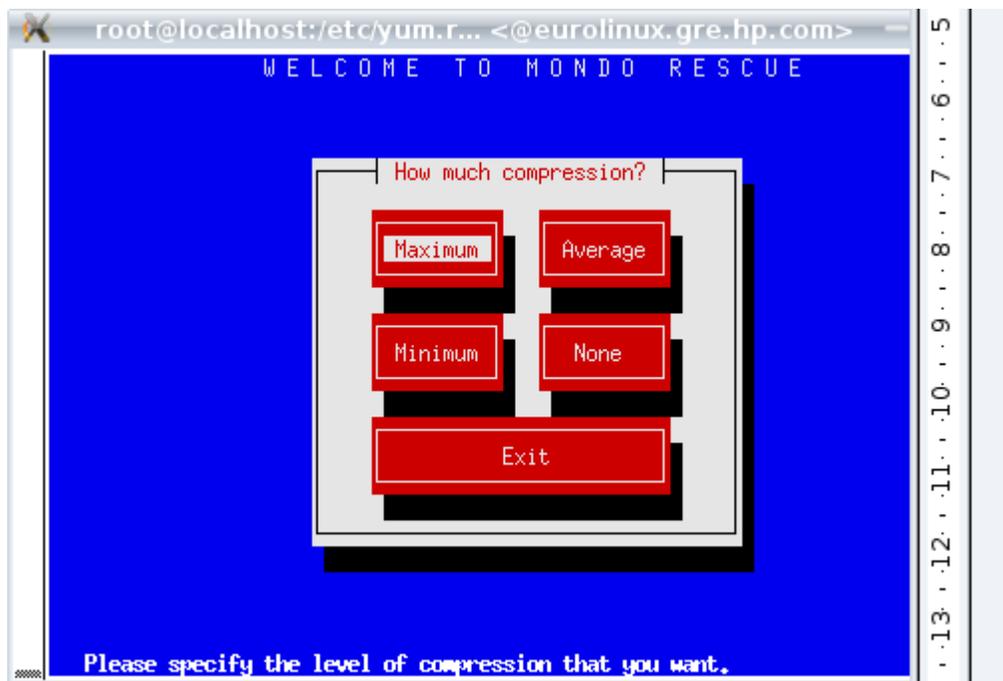
Disaster Recovery (DR) Set Creation

The goal of this part is to make a DR Set of media under the format of ISO images of 4GB of your server. Use for that simply the `mondoarchive` command:





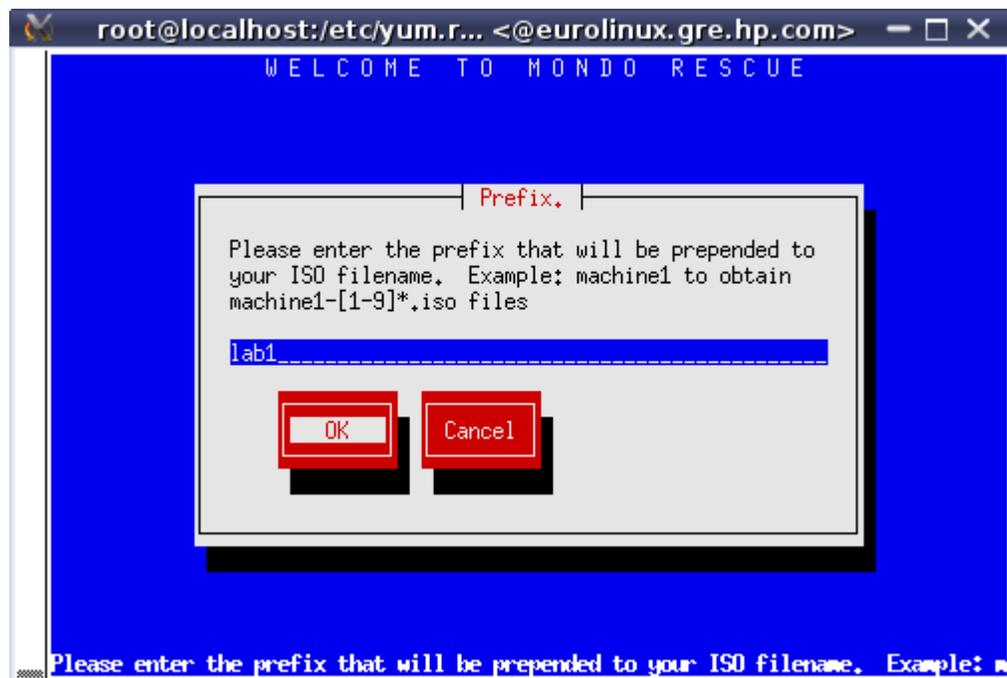
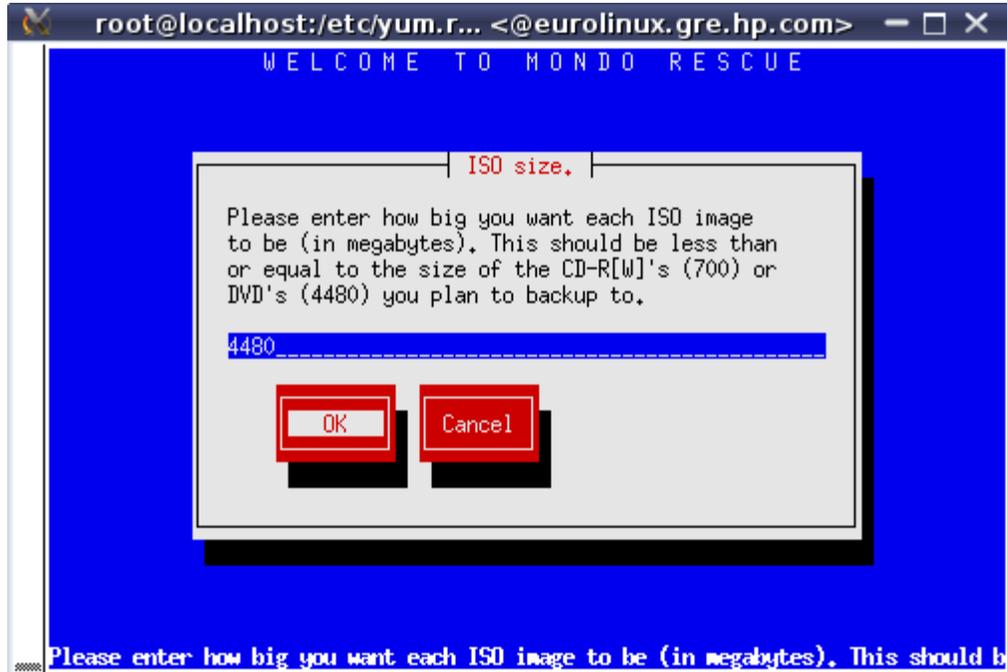
That's where you ISO image will be created at the end. Note that mondo needs room to make that image, which is currently taken were there is the largest place available.



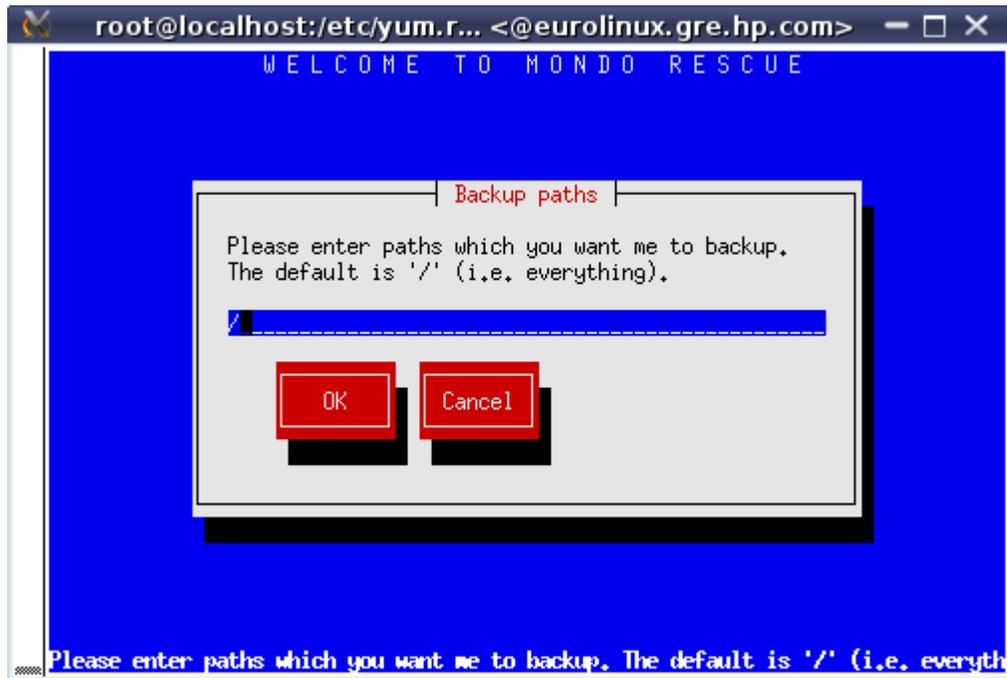
Compression is done by default using bzip2 with that version. Another compressor can only be chosen when using the CLI and options -G (gzip) or -L (lzo) or -Y (lzma) in 2.2.10.



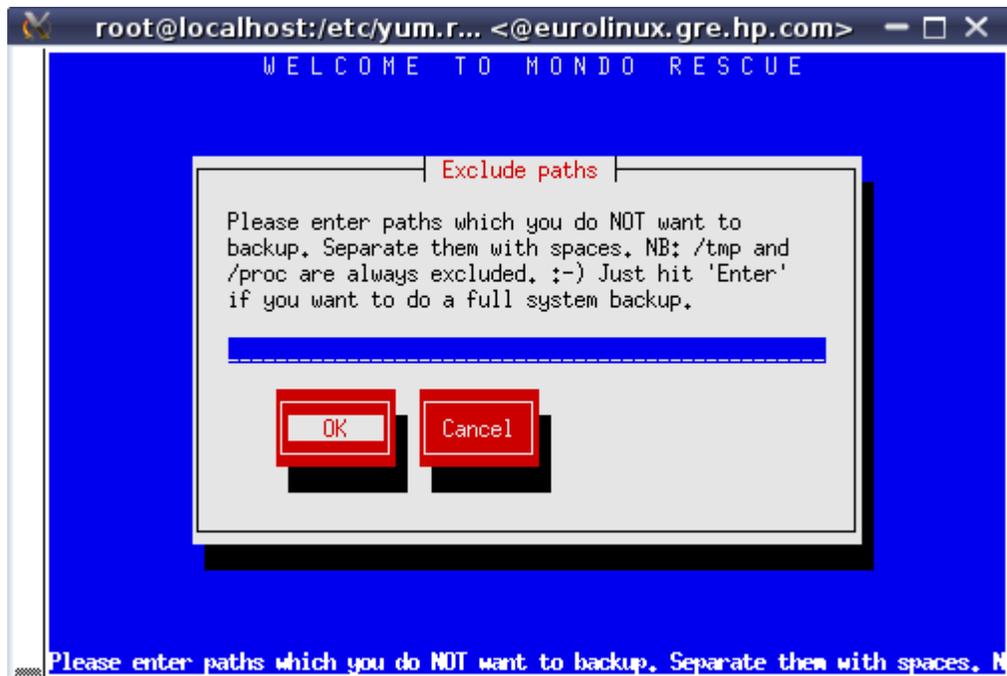
Specify the size of your image (could be larger than a physical media if you do not intend to really burn it).



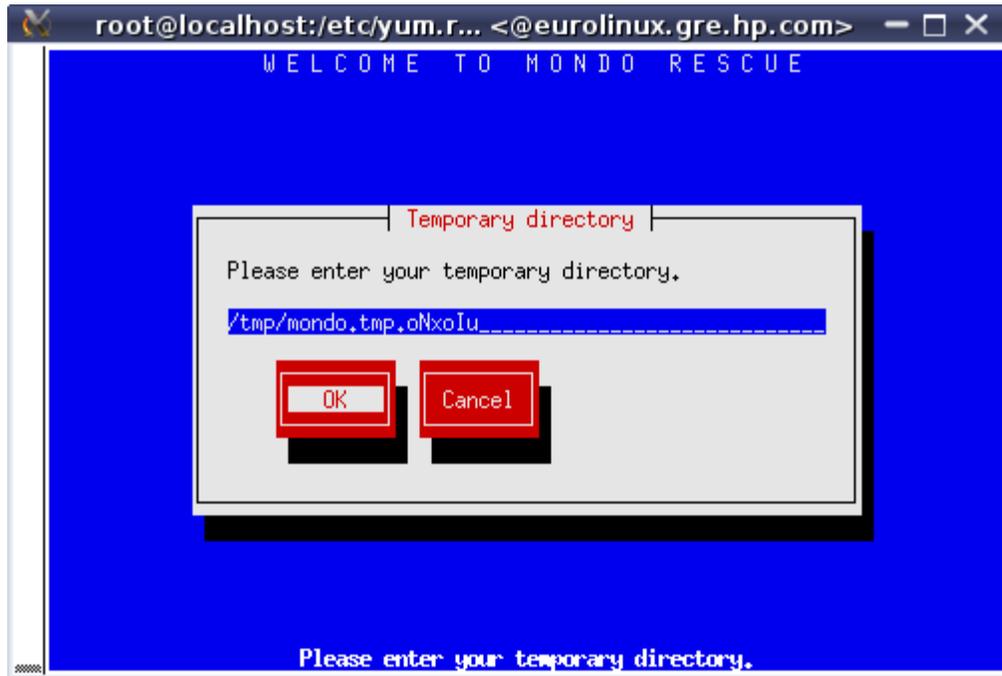
The prefix name allows you to keep multiple images in a central repository for example.



You have to use / for a disaster recovery solution for your include dir.



In order to have a quicker backup, you may want to exclude some useless path (they may be backed up with another tool, which is often the case of data or databases). Advise here is to not exclude what would break the coherency of your package system.



You may now precise where you want your temporary files to be generated (file system having the most place is the default) They are usually not big in size



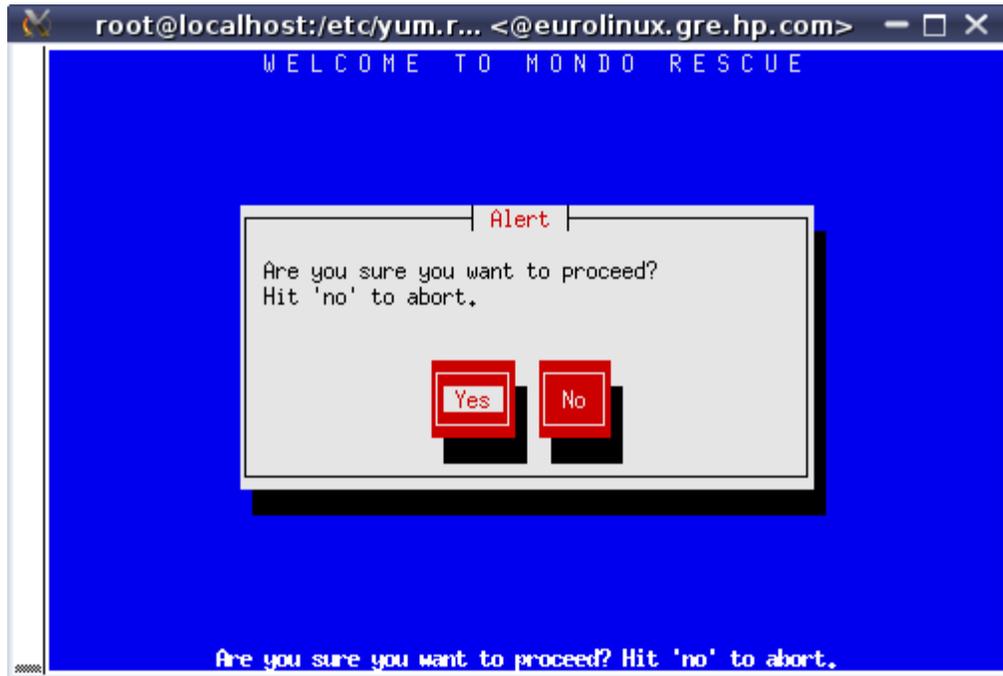
You may now precise where you want your scratch files to be generated (file system having the most place is the default). They will occupy the size of a media so choose with care.



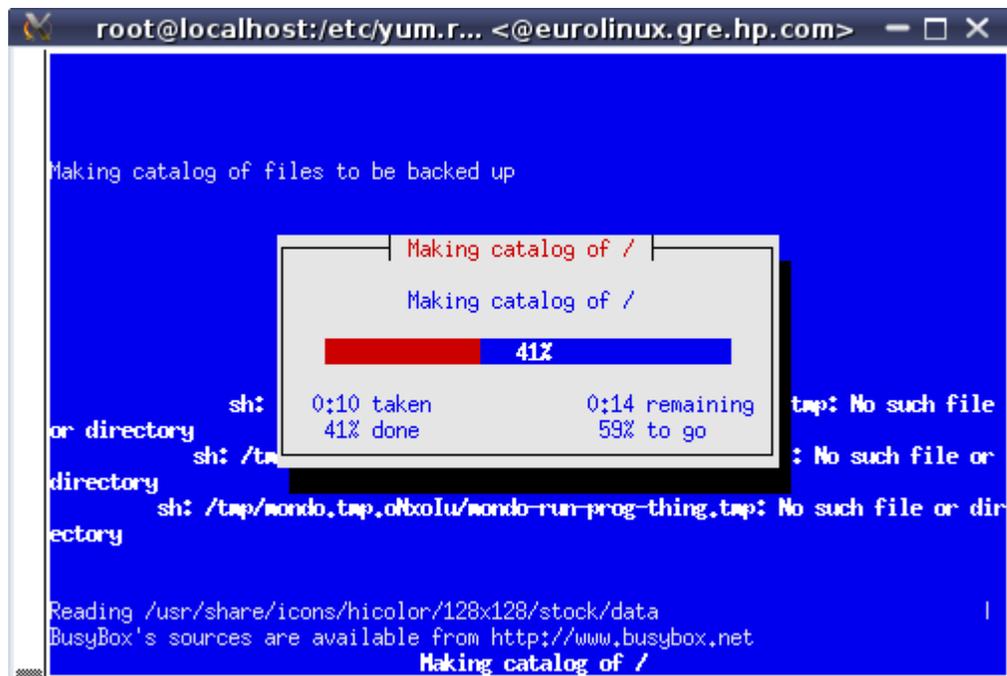
At the end of the archiving run, mondo may optionally verify all the sets created to check their consistency. Of course it takes more time.



With recent versions of mondo you should always say yes here. Even with a Gentoo or old Debian distribution. This message will disappear in 2.2.10.



Fire !



Mondo first creates a catalog of all files to handle.



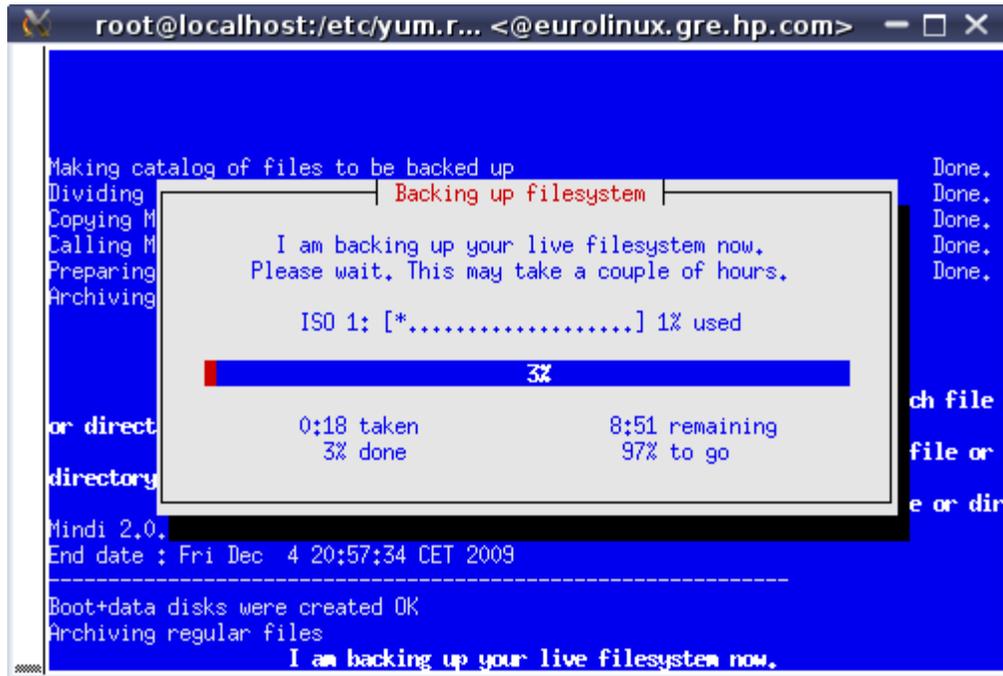
```
root@localhost:/etc/yum.r... <@eurolinux.gre.hp.com>
Making catalog of files to be backed up Done.
Dividing filelist into sets
Dividing filelist into sets
100%
sh: 0:01 taken 0:00 remaining tmp: No such file
or directory sh: /tmp/mondo.tmp.oibxolu/mondo-run-prog-thing.tmp: No such file or
directory sh: /tmp/mondo.tmp.oibxolu/mondo-run-prog-thing.tmp: No such file or dir
ectory
BusyBox's sources are available from http://www.busybox.net
Dividing filelist into sets. Please wait.
Dividing filelist into sets
```

Then it splits those files in various sets to be created.

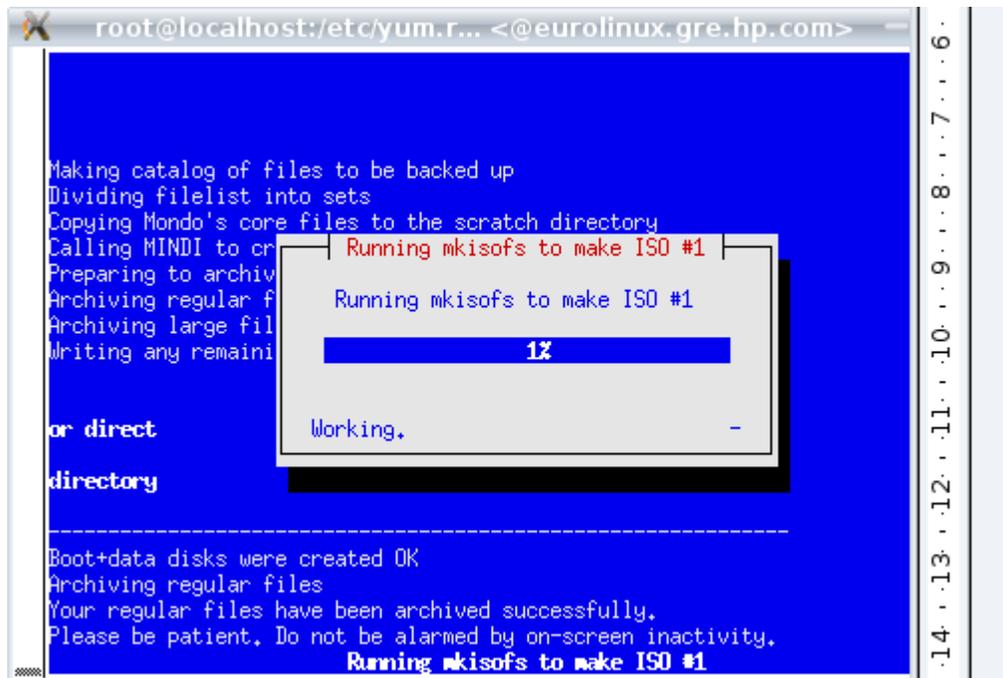
```
root@localhost:/etc/yum.r... <@eurolinux.gre.hp.com>
Making catalog of files to be backed up Done.
Dividing filelist into sets Done.
Copying Mondo's core files to the scratch directory Done.
Calling MINDI to create boot environment
Generating boot+data disks
1%
Working....
kernelpath = /boot/vmlinuz-2.6.18-164.el5
Using modules for kernel: 2.6.18-164.el5
Using modules for kernel: 2.6.18-164.el5
Mindi-BusyBox v1.7.3-r2495 (2009-11-30 23:26:59 CET) multi-call binary
Analyzing dependency requirements^M
Generating boot+data disks
```

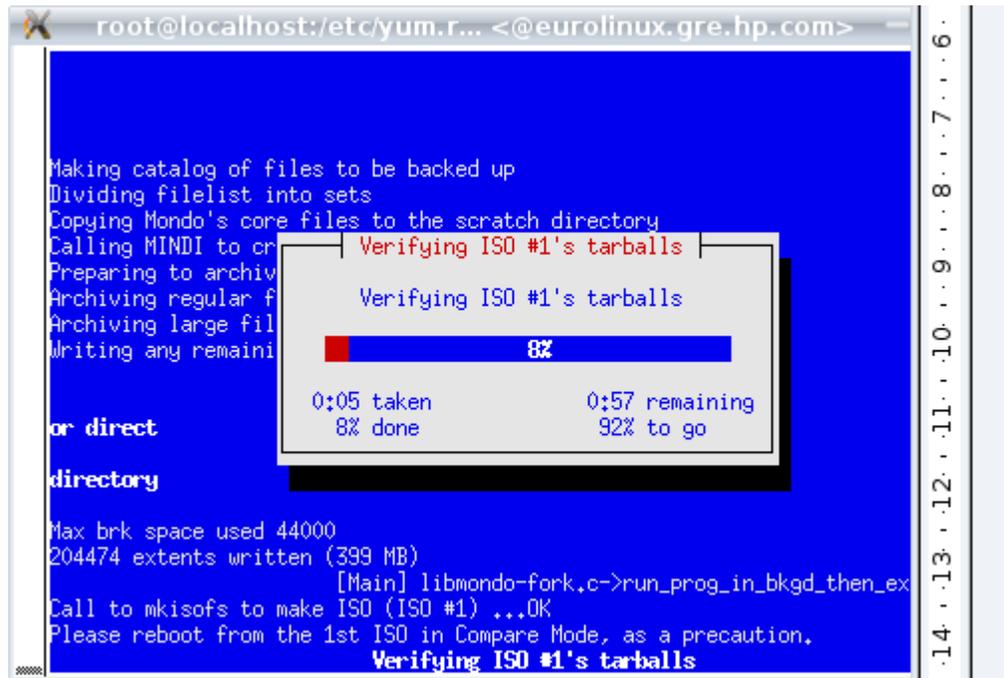
Then mondo calls mindi which generates the boot environment which will be used at restore time to prepare the machine (modules loading, device creation, partitioning, lvm management, ...)

Now it's time to create all the DR sets.

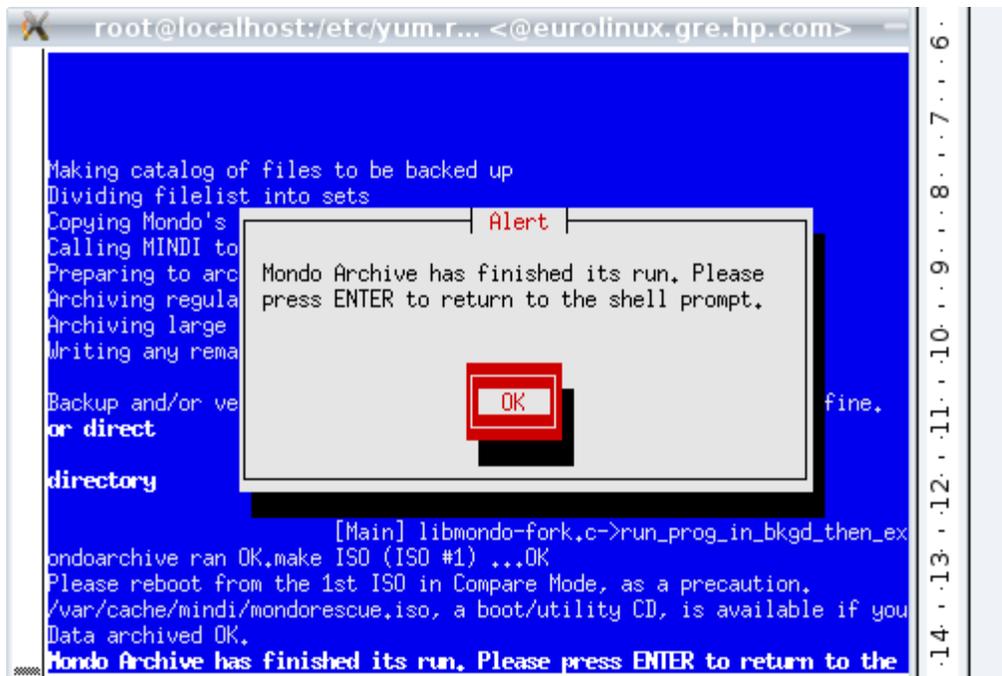


During that phase, you may want to have a look at the documentation on tools provided with this Lab.





The verification of ISO is done upon previous request.



That's all folks !

```
[root@localhost yum.repos.d]# ls -l /var/cache/mondo/  
total 409368
```



```
-rw-r--r-- 1 root root          11 Dec  4 20:55 difflevel.0  
-rw-r--r-- 1 root root 418762752 Dec  4 21:00 lab1-1.iso
```

Scripted Disaster Recovery (DR) Set Creation with a VM

This option is more suited if you want to fully control the tool. In particular, some options are only available using the CLI, such as managing extended attributes e.g.

Here is a scripted example of a backup procedure using a RHEL6 VM (inside the test-mondo-1.qemu file):

```
[hypervisor #] mkdir -p /mnt/mondo  
[hypervisor #] echo "/mnt/mondo *(rw,insecure,no_root_squash)" >>  
/etc/exports  
[hypervisor #] exportfs -a  
[hypervisor u$] kvm -boot c -m 1024 -redir tcp:2222:10.0.2.15:22 test-  
mondo-1.qemu  
[hypervisor u$] ssh -p 2222 root@localhost  
[guest-rhel6 #] cd /etc/yum.repos.d/  
[guest-rhel6 #] wget  
ftp://ftp.mondorescue.org/rhel/6/x86\_64/mondorescue.repo  
[guest-rhel6 #] wget ftp://10.0.2.2/rhel6.repo  
[guest-rhel6 #] cat rhel6.repo  
[RHEL6]  
name=RHEL6 - x86_64  
baseurl=ftp://eurolinux.gre.hp.com/pub/linux/redhat/6/Server  
gpgcheck=0  
enabled=1  
[guest-rhel6 #] yum -y install mondo  
Loaded plugins: rhnplugin  
This system is not registered with RHN.  
RHN support will be disabled.  
Setting up Install Process  
Resolving Dependencies  
--> Running transaction check  
---> Package mondo.x86_64 0:2.2.9.5-1.rhel6 set to be updated  
--> Processing Dependency: syslinux >= 1.52 for package: mondo-2.2.9.5-  
1.rhel6.x86_64
```



```
--> Processing Dependency: mindi >= 2.0.7 for package: mondo-2.2.9.5-1.rhel6.x86_64

--> Processing Dependency: cdrecord for package: mondo-2.2.9.5-1.rhel6.x86_64

--> Processing Dependency: buffer for package: mondo-2.2.9.5-1.rhel6.x86_64

--> Processing Dependency: afio for package: mondo-2.2.9.5-1.rhel6.x86_64

--> Processing Dependency: mkisofs for package: mondo-2.2.9.5-1.rhel6.x86_64

--> Running transaction check

---> Package afio.x86_64 0:2.5-1.rhel6 set to be updated

---> Package buffer.x86_64 0:1.19-4.rhel6 set to be updated

---> Package genisoimage.x86_64 0:1.1.9-11.el6 set to be updated

---> Package mindi.x86_64 0:2.0.7.6-1.rhel6 set to be updated

--> Processing Dependency: mindi-busybox >= 1.7.3 for package: mindi-2.0.7.6-1.rhel6.x86_64

--> Processing Dependency: parted for package: mindi-2.0.7.6-1.rhel6.x86_64

--> Processing Dependency: mtools for package: mindi-2.0.7.6-1.rhel6.x86_64

--> Processing Dependency: dosfstools for package: mindi-2.0.7.6-1.rhel6.x86_64

---> Package syslinux.x86_64 0:3.86-1.1.el6 set to be updated

--> Processing Dependency: perl(Digest::SHA1) for package: syslinux-3.86-1.1.el6.x86_64

--> Processing Dependency: perl(Crypt::PasswdMD5) for package: syslinux-3.86-1.1.el6.x86_64

---> Package wodim.x86_64 0:1.1.9-11.el6 set to be updated

--> Running transaction check

---> Package dosfstools.x86_64 0:3.0.9-3.el6 set to be updated

---> Package mindi-busybox.x86_64 0:1.7.3-2.rhel6 set to be updated

---> Package mtools.x86_64 0:4.0.12-1.el6 set to be updated

---> Package parted.x86_64 0:2.1-10.el6 set to be updated

---> Package perl-Crypt-PasswdMD5.noarch 0:1.3-6.el6 set to be updated

---> Package perl-Digest-SHA1.x86_64 0:2.12-2.el6 set to be updated

--> Finished Dependency Resolution
```



Dependencies Resolved

```
=====
=====
Package                               Arch                               Version
Repository                             Size
=====
=====
```

Installing:

```
  mondo                               x86_64
2.2.9.5-1.rhel6                       mondorescue                       1.3 M
```

Installing for dependencies:

```
  afio                               x86_64
1.rhel6                               mondorescue                       2.5-81 k

  buffer                             x86_64
4.rhel6                               mondorescue                       1.19-24 k

  dosfstools                          x86_64
3.el6                                 RHEL6                             3.0.9-77 k

  genisoimage                         x86_64
11.el6                                RHEL6                             1.1.9-348 k

  mindi                               x86_64
2.0.7.6-1.rhel6                       mondorescue                       276 k

  mindi-busybox                       x86_64
1.rhel6                               mondorescue                       1.18.3-282 k

  mtools                              x86_64
1.el6                                 RHEL6                             4.0.12-194 k

  parted                              x86_64
10.el6                                RHEL6                             2.1-606 k

  perl-Crypt-PasswdMD5                noarch
6.el6                                 RHEL6                             1.3-11 k

  perl-Digest-SHA1                   x86_64
2.el6                                 RHEL6                             2.12-49 k

  syslinux                            x86_64
1.1.el6                                RHEL6                             3.86-928 k

  wodim                               x86_64
11.el6                                RHEL6                             1.1.9-370 k
```



Transaction Summary

=====
=====
Install 13 Package(s)

Upgrade 0 Package(s)

Total download size: 4.5 M

Installed size: 12 M

Downloading Packages:

(1/13): afio-2.5-1.rhel6.x86_64.rpm
| 81 kB 00:00

(2/13): buffer-1.19-4.rhel6.x86_64.rpm
| 24 kB 00:00

(3/13): dosfstools-3.0.9-3.el6.x86_64.rpm
| 77 kB 00:00

(4/13): genisoimage-1.1.9-11.el6.x86_64.rpm
| 348 kB 00:01

(5/13): mindi-2.0.7.6-1.rhel6.x86_64.rpm
| 276 kB 00:00

(6/13): mindi-busybox-1.18.3-1.rhel6.x86_64.rpm
| 282 kB 00:00

(7/13): mondo-2.2.9.5-1.rhel6.x86_64.rpm
| 1.3 MB 00:04

(8/13): mtools-4.0.12-1.el6.x86_64.rpm
| 194 kB 00:00

(9/13): parted-2.1-10.el6.x86_64.rpm
| 606 kB 00:01

(10/13): perl-Crypt-PasswdMD5-1.3-6.el6.noarch.rpm
| 11 kB 00:00

(11/13): perl-Digest-SHA1-2.12-2.el6.x86_64.rpm
| 49 kB 00:00

(12/13): syslinux-3.86-1.1.el6.x86_64.rpm
| 928 kB 00:03

(13/13): wodim-1.1.9-11.el6.x86_64.rpm
370 kB 00:01



```
Total
150 kB/s | 4.5 MB      00:30

Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction

Warning: RPMDB altered outside of yum.

  Installing      : mtools-4.0.12-1.el6.x86_64
1/13

  Installing      : genisoimage-1.1.9-11.el6.x86_64
2/13

  Installing      : afio-2.5-1.rhel6.x86_64
3/13

  Installing      : dosfstools-3.0.9-3.el6.x86_64
4/13

  Installing      : mindi-busybox-1.18.3-1.rhel6.x86_64
5/13

  Installing      : perl-Digest-SHA1-2.12-2.el6.x86_64
6/13

  Installing      : buffer-1.19-4.rhel6.x86_64
7/13

  Installing      : wodim-1.1.9-11.el6.x86_64
8/13

  Installing      : parted-2.1-10.el6.x86_64
9/13

  Installing      : mindi-2.0.7.6-1.rhel6.x86_64
10/13

  Installing      : perl-Crypt-PasswdMD5-1.3-6.el6.noarch
11/13

  Installing      : syslinux-3.86-1.1.el6.x86_64
12/13

  Installing      : mondo-2.2.9.5-1.rhel6.x86_64
13/13

Installed:

  mondo.x86_64 0:2.2.9.5-1.rhel6
```



Dependency Installed:

```
afio.x86_64 0:2.5-1.rhel6          buffer.x86_64 0:1.19-4.rhel6
dosfstools.x86_64 0:3.0.9-3.el6
genisoimage.x86_64 0:1.1.9-11.el6  mindi.x86_64 0:2.0.7.6-1.rhel6
mindi-busybox.x86_64 0:1.18.3-1.rhel6
mttools.x86_64 0:4.0.12-1.el6      parted.x86_64 0:2.1-10.el6
perl-Crypt-PasswdMD5.noarch 0:1.3-6.el6
perl-Digest-SHA1.x86_64 0:2.12-2.el6  syslinux.x86_64 0:3.86-1.1.el6
wodim.x86_64 0:1.1.9-11.el6
```

Complete!

```
[guest-rhel6 #] mount 10.0.2.2:/mnt/mondo /mnt
```

```
[guest-rhel6 #] mondoarchive -On 10.0.2.2:/mnt/mondo -d. -G -N -E
"/var/log|/usr/share/doc|/var/cache|/usr/src|/usr/share/texfm|/usr
/share/games|/usr/share/webmin|/var/tmp|/tmp" -s 4380m -p test-small -z
```

Initializing...

See /var/log/mondoarchive.log for details of backup run.

Checking sanity of your Linux distribution

Disk /dev/dm-0 doesn't contain a valid partition table

Disk /dev/dm-1 doesn't contain a valid partition table

Done.

BusyBox's sources are available from <http://www.busybox.net>

Making catalog of files to be backed up

Done.

Dividing filelist into sets

Dividing filelist into sets. Please wait.

```
---evalcall---1---          Dividing filelist into sets
---evalcall---2--- TASK:  [**.....]    6% done;  0:00 to go
---evalcall---E---
---evalcall---1---          Dividing filelist into sets
---evalcall---2--- TASK:  [**.....]    9% done;  0:00 to go
[...]
---evalcall---1---          Dividing filelist into sets
---evalcall---2--- TASK:  [*****.]    92% done;  0:00 to go
```



```
---evalcall---E---  
Your backup will probably occupy a single nfs. Maybe two.  
Done.  
Copying Mondo's core files to the scratch directory  
Done.  
Calling MINDI to create boot+data disks  
Your boot loader is GRUB and it boots from /dev/sda  
Waiting for 'echo hi > //mondo.tmp.W2jLkE/mojo-jojo.bla.bla ; mindi  
--custom /  
Boot+data disks were created OK  
Done.  
Preparing to archive your data  
Done.  
Archiving regular files to media  
Archiving regular files  
---progress-form---1--- I am backing up your live filesystem now.  
---progress-form---2--- Please wait. This may take a couple of hours.  
---progress-form---3--- Working...  
---progress-form---E---  
---progress-form---4--- TASK: [*.....] 1% done; 0:00 to  
go  
---progress-form---1--- I am backing up your live filesystem now.  
---progress-form---2--- nfs 1: [*.....] 1% used  
---progress-form---3--- Please wait. This may take a couple of hours.  
---progress-form---E---  
---progress-form---4--- TASK: [*.....] 2% done; 6:14 to  
go  
---progress-form---1--- I am backing up your live filesystem now.  
---progress-form---2--- nfs 1: [*.....] 1% used  
---progress-form---3--- Please wait. This may take a couple of hours.  
---progress-form---E---  
---progress-form---4--- TASK: [*.....] 5% done; 3:18 to  
go
```



```
[...]  
---progress-form---4--- TASK: [*****] 97% done; 0:06 to  
go  
---progress-form---1--- I am backing up your live filesystem now.  
---progress-form---2--- nfs 1: [**.....] 7% used  
---progress-form---3--- Please wait. This may take a couple of hours.  
---progress-form---E---  
---progress-form---4--- TASK: [*****] 100% done; 0:00 to  
go  
Your regular files have been archived successfully.  
Done.  
Archiving large files to media  
I am now backing up all large files.  
---progress-form---1--- I am now backing up all large files.  
---progress-form---2--- Please wait. This may take some time.  
---progress-form---3---  
---progress-form---E---  
---progress-form---4--- TASK: [*.....] 1% done; 0:00 to  
go  
---progress-form---1--- I am now backing up all large files.  
---progress-form---2--- nfs 1: [**.....] 7% used  
---progress-form---3--- Please wait. This may take some time.  
---progress-form---E---  
---progress-form---4--- TASK: [*.....] 1% done; 0:00 to  
go  
[...]  
---progress-form---4--- TASK: [*****...] 83% done; 0:00 to  
go  
---progress-form---1--- I am now backing up all large files.  
---progress-form---2--- nfs 1: [**.....] 7% used  
---progress-form---3--- Please wait. This may take some time.  
---progress-form---E---  
---progress-form---4--- TASK: [*****] 100% done; 0:00 to  
go
```



```
Done.
Writing any remaining data to media
Please be patient. Do not be alarmed by on-screen inactivity.
---evalcall---1---      Running mkisofs to make nfs #1
---evalcall---2--- TASK: [***.....] 11% done; 0:16 to go
---evalcall---E---
---evalcall---1---      Running mkisofs to make nfs #1
---evalcall---2--- TASK: [****.....] 19% done; 0:12 to go
[...]
---evalcall---2--- TASK: [*****.....] 84% done; 0:03 to go
---evalcall---E---
Call to mkisofs to make ISO (nfs #1) ...OK
Done.
Done.
Backup and/or verify ran to completion. Everything appears to be fine.
/var/cache/mindi/mondorescue.iso, a boot/utility CD, is available if you
want it
Data archived OK.
Mondoarchive ran OK.
See /var/log/mondoarchive.log for details of backup run.
Execution run ended; result=0
Type 'less /var/log/mondoarchive.log' to see the output log
[guest-rhel6 #] umount /mnt
[guest-rhel6 #] exit
```

(You may need to install the nfs-utils package)

Bare Metal Recovery on another VM

Estimated time: 20 minutes

After creating an empty VM, you'll be able to boot it from your Mondorescue CD and restore in it through NFS.

```
[hypervisor u$] qemu-img create -f qcow2 ~/test-restore.qemu 8G
[hypervisor u$] kvm -boot d -m 1024 -redir tcp:4444:10.0.2.15:22 -cdrom
/mnt/mondo/test-small-1.iso ~/test-restore.qemu
```

After you restored your system, relaunch the VM to check everything is in order:



```
[hypervisor u$] kvm -m 1024 -boot c -redir tcp:4444:10.0.2.15:22 ~/test-restore.qemu
```

Bare Metal Recovery on another system (with iLO Virtual Media)

Estimated time: 20 minutes

In order to have optimal throughput, use a browser on your system to access the OA of the target machine. Mount the produced ISO image with the virtual media of it, and restore your initial content on it.

You should have nothing special to do except accept it takes much more time than using a physical real media due to the USB emulation provided by the iLO 2 virtual media. If you're lucky to use an iLO3 (ProLiant G7 and upper) based system it will be much quicker.

You may want to try to change some parameters to see how mondorestore deals with it. You may remark that LVM modifications are not easy today, and require to edit the `/tmp/i-want-my-lvm` script. This is expected to be improved in version 3.0.

If you have ssh access to the iLO2 you may want to use CLI to manipulate the Mondo CD:

```
[client u$] ssh Administrator@ilo2  
hpiLO-> vm cdrom insert http://machine/path/to/mondo-cd  
hpiLO-> vm cdrom set boot_once  
hpiLO-> vm cdrom get  
hpiLO-> power reset
```

Bare Metal Recovery in KVM/QEMU (aka P2V)

Estimated time: 30 minutes

For this part, in addition to the above mentioned documentation, please refer to the excellent manual provided by Lester Wade at <http://www.mondorescue.org/docs/p2v.pdf>.

Now you can try to see if you can re-install the image of your physical system just like that on a KVM Virtual Machine.

You first need some additional packages. Then you'll need to create an empty virtual machine. To do that derive the following command:

```
[root@localhost ~]# yum install kvm kvm-qemu-img  
[root@localhost ~]# ln -sf /usr/libexec/qemu-kvm /usr/bin/kvm  
[root@localhost ~]# modprobe kvm-intel  
[root@localhost ~]# mkdir -p /home/vm  
[root@localhost ~]# qemu-img create -f qcow2 /home/vm/img.qemu 130G
```

where 130 is the number of GB to allocate for the VM, which should correspond to the original disk size. The space will only increase as the VM is used.

You can now use the MondoRescue image to initiate your virtual machine with KVM. So type a command similar to the following:



```
[root@localhost ~]# kvm -cdrom /var/cache/mondo/lab1-1.iso -m 512 -boot  
d /home/vm/img.qemu
```

It will launch the VM and boot on the CD which is associated with your MondoRescue image, thus beginning the installation. 512 is the size of the RAM in MB, that you may adapt at will.

```
QEMU
These disks were generated on the following system:-
    localhost.localdomain
    CentOS release 5.4 (Final)
    Kernel 2.6.18-164.el5
    on a x86_64 architecture

To format and restore all files automatically, type 'nuke' <enter>.
To restore some/all files interactively, type 'interactive' <enter>.
To compare the archives with your filesystem, type 'compare' <enter>.
To boot to a command-line prompt (expert mode), type 'expert' <enter>.
You may add one or more of the following parameters as well:-

oondonteject - mondorestore will not eject the CD; this is useful if, for
oo            instance, your PC's case has a concealed CD-ROM drive
oonoresize  - your mountlist will not be adjusted to use your unallocated
oo            hard disk space
ootextonly  - do not call any Newt library routines; this is unattractive
oo            but valuable if you find your Newt library has bugs in it

e.g. Type 'nuke donteject textonly' if you have an unstable Newt library and
a PC whose CD-ROM drive tray would be damaged if it unexpectedly ejected.

boot:
```

Choose automatic mode to restore. It should fail.

You'll have to adapt the mountlist as hard disk device names are different (`/dev/sda => /dev/hda`)

Then continue by trying to restore your content using that new mountlist.

However, it will still not work as you'll see, as we were using LVM and that LVM configuration uses the old device names as well, but can not be changed through the interface. So after the error message concerning the formatting of the File Systems which aborts, change terminal (ALT-F2) and do the following:

:

```
# perl -pi -e 's/sda/hda/' /tmp/i-want-my-lvm
# perl -pi -e 's/sda/hda/' /tmp/mountlist.txt
```

Then relaunch `mondorestore` and this time use the interactive mode.

All those tasks have to be done, as we are changing fundamentally the underlying hardware of our system. However, the tool supports it, providing some adaptations. As seen earlier, the DR on the same hardware is strait forward. An enhancement request exists in trac to have a more automatic support of cloning on different hardware.



```
VNC: QEMU <@o185i228.gre.hp.com>
-- MARK --ns were mounted OK.
syslogd: select: Operation not permittedre all of your data?
-- MARK --
syslogd: select: Operation not permitted
-- MARK --
syslogd:
Restoring: +-----+ Restoring from archives +-----+
Restoring:           Restoring data from the archives.
-- MARK --           Please wait. This may take some time.
syslogd:           Restoring from fileset #14 on CDR #1
-- MARK --
syslogd:           16%
-- MARK --
syslogd:           8:31 taken           42:35 remaining
-- MARK --           16% done           84% to go_ -- MARK --
syslogd: select: Operation not permitted
syslogd:
...OKMARK
...OKlogd: select: Operation not permitted
Errors during disk partitioning were handled OK.
Partitions were formatted OK despite those errors.
All partitions were mounted OK.
Restoring from CDR #1
Restoring data from the archives.
```

At the end of the restore process, you'll also have to adapt the disk used for the boot loader. Say yes when asked for a modified mountlist and review the content of the files proposed. Rebuild your initrd as proposed.

After that `mondorestore` will let you at the prompt. Review the log file `/var/log/mondorestore.log`, and check for potential issues at reboot time (especially boot loader).

Then close KVM/QEMU, and relaunch it without booting on the DVD:

```
[root@localhost ~]# kvm -m 512 /home/vm/img.qemu
```



```
VNC: QEMU <@o185i228.gre.hp.com>
root (hd0,0)
Filesystem type is ext2fs, partition type 0x83
kernel /vmlinuz-2.6.18-164.el5 ro root=/dev/vg_root/lv_root rhgb quiet
[Linux-bzImage, setup=0x1e00, size=0x1d5bfc]
initrd /initrd-2.6.18-164.el5.img
[Linux-initrd @ 0x1fcae000, 0x331947 bytes]

Memory for crash kernel (0x0 to 0x0) notwithin permissible range
PCI: PIIX3: Enabling Passive Release on 0000:00:01.0
Red Hat nash version 5.1.19.6 starting
Reading all physical volumes. This may take a while...
Found volume group "vg_root" using metadata type lvm2
3 logical volume(s) in volume group "vg_root" now active
Welcome to CentOS release 5.4 (Final)
Press 'I' to enter interactive startup.
Setting clock (utc): Sat Dec 5 13:51:53 CET 2009 [ OK ]
Starting udev: _

Kernel alive
kernel direct mapping tables up to 100000000 @ 10000-15000
```

Network based Disaster Recovery

Estimated time: 30 minutes

Disaster Recovery (DR) Set Creation

You'll have to use the following script to realize your backup:

```
# mkdir -p /mondo
# cat > /usr/local/bin/postmondo << EOF
perl -pi -e 's/^\HWADDR.*//' /etc/sysconfig/network-scripts/ifcfg-eth*
EOF
# mount -o rsize=32768,wsize=32768 eurolinux.gre.hp.com:/pub3/svg/mondo-
test /mondo
# mkdir -p /mondo/labn
# mondoarchive -O -d /labn -p labn -g -s 4380m -N -n
eurolinux.gre.hp.com:/pub3/svg/mondo-test -E /home/vm
```

Just replace `labn` above by your lab number (1 to 8). And then launch the script which will do the backup through NFS on eurolinux. If you're not on the HP LAN, then the instructor will give you the IP address of the system to use instead.

Network based Bare Metal Recovery (with PXE)

In order to restore your target blade, you'll have to populate the PXE server on eurolinux. Entries like the following have already been declared:

```
label lab2
```



```
kernel mondo/klab2

append initrd=mondo/ilab2 nuke load_ramdisk=1 prompt_ramdisk=0
ramdisk_size=130000 rw root=/dev/ram iso_mode pxe ipconf=eth0:dhcp
prefix=lab2 nfsmount=eurolinux.gre.hp.com:/pub3/svg/mondo-test
ramdisk_blocksize=1024 post=/mnt/RESTORING/usr/local/bin/postmondo
serial=/dev/ttyS1 console=ttyS1
```

What remains to do is to create/update the kernel and initrd from your ISO image in the tftp directory. Just use the Web interface at <http://eurolinux.gre.hp.com/cgi-bin/lab.cgi>

then boot your target blade using F12 to activate PXE boot. At pxelinux boot prompt type `labn` and check the status of your restore.

SSSTK usage with Mondo Rescue

Estimated time: 5 minutes – For information only – no suitable hardware available.

Disaster Recovery (DR) Set Creation

This time, before launching the backup, look at the file `/usr/share/doc/mindi-2.0.7.3/README.bkphw` in order to add SSSTK support to your environment. Do it only for the Smart Array Controller.

Solve the issues linked to that installation ;-) (Hint you may need to use `yum search` and install additional HP Insight Manager drivers. You may even need to launch some tools manually to determine the real issue)

Use `hpacucli` to change the stripe size of your source blade, so that it doesn't match the one on your destination blade.

Also check in `mindilog` file `/var/log/mindi.log`, that the hardware profile has been taken in account.

Bare Metal Recovery (with PXE)

You should be questioned about the possibility to restore the hardware profile of your original Blade, in the early stages of the restore process. Confirm. Then you should have nothing special to do.

During the reboot following the restore, check that the parameter of the blade has been changed back to the value of the one originally on your source blade.

There is much more to discover around MondoRescue, but I leave that for an update of this document.