Reprinted from the

Proceedings of the
Linux Symposium

July 23rd–26th, 2008
Ottawa, Ontario
Canada

Conference Organizers

Andrew J. Hutton, *Steamballoon, Inc., Linux Symposium,*
Thin Lines Mountaineering

C. Craig Ross, *Linux Symposium*

Review Committee

Andrew J. Hutton, *Steamballoon, Inc., Linux Symposium,*
Thin Lines Mountaineering

Dirk Hohndel, *Intel*
Gerrit Huizenga, *IBM*
Dave Jones, *Red Hat, Inc.*
Matthew Wilson, *rPath*
C. Craig Ross, *Linux Symposium*

Proceedings Formatting Team

John W. Lockhart, *Red Hat, Inc.*
Gurhan Ozen, *Red Hat, Inc.*
Eugene Teo, *Red Hat, Inc.*
Kyle McMartin, *Red Hat, Inc.*
Jake Edge, *LWN.net*
Robyn Bergeron
Dave Boutcher, *IBM*
Mats Wichmann, *Intel*

Authors retain copyright to all submitted papers, but have granted unlimited redistribution rights to all as a condition of submission.

MondoRescue: a GPL Disaster Recovery Solution

Bruno Cornec
Project Lead
bruno.cornec@hp.com

Abstract

MondoRescue is a GPL Disaster Recovery solution. It has existed since 2000, and has now matured to a global solution used both to restore systems in case of emergency as well as to deploy dozens of systems having the same or nearly the same configuration. The main Web site is at `http://www.mondorescue.org` where all the detailed information is contained.

This paper will explain the various functions of the solution: backup image creation, (various filesystems supported, compression schemas, media, distributions, . . . mindi, the mini-distro creating the environment needed for restoration), restore options, (fully automated, interactive, compare mode, online, . . .), some extended usages for which the presenter made patches (LVM v2 support, PXE support. . .), creation of an image deployment server suited for PXE, restoration of mondo images, cloning techniques with mondo, and so on.

It will also explain the new orientation given to the project since September 2005 with the new maintainership of B. Cornec and the various aspects considered for 3.0.x.

At last, we will also cover some important points around project management (SVN, ML, trac, . . .).

1 First encounter with MondoRescue

My initial discovery of the tool was back in 2000, when I was in charge of a project in HP for which I had to find a way to provide an easy installation mechanism of a preload of Linux on our servers in EMEA. More over, the idea was to do that with media that the customer would get in the same box as the server.

At that time, only 2 tools existed for this purpose: `mkCDrec` and `MondoRescue`. As I more easily understood the way to use MondoRescue for my project, combined with the fact that the project already covered 90% of my needs, and that the developer (Hugo Rabson) was cooperative and inclined to integrate patches, I then chose it as my tool for this project, and have used it ever since.

Having done a patch to automate the recovery steps to make it easy to both customers and plant people in charge of the installation, I was quickly integrated in the development team of the project (even if not very active). I tried to make an Itanium (ia64) port, but didn't follow up with it at that time, as our HP team wasn't working on the Itanium program anymore. In the other hand, the pre-install Linux project was really successful and gave me a good opportunity to jump on the project.

The goal of MondoRescue is thus to provide a GPL Disaster Recodery (DR) or Cloning technology which is distribution-neutral (it runs on Mandriva, Fedora, OpenSuSE, RHEL, CentOS, SLES, Debian, Ubuntu, Gentoo, Slackware), filesystem-neutral (it supports ext2/3, reiserfs, XFS, JFS, LVM v1 and v2, and software & hardware RAID), media-neutral (CD-R/RW, DVD+-R/RW, tape, network, ISO files), kernel-neutral (provides a failsafe one, or use the one you run or another one, as long as MondoRescue requirements are fulfilled).

The interface is either command-line based (via options) or a semi-graphical interface based on `newt`. A Web interface to the CLI is also planned for version 3.x.

1.1 `mindi`

MondoRescue uses `mindi` to create the bootable part of the DR media set, which uses itself `busybox` to provide a light Linux environment at restore time. `mindi` is a shell script. Its role is to create the necessary boot environment needed at restore time to be able to correctly set up your hardware and have access to your recovery images.

Typically, `mindi` is called through `mondoarchive` to build that small boot environment. It can also be called separately in order to provide a minimal boot image matching your current installation, so you can use it as a rescue disc.

`mindi` creates a bootable image that cat be used to either create recovery USB devices, using `syslinux`, or serve as the boot part of an El-Torito ISO9660 CD, using `isolinux`.

`mindi` has some interesting options to help you analyze the underlying environment and check that it will be handled correctly.

`mindi -findkernel` finds the image of the running kernel in the file system. Example on a Mandriva 2007.1 distribution:

```
# mindi --findkernel
/boot/vmlinuz-2.6.17.14-mm-desktop-5mdv
```

Example on a RHEL 4 distribution:

```
# mindi --findkernel
/boot/vmlinuz-2.6.9-42.ELsmp
```

Another useful option of `mindi` for debuging problems is the `-makemountlist`. Example on a MDV 2008.1 distribution (`mindi` 2.0.2):

```
# mindi --makemountlist /tmp/1
Your mountlist will look like this:
Analyzing LVM...

DEVICE      Mountpoint   Format     Size   Label/UUID
/dev/sda1   /            ext3       1239
/dev/sdb1   /home        ext2     476937   /home
/dev/sda8   /tmp         ext3       1129
/dev/sda6   /usr         ext3      12033
/dev/sda7   /var         ext3       4235
/dev/sda5   swap         swap        980
```

This exhibits some news brought by the more recent version (support of UUID for Ubuntu, for instance). If the output of that command doesn't reflect the layout of partitions on your system, then you will experience problems during normal `mindi`'s run.

Most of hardware support issues met with Mondorescue come in fact from `mindi`. As its responsibility is to create a correct boot environment for the machine, it has to put all the required drivers on the boot part of your rescue media. In order to do that, in current versions, a lot of supported drivers are included in `mindi`'s code for all the required drivers that you may want to use (CD/DVD drivers, IDE/SATA/SCSI controllers, USB controllers, tape controllers, and network controllers for those doing PXE restore). Code example:

```
FLOPPY_MODS="ide-floppy floppy"
TAPE_MODS="ht st osst ide-tape ide_tape"
SCSI_MODS="3w-xxxx 3w_xxxx 3x_9xxx\\
3x-9xxx 53c7,8xx a100u2w a320raid\\
aacraid adpahci advansys aha152x\\
aha1542 aha1740 aic79xx aic79xx_mod\\
aic7xxx aic7xxx_mod aic7xxx_old\\
AM53C974 atp870u BusLogic cciss\\
cpqfc dmx3191d dpt_i2o dtc eata\\
eata_dma eata_pio fdomain gdth\\
g_NCR5380 i2o_block i2o_core\\
ide-scsi ieee1394 imm in2000\\
initio ips iscsi isp megaraid\\
megaraid_mm megaraid_mbox\\
mptbase mptscsih mptsas mptspi\\
mptfc mptscsi mptctl NCR53c406a\\
ncr53c8xx nsp32 pas16 pci2000\\
pci2220i pcmcia ppa psi240\\
qla1280 qla2200 qla2300 qlogicfas\\
qlogicfc qlogicisp raw1394\\
scsi_debug scsi_mod sd_mod seagate\\
sg sim710 sr_mod sym53c416 sym53c8xx\\
sym53c8xx_2 t128 tmscsim u14-34f\\
ultrastor wd7000 vmhgfs"

# ide-probe-mod
IDE_MODS="ide ide-generic ide-detect\\
ide-mod ide-disk ide-cd ide_cd\\
ide-cs ide-core ide_core edd paride\\
ata_generic ata_piix libata\\
via82cxxx generic nvidia ahci\\
sata_nv cmd64x"
PCMCIA_MODS="pcmcia_core ds\\
yenta_socket"
USB_MODS="usb-storage usb-ohci\\
usb-uhci usbcore usb_storage input\\
hid uhci_hcd ehci_hcd uhci-hcd\\
ehci-hcd ohci-hcd ohci_hcd usbkbd\\
usbhid keybdev mousedev"
CDROM_MODS="$TAPE_MODS $FLOPPY_MODS\\
$IDE_MODS af_packet cdrom isocd\\
isofs inflate_fs nls_iso8859-1\\
nls_cp437 sg sr_mod zlib_inflate\\
$USB_MODS $PCMCIA_MODS"
NET_MODS="sunrpc nfs nfs_acl lockd\\
fscache loop mii 3c59x e100\\
bcm5700 bnx2 e1000 eepro100\\
ne2k-pci tg3 pcnet32 8139cp\\
8139too 8390 forcedeth vmxnet vmnet"
EXTRA_MODS="$CDROM_MODS vfat fat\\
loop md-mod linear raid0 raid1 xor\\
raid5 raid456 lvm-mod dm-mod\\
dm-snapshot dm-zero dm-mirror\\
jfs xfs xfs_support pagebuf\\
reiserfs ext2 ext3 minix nfs\\
nfs_acl nfsd lockd sunrpc jbd\\
mbcache"
```

Of course, maintaining that list is an endless story. ;-) That's why an evolution planned for the next stable release will be to automatically include the full list of all drivers to use. Anyway, this is a difficult task when you want to extend Mondorescue in the cloning area. For example, if you want to restore the content of your physical system in a Virtual Machine (VM) for demo purposes, you need to have the drivers for the VM in your image, which may not even be part of your original distribution and that you may want to be able to add. The same is true if you backup on SCSI and restore on SATA controllers...

However, currently you have to deal with a fixed list of drivers, and modify mindi's code by amending those variables in case some driver is missing.

When called from mondoarchive, mindi has a high number of parameters for the moment passed to him with the -custom option, which triggers the fact that it's used in a mondoarchive context. This will change in version 3.0.x as instead a configuration file will be used, generated by mondoarchive to override the defaults held in mindi's configuration file.

The most important information in trying to understand mindi's issues is contained in the log file /var/log/mindi.log. If used alone, you should always attach it to your request on the mailing list (if used from mondoarchive, the tool includes it in its own log file, so you don't need it anymore).

Another tool provided by mindi is the /sbin/init script launched at boot time, whose role is as said to prepare the drivers to support your hardware, but also to configure all the software stacks required to be able to access the DR sets stored on some media somewhere. It goes from just mounting a physical CD-ROM on the right directory to setting up LVM on your hard driver and the network stack, mounting via NFS the remote file system, finding on it the right ISO image to loopback-mount on the same directory.

The steps involved in init include:

- Load the mandatory modules from initrd: InsertEssentialModules
- Get restore media going: HandleTape | HandleCDROM | start-nfs | start-usb
- Get additional stuff from restore media including all other modules: install-additional-tools (gets ordinary NFS working)
- Now that all modules are available, load the ones we need: insert-all-my-modules.

As all those scripts are currently bash scripts run by busybox, one way to debug a problem is to include a set -x command inside them at the interesting line where you want to have a detailed view of actions run by the shell. This is, of course, also true for mindi in itself. Then, once those tasks have been performed, the command mondorestore is launched to deal with the restoration process in itself from the DR set.

1.2 mondoarchive

MondoRescue provides one command, mondoarchive, which handles the creation of the DR set. mondoarchive is written in C. It accepts multiple options to force it to act specifically (and can be launched that way with cron); or without options, you will get a curses-based interface asking you the questions required to make your archive. After having done some sanity checks on your system, mondoarchive will first build a catalog of all the files to archive. Then it will call mindi to create the boot image used at restore time, passing to it all the parameters required with its -custom option. Then mondoarchive will call afio and your compressor of choice to prepare the package of files to back up. When it estimates that it won't be able to add more files in order to fit within the planned capacity, it will call mkisofs on the temporary directory to produce the final ISO, and loop through those steps again up to the last file in the catalog. At the end of that step, it will do the same with the so-called "big files," which are split in order to take the maximum advantage of media size.

It supports multiple compression tools:

- bzip2: the default one historically. Best compression ratio, slower backup/restore time, ideal when space contraints are predominant.
- lzo: added second to improve performance while retaining a good compression ratio. The lzo package has to be added to the distribution as most of them do not include it.

- gzip: added recently in 2.2.1. Offers probably the best balanced solution between time and compression ratio. Standard in all distributions.

MondoRescue is based on the afio tool to create archive packages. The choice of afio may be surprising, but it presents some advantages compare to the more classical tar or cpio:

- Compatible with cpio to allow for easy recovery of content in case of catastrophe where even afio is not available.
- Does compression on a file-by-file basis, instead of doing it on the resulting package. This improves the reliability of the solution as even if a media has an error, the user may still be able to recover all the other files; whereas with a single compressed package, he may lose up to the whole package.
- Doesn't compress already compressed files (based on files' suffixes, whose list can be modified).
- Archives are portable between different types of UNIX systems, as they contain only ASCII-formatted header information.
- Supports various block sizes (useful for tape backups).

With mondoarchive you can generate files (ISO images that you can burn later or keep on an NFS server for further deployment with PXE) or you can directly create media (CD, DVD, tapes, and USB devices) that you can then use to regenerate your system. The size of the image is a parameter the administrator may choose at will (e.g., 700MB (CD-size), 4GB (DVD-size), 50GB through the LAN, etc.); mondoarchive will create the number of media needed automatically and potentially invoke the burning command to make your media.

mondoarchive (and mindi) can use either your running kernel to boot your system at restore time, or whatever kernel you specify to it. Among those you may want to use in some very particular cases is the failsafe kernel that mindi can provide as a replacement.

It has to be noted that mondoarchive is working on a live system, so precautions need to be taken when backing up databases, or evolving sets of data that need to remain consistent (you may want to stop or snapshot your application before launching the archival). A possibility is to run the system in the S run-level to avoid those issues. Also the filesystems are backed up when they are still mounted, so a fsck will have to occur at the first boot after restore.

Here is an example of mondoarchive's CLI usage:

```
/usr/sbin/mondoarchive -OV \\
-k /boot/vmlinuz-2.6.16 -d /bkp -s 700M \\
-E "/usr/share/doc /usr/src /mnt/fs2" \\
-N -p machine1 -T /home/mondo/tmp \\
-S /home/mondo/scratch \\
-n server.hpintelco.org:/writer/nfs
```

- -O: Backup the machine
- -V: Verify the backup
- -k: Do not use the running kernel but a specific one
- -d: Destination path for the images created (ISO files)
- -s: Size of the images created
- -E: List of directories to exclude from backup
- -N: Do not backup network file systems
- -p: Prefix to be use in image name
- -T: Path of the temporary files (useful for NFS mode to have them created locally)
- -S: Path of the scratched files (useful for NFS mode to have them created locally)
- -n: NFS location of the images

Look at the man page of mondoarchive for more details on these options.

The usage of a configuration file in v3.0.x will allow to avoid using all those parameters in the future. That same configuration file will be used as a base to fill all the fields of the Web interface in order to generate the right mondoarchive command.

1.3 mondorestore

MondoRescue provides one command, mondorestore, which handles the restore of the DR set. mondorestore is written in C. It will handle the restore process on your system once /sbin/init has set things up. One option you can pass to it is the -p option to specify the prefix used at backup time, which is only useful when using NFS restore (name doesn't matter for physical media).

mondorestore is generally called by /sbin/init during the boot of your DR media. It can also be called manually in the case you booted in expert mode (in order to setup your hardware before for whatever reason) or in interactive mode from your running system to restore selectively chosen files.

mondorestore will get the parameter passed to /sbin/init from the boot prompt, from the post-init script, which could be -Z compare, -Z nuke or (by default) -Z interactive (see below for details). Without parameter, mondorestore will first ask you whether it should operate in compare, interactive or automatic mode. Then it will ask from which media you want to restore (CD/DVD/Tape/HDD/NFS are among the choices). Then depending on the type of media, it may ask complementary questions before displaying an interface allowing you to change your file system's layout (could be size, type), as long as the mount point remains accessible and of a sufficient size to host the data. When done with that layout, the restore process will start by formatting your hard drive and then restoring your data from the DR media set. At the end it will restore the boot loader (including your modifications if necessary) and relabel if needed.

mondorestore supports multiple ways of doing the restoration:

- From bootable media: CD/DVD, USB devices (keys, disks), and OBDR tapes
- From Virtual Media (HP Proliant only): USB CD emulation
- From the network: PXE.

The content of the archives can then be on the same media, or located on a hard disk on the machine, on a tape, or on a network share (NFS). Even if MondoRescue is primarily a DR solution, some derived usage may be envisioned. ;-) One of such usage is as a backup solution. Thus the mondorestore command can also be launched from the shell and will provide you with a semi-graphical interface to select the files and directories you want to restore.

Different modes are provided at restore time when you boot from your MondoRescue DR set:

- Automatic: In this mode, no questions will be asked of the user, and the restore will be done fully automatically, as much as possible. This is the option to use in another derived usage of MondoRescue, cloning machines.
- Interactive: This is the standard mode. In this mode, you will be asked a certain number of questions (with generally good defaults coming from your original machine) on how you want to restore your system. This mode allows you to make changes compared to the original machine—for instance, to adapt to a new hardware configuration, etc. Thus the type of the filesystem, its size, and layout can be modified, as well as new hardware supported (if the drivers are available). It is indeed possible to restore on a SATA controller when the backup was made on a SCSI controller.
- Compare: This mode is a non-destructive one, allowing you to compare the content of what is on your DR media with what is currently on the system. A report will be generated at the end to give you the list of all differing files.
- Expert: in this mode, mondorestore is not even launched, allowing you to have access to a shell giving you full control before launching it to prepare the hardware the way you want.

Another derived usage is to use MondoRescue as a cloning tool. This generally requires knowledge of both the master and target machine. If those machines are identical, you may use the -H option, which will allow you to use the automatic mode at restore time, making it easy for people in charge of the deployment to indeed restore the system (only one keyword to type at boot time). If those machines are different—especially in terms of drivers—you'll have to inform mindi that it will have to include additional modules in order to support the target platform correctly.

2 Historical view

Coming back to my history with the MondoRescue project, I have to confess that after that initial work, I wasn't really involved around MondoRescue until 2004, when I worked again on my Itanium port to finish it in April. Patches were committed upstream as usual in the CVS repository where I have had write access since June, 2003. I also produced (around the end of 2004) some other patches to improve Proliant support (the `/dev/cciss` device file) but this time was unable to commit them, as we were in a transition phase between CVS and SVN. . . which never really worked (or during a too-short period of time). `:-(` The original author of the project then became less interested in it, and questions appeared on the mailing-list and between the developers on how to proceed, without conclusion at that time.

Then there was the "famous" FastServers affair! (look at Google for "FastServers mondo" if you want the history). Fortunately I also had PXE patches I published under the GPL at the same time, so the story wasn't a very good one indeed. I needed the ability to store a large number of MondoRescue images on a central repository and be able to redeploy them from that same server through the network, without media manipulation. To achieve this, the central server needs to be configured as an NFS server (the only protocol supported for file sharing) and PXE server (using ISC DHCP server, hpa TFTP server, and `pxelinux`), and some support was added in `mindi` to handle the loading of network drivers, IP setup in `busybox`, the NFS stack, and mounting the images from that file share.

But with all that, we were still unable to commit code; patches were floating on the mailing-list without real integration; Hugo published his last version the 3rd of May 2005 (2.04) and then gave nearly no news; developers were discussing from time to time what to do. . . clearly the project was going to die.

And I was a bit upset by that, as I was using it quite a lot in the HP/Intel Solution Center in which I was (and still am) working, I had patches that would improve some cases that I was unable to commit, so was attaching them to mails on the mailing-list, . . . So finally I decided it was time to have a working development environment, and hosted it on `Berlios.de`. I chose them as they were offering subversion access, whereas Sourceforge was still using CVS only (and I was convinced, during the few months of availability, of subversion's advantages over CVS). I proposed to all the people working around MondoRescue to give them access as they were accustomed to, and that's how Andree Leidenfrost, who was already committing bug fixes and was managing Debian packages, was involved in the new development team. Here is the message which announced the "fork" (with original typos `:-)`):

```
From: Bruno Cornec <Bruno.Cornec@hp.com>
Date: Fri, 9 Sep 2005 20:48:30 +0200
Subject: Announce: mondo 2.04_berlios/
mindi 1.04_berlios available (was:
[Mondo-devel] Is Mondo still being worked
on?)

Hello,

First some feedback:

I'm working with mondorescue since 2000 to
integrate some patches I needed at that time.
I've never been a very active developper, but
provided from time to time some patches to
improve things (PXE, cciss support, -H
option, ...)  and was SVN committer.

After monthes of problems to contribute to
the development of mondo (CVS to SVN
migration, then SVN repository not accessible
for me since nearly one year now), I'm now in
the "scratching an itch" phase :-) (Cf:
http://www.catb.org/~esr/writings/
cathedral-bazaar/cathedral-bazaar/ar01s02.
html)

So I decide it was time to do something in
order to avoid the project to decline.  I
like it, I use it a lot, it makes my life
easier, both at work and at home.  So here
are the news:

I have registered a mondorescue project on
BerliOS, which provides a sourceforge like
environement, including SVN support.

I've taken the latest tar ball available from
mondo (2.04_cvs_20050523) and mindi
(1.04_cvs_20050523) on
http://www.mondorescue.org and initiate an
SVN repository with them.  I've patches the
code with what I had locally.  The Changelog
looks like:

v2.04_berlios (Bruno Cornec <bcornec
@users.berlios.de)
- Add -p option to generate ISO images
file names with prefix.  The new default
name for ISO images is mondorescue-1.iso
- Mandrake 2005 support
- NFS patches (Yann Aubert <technique
@alixen.fr>)

I'd be happy to add new developpers to SVN
write access list.  You just have to create
an account and send it to me so that I can
add you.  The SVN access is available through
https:
//developer.berlios.de/svn/?group_id=2524

I've setup a tiny Web page and Wiki.  The
main access is at https:
//developer.berlios.de/projects/mondorescue/
```

> And for those of you who would like to try the new version, it can be downloaded from there or from ftp://ftp.berlios.de/pub/mondorescue Please register bugs and feature requests as well.
>
> I do not promise anything, as I don't have much time myself, but hope to be able to get help from you so that the project goes on living.
>
> Bruno.

It soon became obvious that it would not just be a transition in tools to manage the project, but also a change in the development team. At that time, I considered myself just as a maintainer of a branch that should be reintegrated in the standard tree, when possible. But in fact, that branch was becoming the standard tree! So after some discussions on the mailing list, and to avoid confusion, it was decided that the next version would be named 2.05 to show it was a better version than the prior one made by Hugo (integration of lots of patches, and a first build process approach).

As in addition to maintaining the code itself (enhancements, bug fixes), it was also necessary to take over all the things related to the project: The subversion repository (September, 2005), the Web site content (fully rewritten and delivered in February, 2006), the domain names (`mondorescue.org/.com` in January, 2006), the HOWTO (January, 2006), animation of the mailing-list (could be a full-time job `;-)`, announcements on freshmeat, etc.

Since we took over the maintenance, 11 versions have been published. To sustain the development effort, it also became obvious that being hosted was less convenient and powerful than having a dedicated machine providing those services for the project. So in July of 2006, HP gave me a server dedicated to the project which is hosted by the HP/Intel Solution Center in Grenoble, France, where I work. It has allowed us to manage our own Subversion repository and the Web site (without the unavailability met on hosting platforms), and to add additional functions such as a real ftp site and Trac, an ingenious tool to help manage projects (Wiki, Bug tracking, and SVN browsing). We couldn't decently work without it today. Sympa is probably the next candidate to host mailing-lists.

Three generations of build process have also been developed based on another ingenious tool, QEMU (for Virtual Machine (VM) emulation), and home-made scripts (which have evolved into a separate project at `http://trac.project-builder.org`) to build for 50+ distributions now from a single set of source files. This allows us to launch a single script to activate all those builds in sequence from the SVN repository automatically. This is a great help in ensuring coherency and consistency across distributions, and makes it easy for users to install it on their distribution of choice. This work will also serve as a base for automated testing in the future.

The future version 2.2.7 is planned for end of July, 2007, and will propose support of the latest distributions (OpenSuSE 11, etc.), various fixes on 2.2.6, etc.

The 2.2.x branch of subversion should then enter into maintenance mode and all our efforts will concentrate on producing the long-promised 3.0.X version. The main changes are internal, with the introduction of a new low-level internal library, with test routines to improve reliability using only dynamic memory allocation. The rest of the code is being parsed to remove as much static memory usage as possible during the 3.0.X timespan. As part of that work, configuration files for both `mindi` and `mondoarchive` will be introduced to help users change the way the tool works, without dealing with the code at all. Also a web interface used to generate the options of the `mondoarchive` will be available, helping with the integration of mondo in another project called `dploy.org`, made in conjunction with the LinuxCOE team managed by HP fellows (cf. `http://trac.dploy.org`). New features will also include a internationalization support, maybe reactivated FreeBSD support, etc.

3 Getting Help

There are multiple ways to get help when you encounter issues with MondoRescue.

- Look at the documentation available on Mondorescue Web site at `http://www.mondorescue.org/docs.shtml` (includes HOWTO, man pages, migration docs, etc.).

- Look at the FAQ on the Wiki at `http://trac.mondorescue.org/wiki/FAQ`. It contains valuable tips and tricks for some common issues.

- Read the messages on screen, and the log files. 90% of the time they just contain the information which will help you help yourself. They are located under `/var/log/mindi.log` for `mindi`, `/var/log/mondoarchive.log` for mondoarchive, and `/var/log/mondorestore.log` during the restoration process.

- Look at `http://sourceforge.net/mailarchive/forum.php?max_rows=25&style=ultimate&offset=25&forum_name=mondo-devel` for archives of the mailing list, which may already contain information about your issue.

- If you think you need external help, you may reach the Mondorescue community via the regular mailing list (`mailto:mondo-devel@lists.sourceforge.net`). But that community can only help you if your provide at least the above-mentioned log files.

- If your think you have found a bug in Mondorescue, please fill a bug report at `http://trac.mondorescue.org/newticket` and also include the log files. Likewise, if you wish Mondorescue had your new shiny dream-function, feel free to create a feature request at the same URL.

- Access to source code is provided as per the license of the software (GPL). So you may want to add/remove/modify whatever feature; access is made available through FTP at `ftp://ftp.mondorescue.org/src` and SVN at `svn co svn://svn.mondorescue.org/mondorescue/branches/stable` (replace stable by the branch you want to get, which could be 2.2.6 for the latest stable 2.2 published—the stable name is for 3.0.x). Patches are more than welcome `;-)`. All packages provided are available at `ftp://ftp.mondorescue.org/`. Read the instructions on `http://trac.mondorescue.org/wiki/DistributionPackaging` if you want to rebuild either your packages or a version from source of the MondoRescue project, or apply some specific patches.