RAID set creation takes only a few moments, and once it's complete, you should see new RAID set volume in the Disk Utility list and in the Finder. Selecting the newly created RAID set from the column on the left, and then clicking the RAID tab, should reveal that the set is "Online."

Rebuilding a RAID Set

As mentioned earlier, using multiple drives in a RAID set actually introduces more points of storage failure. Fortunately, RAID 1 data mirroring configurations are designed specifically to prevent data loss when a drive fails. Mac OS X even includes the ability to automatically repair mirrored RAID sets if you specified that option during RAID creation.

Before the system mounts a RAID volume, it will check the set for consistency. If the system finds a degraded striped RAID 0 set, you'd better have a good backup because all that data is lost. The system will report the degraded RAID set in Disk Utility, but it will not mount the volume. Only a data recovery service, such as DriveSavers, might have a chance at recovering your data.

Note: For more about data recovery service providers that can be used without voiding an Apple warranty, see http://docs.info.apple.com/article.html?artnum=31077.

On the other hand, if the system finds that a mirrored RAID 1 set is degraded it will either warn you or automatically start rebuilding the RAID set if configured. Either way, the volume will still mount and be accessible to you in the Finder. You should avoid writing new data to a degraded RAID set until you have completed the rebuilding process.

There are two main failure modes for a mirrored RAID 1 set:

One of the drives appears to be responding properly, but the data on the drive is not consistent with the other drives in the set. If configured, the system will automatically start rebuilding the RAID set data by recopying it from a working drive to the apparently corrupted drive. Otherwise, the system will wait for you to manually engage the rebuild process from Disk Utility.

One of the drives in the set is no longer available. If a spare is configured, the system will automatically start rebuilding the RAID set data by copying it from a working drive to the spare drive. Otherwise, the system will wait for you to manually replace the drive and manually engage the rebuild process from Disk Utility.

media. It doesn't matter how good the software is, though; if the hardware is no longer reading or writing bits, it's pretty much useless to the file system. If, during troubleshooting, you determine that catastrophic hardware failure is the problem, there really isn't anything you can do to repair the device. Only a data recovery service, such as DriveSavers, might have a chance at recovering your data.

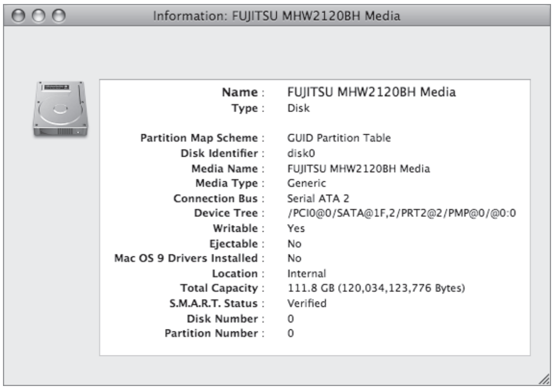
More Info: For more about data recovery service providers that can be used without voiding an Apple warranty, see http://docs.info.apple.com/article.html?artnum=31077.

Conversely, if you're experiencing file system issues but the storage hardware still appears to function, you may be experiencing partial hardware failure or file system corruption. In these cases, there are some steps you can try to repair the volumes or at least recover data.

Gathering File System Information

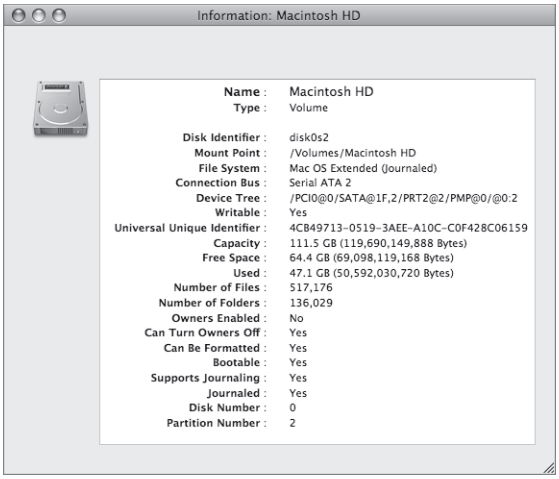
Before attempting any fixes, you should become fully familiar with the file system configuration you're dealing with. Once again, /Applications/Utilities/Disk Utility will be your main tool for gathering file system information. The availability and status of storage hardware in Disk Utility will help determine if you are indeed experiencing hardware failure.

When you open Disk Utility, it will scan the file system for all attached devices and volumes. To gather detailed information about a specific drive or volume in Disk Utility, simply select the item from the column on the left and then click the Info button in the toolbar. Remember, the drive's name is its size and manufacturer information, whereas a drive's volumes appear directly below the drive name in the list.



Information: FUJITSU MHW2120BH Media

Name :	FUJITSU MHW2120BH Media
Type :	Disk
Partition Map Scheme :	GUID Partition Table
Disk Identifier :	disk0
Media Name :	FUJITSU MHW2120BH Media
Media Type :	Generic
Connection Bus :	Serial ATA 2
Device Tree :	/PCI0@0/SATA@1F,2/PRT2@2/PMP@0/@0:0
Writable :	Yes
Ejectable :	No
Mac OS 9 Drivers Installed :	No
Location :	Internal
Total Capacity :	111.8 GB (120,034,123,776 Bytes)
S.M.A.R.T. Status :	Verified
Disk Number :	0
Partition Number :	0



Information: Macintosh HD

Name :	Macintosh HD
Type :	Volume
Disk Identifier :	disk0s2
Mount Point :	/Volumes/Macintosh HD
File System :	Mac OS Extended (Journaled)
Connection Bus :	Serial ATA 2
Device Tree :	/PCI0@0/SATA@1F,2/PRT2@2/PMP@0/@0:2
Writable :	Yes
Universal Unique Identifier :	4CB49713-0519-3AEE-A10C-C0F428C06159
Capacity :	111.5 GB (119,690,149,888 Bytes)
Free Space :	64.4 GB (69,098,119,168 Bytes)
Used :	47.1 GB (50,592,030,720 Bytes)
Number of Files :	517,176
Number of Folders :	136,029
Owners Enabled :	No
Can Turn Owners Off :	Yes
Can Be Formatted :	Yes
Bootable :	Yes
Supports Journaling :	Yes
Journaled :	Yes
Disk Number :	0
Partition Number :	2

Navigation Concepts

First, a few common navigation terms must be clearly defined. This book has thus far referred to file system containers using the common term “folder.” Yet another term, “directory,” is often used in the CLI as a synonym for folder. Though the terms can be used interchangeably, this book will continue to favor the word “folder” to describe file system containers, as the word “directory” is often used for other non-folder-like items. For example, network databases used to store user information are often referred to as “directories.” Furthermore, the process in Mac OS X for accessing these user databases is called the DirectoryService.

A new term you’ll see in this book will is “path.” A path represents a file or folder’s location in the file system. You have already seen paths in this book used to describe the specific location of an application or utility. For instance, the Disk Utility application’s file system path is /Applications/Utilities/Disk Utility. The CLI uses pathnames exclusively for navigating and locating items in the file system.

There are two types of file system pathnames: absolute paths and relative paths. Either type is valid for navigating or locating items in the CLI, but they differ in where they start:

- Absolute paths—Absolute paths are full descriptions of an item’s location starting from the root, or beginning, of the file system. Thus, an absolute path will always begin with a forward slash to indicate the beginning of the file system. This book uses absolute paths to describe the location of items. An example of the absolute path to the user Michelle’s Drop Box folder would be ~~/Applications~~/Users/michelle/Public/Drop Box.

[delete:] /Applications

- Relative paths—Relative paths are partial descriptions of an item’s location based on where you’re currently working in the file system from the CLI. When you first open the Terminal application, your CLI session starts out working from the root, or beginning, of your home folder. Therefore, the relative path from the beginning of your home folder to your Drop Box would be Public/Drop Box.

Navigation Commands

You will use three basic commands for navigating the file system in the CLI: pwd, ls, and cd.

pwd. Short for “print working directory,” this command will report the absolute path of your current working location in the file system:

```
MyMac:~ michelle$ pwd
/Users/michelle
```

ls. Short for “list,” ls will list the folder contents of your current working location. Entering a pathname following the ls command will list the contents of the specified item. The ls command has many additional options for listing file and folder information that will be covered throughout this book.

```
MyMac:~ michelle$ ls
Desktop   Library   Pictures
Documents  Movies   Public
Downloads  Music   Sites
MyMac:~ michelle$ ls Public
Drop Box
```

cd. Short for “change directory,” cd is the command you will use to navigate in the CLI. Entering a pathname following the cd command will change your current working location to the specified folder. Entering cd without specifying a path will always return you to your home folder. In the following example, Michelle will use the cd command to navigate to her Drop Box folder, and then she will navigate back to her home folder:

```
MyMac:~ michelle$ cd Public/Drop\ Box/
MyMac:Drop Box michelle$ pwd
/Users/michelle/Public/Drop Box
MyMac:Drop Box michelle$ cd
MyMac:~ michelle$ pwd
/Users/michelle
```

Navigating to Other Volumes

[replace:] forward slash,

In the CLI, the system volume is also known as the root volume and it’s identified by the lone ~~backslash~~, /. It may come as some surprise to you, however, that in the CLI other nonroot volumes appear as part of the main file system in a folder called Volumes. In the following example, Michelle will start in her home folder, navigate to and list the items in the /Volumes folder, and then finally navigate into a volume connected to this Mac via a FireWire disk drive:

```
MyMac:~ michelle$ pwd
/Users/michelle
MyMac:~ michelle$ cd /Volumes/
MyMac:Volumes michelle$ pwd
/Volumes
```

This practice is necessary because the CLI uses the spaces between items to parse the command entry into separate logical pieces. A space in a filename without the ~~forward slash~~ will confuse the CLI, and your command will not execute properly.

[replace:] backslash

There are other methods for entering filenames and paths with spaces. One alternative is to surround filenames and paths with quotation marks:

```
MyMac:~ michelle$ cd “Public/Drop Box”
MyMac:Drop Box michelle$ pwd
/Users/michelle/Public/Drop Box
```

Another solution involves dragging and dropping items from the Finder to the Terminal window. The Terminal automatically enters the item’s absolute path with the appropriate backslash characters before spaces in names. The most efficient solution, though, is to use the tab-complete feature built into Mac OS X’s CLI to automatically complete file and pathnames for you. Saving time by using tab completion is covered next in this lesson.

When navigating the file system, you can also save time by using the double period (..) shortcut to indicate the previous folder. In other words, if you were in working your home folder located at /Users/username, entering `cd ..` would tell the CLI that you want to navigate to the /Users folder. In the following example, Michelle navigates to her Drop Box folder, backs up to her Public folder, and then finally backs up twice to the /Users folder:

```
MyMac:~ michelle$ cd Public/Drop\ Box/
MyMac:Drop Box michelle$ pwd
/Users/michelle/Public/Drop Box
MyMac:Drop Box michelle$ cd ..
MyMac:Public michelle$ pwd
/Users/michelle/Public
MyMac:Public michelle$ cd ../..
MyMac:Users michelle$ pwd
/Users
```

Finally, there is the ~, or tilde. This little guy is used as shorthand to describe the current user’s home folder in a pathname. Once again using the example from earlier, the current user’s Drop Box is located at ~/Public/Drop Box. This also helps to explain the tilde you see in the default command prompt. For example, if Michelle opened the Terminal and logged on as herself on a computer called MyMac, the command prompt would show

In the following example, Michelle will use the chmod command to change the permissions of testfile1 and testfolder to allow read and write access for the owner and the group but read-only access for everyone else. She will first use alphanumeric abbreviations, and then octal permission equivalents.

NOTE ▶ Remember, if you want to change the permissions of a folder and its contents, you must tell the chmod command to run recursively by adding the -R option.

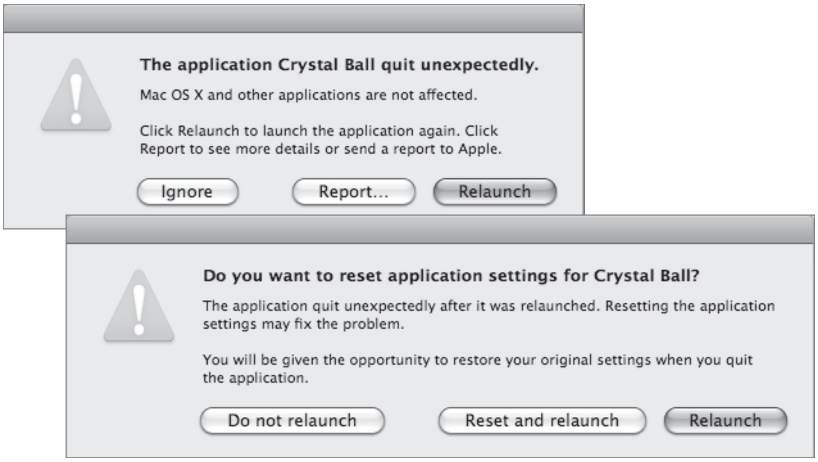
```
MyMac:~ michelle$ ls -l Desktop/
total 0
-rw-r--r-- 1 michelle staff 0 Aug 20 15:49 testfile1
drwxr-xr-x 4 michelle staff 136 Aug 20 15:47 testfolder
MyMac:~ michelle$ chmod ug=rw,o=r Desktop/testfile1
MyMac:~ michelle$ ls -l Desktop/
total 0
-rw-rw-r-- 1 michelle staff 0 Aug 20 15:49 testfile1
drwxr-xr-x 4 michelle staff 136 Aug 20 15:47 testfolder
MyMac:~ michelle$ chmod ~~775~~ Desktop/testfolder/          [replace:] 664
MyMac:~ michelle$ ls -l Desktop/
total 0
-rw-rw-r-- 1 michelle staff 0 Aug 20 15:49 testfile1
drwxrwxr-x 4 michelle staff 136 Aug 20 15:47 testfolder
```

Using the Sticky Bit

As mentioned previously in this lesson, the /Users/Shared folder has a unique permissions setting that allows all local users to read and write items into the folder yet prevents other users from being able to delete files that they didn't originally put in this folder. This special permissions configuration is brought to you courtesy of the "sticky bit." Essentially, enabling the sticky bit on a folder defines it as an append-only destination, or, more accurately, a folder in which only the owner of the item can delete the item.

You can clearly see the sticky bit setting of the /Users/Shared folder when you view its ownership and permissions. Note the t on the end of the permissions information, which indicates that the sticky bit is enabled:

```
MyMac:~ michelle$ ls -l /Users/
total 0
drwxr-xr-x+ 13 Guest  staff 442 Aug 9 18:33 Guest
```



The application Crystal Ball quit unexpectedly.

Mac OS X and other applications are not affected.

Click Relaunch to launch the application again. Click Report to see more details or send a report to Apple.

Do you want to reset application settings for Crystal Ball?

The application quit unexpectedly after it was relaunched. Resetting the application settings may fix the problem.

You will be given the opportunity to restore your original settings when you quit the application.

Preference Troubleshooting

Application preference files are the most common application resources to cause problems. These files contain both internal application configuration information and user-configured preferences. Even if you haven't changed any preferences, odds are the application is constantly storing new information in this file. It is the only file required by the application that is constantly being rewritten, so it's ripe for corruption. Corrupted preferences typically result in an application that crashes frequently or during startup.

Not all applications are compatible with the safe launch feature, so you may have to resolve corrupted preferences issues manually. As covered earlier in this lesson, application preferences are located in the ~/Library/~~Permissions~~ folder and are named by the application and its maker. The most convenient method of isolating a corrupted preference in this case is to rename the suspect preference file in the Finder by adding an identifier to the end of the file name—something like ".bad." Restarting the application creates a new preference file based on the application's defaults. If this resolves the issue, go ahead and trash the old preference file; if not, you should move on to application resource troubleshooting. If you eventually resolve the problem elsewhere, you can restore the user's previous settings by deleting the newer preference file and then removing the filename identifier you added to the original preference file.

[change "Permissions" to "Preferences"]