

FileVault 2 Decoded

Rich Trouton

Howard Hughes Medical Institute,
Janelia Farm Research Campus

HHMI
HOWARD HUGHES MEDICAL INSTITUTE

Before we get started, there's two things I'd like to mention. The first is that, all of the slides, speakers' notes and the demos are available for download and I'll be providing a link at the end of the talk. I tend to be one of those folks who can't keep up with the speaker and take notes at the same time, so for those folks in the same situation, no need to take notes. Everything I'm covering is going to be available for download.

The second is to please hold all questions until the end. If you've got questions, make a note of them and hit me at the end of the talk. With luck, I'll be able to answer most of your questions during the talk itself.

Similar Names, Different Beasts

- Apple has completely revamped FileVault in Lion
- Grown from an encryption solution that protected only home folders to one that can protect entire drives.
- For simplicity, the older FileVault encryption will be referred to as “FileVault 1” during this talk.

One of the changes that Apple introduced with Lion is that its FileVault encryption solution has been completely revamped, changing it from encryption that primarily protected your account's home folder to encryption that protects your whole boot volume. Despite the common name, the two solutions are very different beasts. If you've used the older FileVault, be aware that everything you know about it is changed in Lion and later. Since Apple's dubbed the new encryption as FileVault 2, I'm going to refer to the older FileVault as FileVault 1.

FileVault 1 Upsides

- Strong encryption
- Came with the OS, no extra charge to use it.
- Designed to work like an decrypted account wherever possible.



FileVault 1 had some upsides and downsides. The biggest upsides were that it used strong encryption, was low cost, and it was designed for ease of use.

FileVault 1 Downsides

- Backups
- Network accounts
- Whole disk encryption not available



The biggest downsides were that it was difficult to back up the encrypted home folder, there were problems with using FileVault 1 with network accounts, and that FileVault 1 was not able to provide whole disk encryption.

Back to the drawing board

- Complete rebuild of FileVault for Lion
 - Uses new virtual volume storage (Core Storage.)
- Core Storage encrypted volumes are built on a per-partition basis.
- Allows both encrypted and decrypted partitions on the same physical hard drive.

Apple had known about its customers' problems with FileVault 1 as it was designed and went back to the drawing board to provide a full disk encryption solution. Ultimately, the effort would require the complete rebuild of FileVault from the ground up, basing the new FileVault 2 on an entirely new type of virtual volume storage, which Apple named Core Storage. While you can make unencrypted Core Storage volumes, as Apple is doing with the new Fusion drives, its most common use in Lion and later is to provide FileVault's encrypted volume storage. Core Storage works on a per-partition basis, which means that you can have both encrypted and unencrypted partitions on the same physical hard drive.

How FileVault 2 works

- On startup, the Mac initially boots to a small decrypted partition that only provides access to the tools to unlock the larger encrypted storage.
- When the right authentication is provided, the encryption unlocks and the Mac boots from the Mac's regular OS.
- By unlocking the encryption before the OS boots, the issues with network accounts and backups are solved.

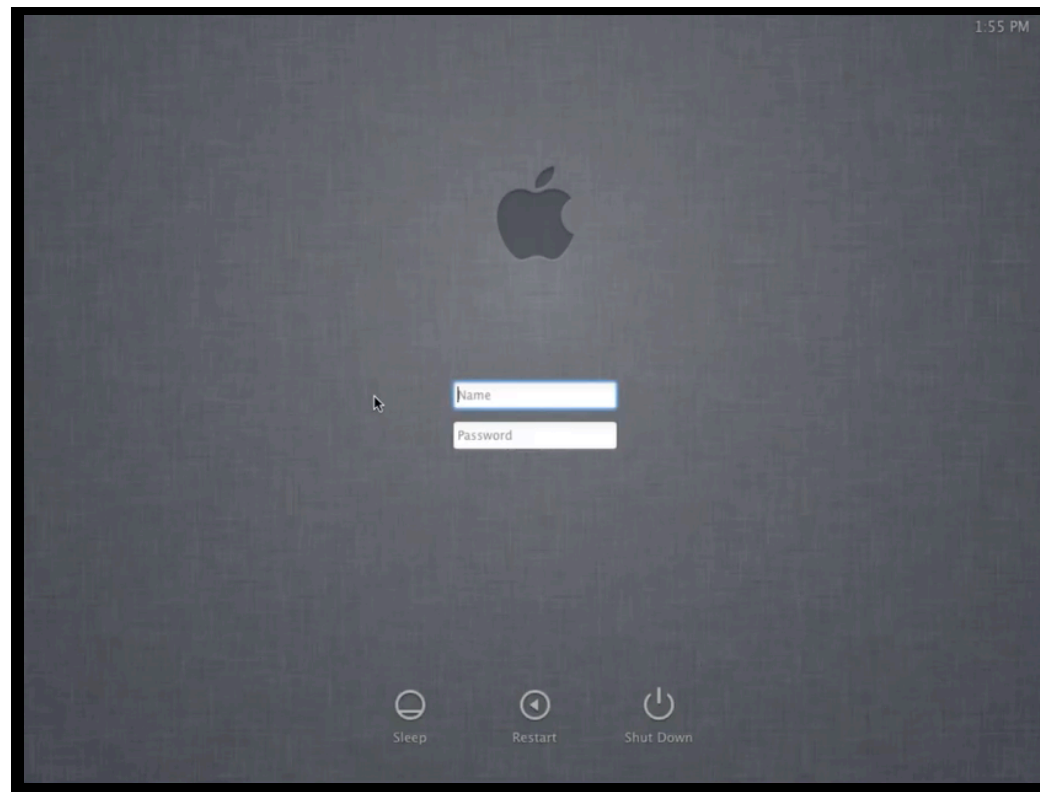
FileVault 2, in broad terms, works like PGP or other full disk encryption solutions available for the Mac. When you start up your system, a small unencrypted partition provides the needed tools to unlock the much larger encrypted storage, usually through what's called a pre-boot login screen. If you provide the right authentication to the pre-boot login, the encrypted storage then unlocks and the OS on the machine takes over and boots the Mac. Once the Mac is unlocked and booted, the user doesn't have the issues with backups and network accounts that FileVault 1 users had because, from the OS's perspective, everything on the hard drive it needs to access is accessible.

FileVault 2 and Recovery HD

- One of the other new features in Lion was the Recovery HD partition.
- Small hidden partition that provides tools to fix or reinstall the OS.
- To use FileVault 2, you need to have the Recovery HD partition present.
- Why? Because Recovery HD provides the unencrypted space needed to unlock and boot your encrypted Mac.

Now, because not everything about FileVault 2 is obvious at first glance, I want to make sure to cover one of the lesser-known relationships in 10.7 and later, which is the relationship between Recovery HD and FileVault 2.

So, what does Recovery HD have to do with FileVault 2? FileVault 2 encrypts your boot partition, but your Mac still needs an unencrypted space to boot to and allow access to the unlock tools. The recovery partition serves as the needed space. The FileVault encryption process will check before beginning the encryption to see if the recovery partition is there and will not start the encryption process if it's not there.



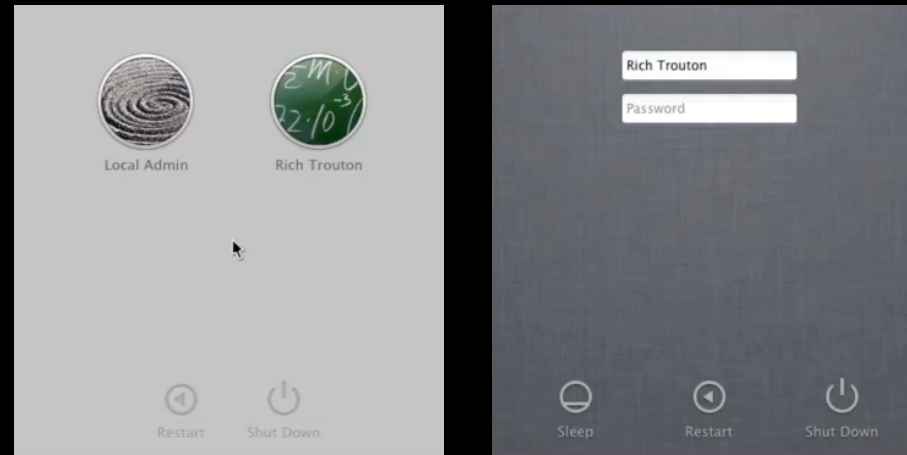
Next, we're going to enable our existing users by clicking Enable User, entering the account's password and clicking OK. However, if you notice, there's one account that's already checked off as enabled. How'd that happen?

How that happens is actually pretty neat. When you click the lock to unlock the FileVault preferences, FileVault 2 cached your authentication credentials. When you clicked Turn On FileVault, those same credentials were passed silently in the background to enable the account.

In fact, Apple took this functionality and extended it in a couple of ways. The first was that, if the administrator account you're using is the only account on the Mac, you're not prompted to authorize any accounts. FileVault 2 is smart enough to cache the credentials and not prompt you.

The second was that, if an existing admin account needs to be enabled, it can become an authorized account via the same authorization handoff. The new account would need to log in at the regular login window, open the FileVault preferences, click the lock, then click on the Enable Users button that will appear at the bottom of the window.

Using the Recovery Key To Reset Password



The chief use of the FileVault 2 individual recovery key is to help you reset your account password in the event that you forget what your password is. To use the recovery key at the pre-boot login screen, enter your password incorrectly three times. After that, the login screen should prompt you for the recovery key.

An important thing to note here is that this functionality was built to reset the password of a local account. If you have a network account, where your password is being managed by a directory service like Active Directory or Open Directory, resetting your password using this method will likely break the password sync between your account on the Mac and the directory service. In that event, you will need to delete and re-setup the account on the Mac in order to fix this.

When you enter the recovery key, FileVault 2 will take you to the regular login window and prompt you to reset your password. Once the password is reset, the rest of the login process will complete.

Managed Deployment

- With the exception of how the recovery key is generated and handled, setting up FileVault for home use and setting it up for a managed deployment is exactly the same.
- To avoid the administrative headache of tracking multiple individual recovery keys, Apple brought one part of FileVault 1 into FileVault 2.

For those of you who want to roll out a managed deployment of FileVault 2, deploying and supporting FileVault 2 in a larger Mac environment is almost exactly the same as setting it up for an individual. Apple recognized that it would be an administrative headache to track multiple individual recovery keys, so the sole difference between managed and unmanaged is how the recovery key is generated. That recovery key system is the one part of FileVault 1 that has survived in FileVault 2.

Sole Survivor - FileVaultMaster.keychain

- What does FileVaultMaster.keychain do?
 - It sets a backdoor to encrypted Macs and is an alternate way to unlock the encryption when the account passwords don't work.
 - Apple calls this the Master Password.
- What role does the password you set as the Master Password actually play?
 - It's the password used to unlock the FileVaultMaster.keychain.



What does FileVaultMaster.keychain do?

FileVaultMaster.keychain serves as a centrally-set backdoor. In the event that you weren't able to get into an encrypted Mac using the usual passwords, you could use FileVaultMaster.keychain to unlock the encryption and recover the data stored inside. In the FileVault context, Apple historically called this a "master password". This master password is in fact the password to a FileVault-specific keychain.

What role does the password that you set as the Master Password play?

It is the password used to secure the FileVaultMaster.keychain itself. Because this password is only used to unlock the keychain, this password can be rotated without affecting the contents of that keychain (much like how changing your account password doesn't affect the contents of your login keychain.)

Sole Survivor - FileVaultMaster.keychain

- Wait, what? If all the Master Password does is unlock a keychain, how does it do recovery?
 - FileVaultMaster.keychain's contents are what actually do the recovery.
 - Inside the keychain is a public key (shows up as a SSL certificate) and an accompanying private key.
 - When you have both keys available in the FileVaultMaster keychain, you can use them to unlock or decrypt FileVault 2's encryption.



FileVaultMaster.keychain is a FileVault-specific keychain, whose only contents are an SSL certificate (referred to in this talk as the public key) and an accompanying private key. These two keys are specifically used for FileVault certificate-based authentication.

When both are available, you can use them to unlock or decrypt FileVault 2's encryption without knowing any of the passwords of the accounts authorized to log in at the pre-boot login screen.

Sole Survivor - FileVaultMaster.keychain

- You can set a institutional recovery key for a FileVault 2 managed deployment in exactly the same way that you set the recovery key in FileVault 1.
- Crucial difference
 - In FileVault 1, you could have both the private and public key stored in the FileVaultMaster.keychain when you encrypted.
 - In FileVault 2, only the public key can be stored in FileVaultMaster.keychain when you encrypt your Mac.

For those who are experienced with a managed FileVault 1 deployment, setting a FileVault 2 recovery key with FileVaultMaster.keychain is exactly the same as setting it with FileVault 1. In fact, for those following Apple's recommended best practice, you don't need to change anything.

The reason I mentioned Apple's recommended best practice is that Apple has always recommended escrowing the FileVault private key somewhere outside of the encrypted Mac. However, in FileVault 1, you could have both the private and public key stored in FileVaultMaster.keychain and it worked fine.

In FileVault 2, this recommendation is now a requirement. If FileVault 2 detects both the private and public keys in FileVaultMaster.keychain, it makes the programmatic assumption that this is leftover from an older OS and ignores it. In this case, it'll default to setting up an individual recovery key.

Preparing FileVaultMaster.keychain

- Make your FileVaultMaster.keychain by setting the Master Password on a specific machine. (You can skip this step if you've already got a set FileVaultMaster.keychain)
 - This FileVaultMaster.keychain will contain both private and public keys.
- Next, make several copies of the FileVaultMaster.keychain file and store the copies in a secure place.
 - A locked safe would be a good place, or in an encrypted disk image on a secured file share.

If you do not already have a pre-built FileVaultMaster.keychain, you'll need to create one and add it to your machines before encrypting them. Build your keychain by setting the Master Password on a particular machine and copying the FileVaultMaster.keychain file off of it. At this point, the keychain will contain both the private and public keys needed for FileVault recovery.

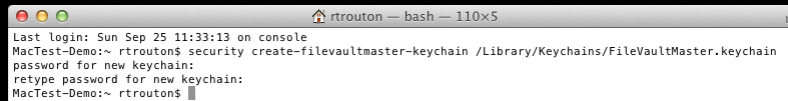
Once you have your keychain, make several copies and store them in a safe place. Everyone has a different conception of "safe", but I usually recommend storing them in a safe, or inside an encrypted disk image that is itself stored somewhere secure.

Preparing FileVaultMaster.keychain

In 10.7.2 and higher, you can create FileVaultMaster.keychain from the command line.

To create a FileVaultMaster.keychain:

security create-filevaultmaster-keychain /path/to/FileVaultMaster.keychain

A screenshot of a macOS terminal window titled 'rtrouton — bash — 110x5'. The terminal shows the following text: 'Last login: Sun Sep 25 11:33:13 on console', 'MacTest-Demo:~ rtrouton\$ security create-filevaultmaster-keychain /Library/Keychains/FileVaultMaster.keychain', 'password for new keychain:', 'retype password for new keychain:', and 'MacTest-Demo:~ rtrouton\$'. The cursor is at the end of the last line.

```
MacTest-Demo:~ rtrouton$ security create-filevaultmaster-keychain /Library/Keychains/FileVaultMaster.keychain
password for new keychain:
retype password for new keychain:
MacTest-Demo:~ rtrouton$
```

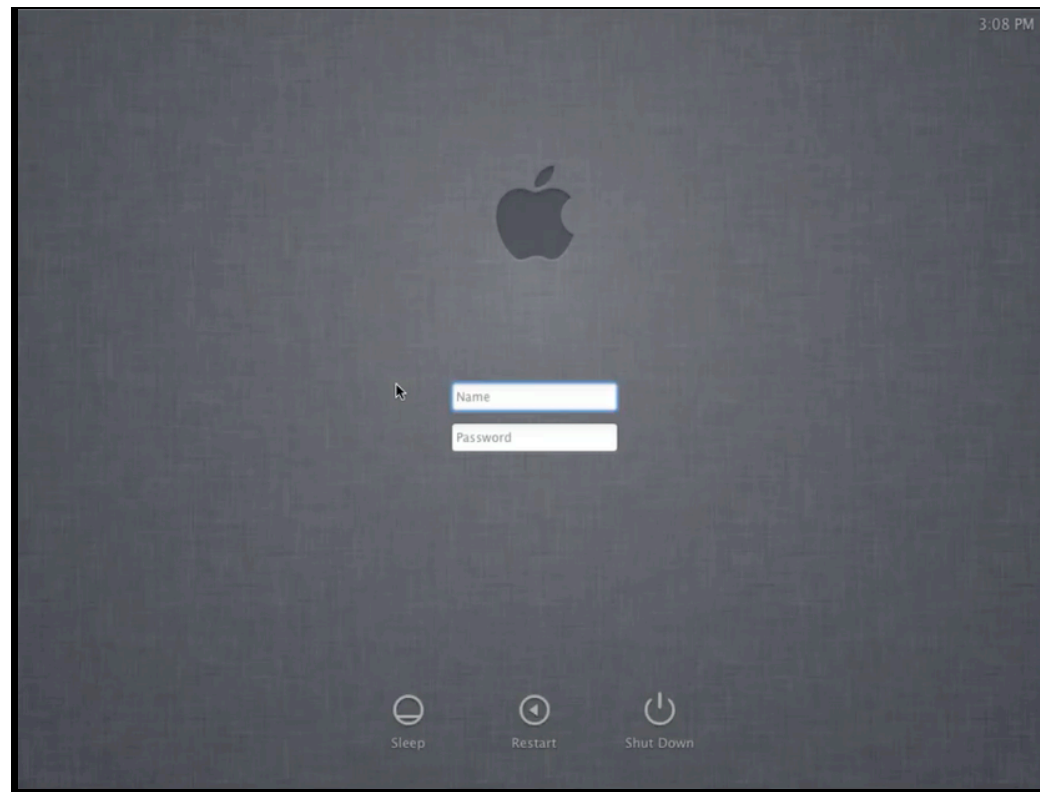
You'll be prompted to set a password. Please enter the Master Password you want to use at this point.

In 10.7.2 and later, you can use the security command to create a FileVaultMaster.keychain from the command line. This method creates a 2048-bit private key by default.

Preparing FileVaultMaster.keychain

- Once you've got copies, unlock your FileVaultMaster.keychain by running the command below and entering the Master Password when prompted for the password:
security unlock-keychain /path/to/FileVaultMaster.keychain
- After FileVaultMaster.keychain unlocks, go into Keychain Access and access FileVaultMaster.keychain
 - Remove the private key. It will be called **FileVault Master Password Key** and its kind is listed as **private key**.

Once you've got your copies safely stowed, edit yet another copy of your keychain and remove the private key to prepare for use as your FileVault 2 recovery key.

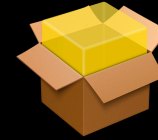


During the demo, the following points will be covered:

1. Going through the procedure of creating an institutional recovery key for FileVault 2.
2. Showing the encryption process when using an institutional recovery key.

Deploying FileVaultMaster.keychain

- You can deploy the prepared FileVaultMaster.keychain using a variety of methods.
 - Install it via an installer package
 - Include it with your image
 - Copy it to your Macs using your system management tool(s).
- A FileVaultMaster.keychain can be built once and then deployed to as many Macs as needed.



Once you have your FileVaultMaster.keychain ready, you can take that keychain and deploy it to as many machines as needed.

Disaster Recovery

Sometimes bad things happen
to good Macs

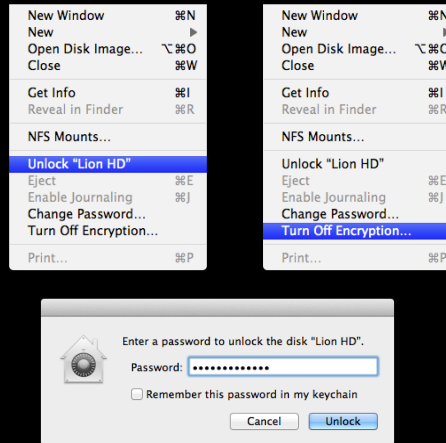
- If you have the password to an authorized account available, you can unlock and/or decrypt from Disk Utility or the command line.
- You can also use your recovery key to unlock and/or decrypt from the command line.

Disaster recovery is always something you should plan for when dealing with encrypted machines. After all, these are designed to protect your data against external threats unless properly authenticated. If your OS takes a dive, your normal way of unlocking may no longer work.

The first thing you should generally try is the password to one of the authorized accounts. This applies to all FileVault 2 encrypted Macs, regardless of the kind of recovery key being used.

Recovering using your password

- Boot your Mac and hold down ⌘-R (Command – R) to boot from the Mac's Recovery HD partition.
- Use Disk Utility and the password of an authorized user to either unlock or turn off the encryption.



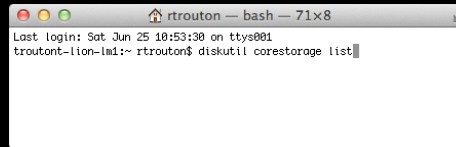
If you have the password to an authorized account, unlocking or even turning off the encryption is pretty straightforward. Boot to Recovery HD or another boot drive running 10.7 and later, open Disk Utility and either select “Unlock” or “Turn off encryption”. You’ll be prompted for a password of an authorized account, you provide it, and you’re golden.

Recovering from Terminal

Boot your Mac and hold down ⌘-R (Command -R) to boot from the Mac's Recovery HD partition.

Open Terminal and use the following to you need to identify the Logical Volume UUID of the encrypted drive.

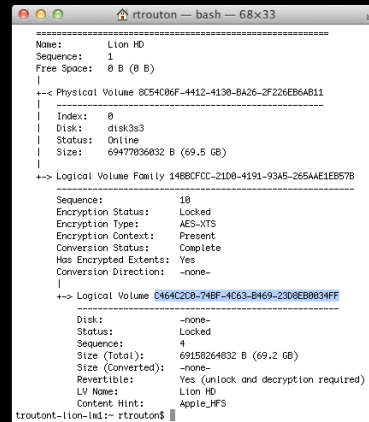
diskutil corestorage list



You can also recover from the Terminal, but you'll need a few more pieces of information. The first is to get the Logical Volume UUID of the drive you want to work on. You can do this by running “diskutil corestorage list”. You can also substitute “cs” in place of “corestorage” in commands.

Recovering from Terminal

Once you have the UUID, you'll use it to identify the disk to be unlocked or decrypted.



```
trouton ~$ diskutil list
Name: Lion HD
Sequence: 1
Free Space: 0 B (0 B)
--< Physical Volume 8C54C06F-4412-4138-BA26-2F226EB6A811
|
| Index: 0
| Disk: disk3s3
| Status: Online
| Size: 69477036832 B (69.5 GB)
--> Logical Volume Family 1488CFCC-21D0-4191-93A5-265AAE1EB57B
|
| Sequence: 10
| Encryption Status: Locked
| Encryption Type: AES-XTS
| Encryption Context: Present
| Conversion Status: Complete
| Has Encrypted Extents: Yes
| Conversion Direction: --none--
|
| --> Logical Volume 0464C2D0-74BF-4C63-B469-23C6E80034FE
|
| Disk: --none--
| Status: Locked
| Sequence: 4
| Size (Total): 69158264832 B (69.2 GB)
| Size (Converted): --none--
| Revertible: Yes (unlock and decryption required)
| LV Name: Lion HD
| Content Hint: Apple_HFS
trouton~lion-lal:~ trouton$
```

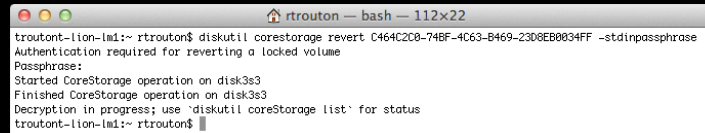
Once you have the UUID, you can then use it to specifically ID the drives you want to either unlock or decrypt. In the slide, I've highlighted what you should be looking for when checking for the UUID.

Recovering from Terminal

Unlocking or encrypting using your password

To unlock: *diskutil corestorage unlockVolume UUID -stdinpassphrase*

To decrypt: *diskutil corestorage revert UUID -stdinpassphrase*

A screenshot of a macOS terminal window titled "rtrouton -- bash -- 112x22". The terminal shows the command "diskutil corestorage revert C464C2C8-74BF-4C63-B469-23D8EB0834FF -stdinpassphrase" being executed. The output indicates that authentication is required, a passphrase is entered, and the CoreStorage operation on disk3s3 is completed. It also shows the decryption in progress and suggests using "diskutil corestorage list" for status. The prompt returns to "trouton-lion-lai:~ rtrouton\$".

```
trouton-lion-lai:~ rtrouton$ diskutil corestorage revert C464C2C8-74BF-4C63-B469-23D8EB0834FF -stdinpassphrase
Authentication required for reverting a locked volume
Passphrase:
Started CoreStorage operation on disk3s3
Finished CoreStorage operation on disk3s3
Decryption in progress; use 'diskutil corestorage list' for status
trouton-lion-lai:~ rtrouton$
```

The *-stdinpassphrase* flag will cause the command to prompt you for the password/passphrase of an account that's authorized to unlock the encryption.

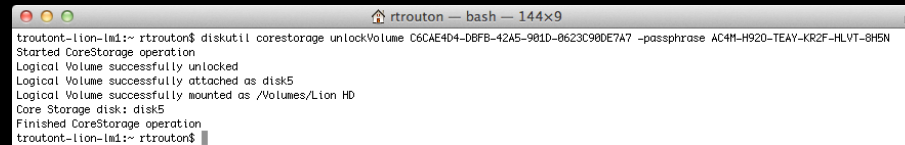
You can unlock or decrypt using the password of an authorized user using the commands shown on the screen.

Recovering from Terminal

Unlocking or decrypting with an individual recovery key

To unlock: *diskutil corestorage unlockVolume UUID -passphrase recoverykey*

To decrypt: *diskutil corestorage revert UUID -passphrase recoverykey*

A screenshot of a macOS terminal window titled "rtrouton — bash — 144x9". The terminal shows the following output for the command `diskutil corestorage unlockVolume C6CAE4D4-DBFB-42A5-981D-8623C98DE7A7 -passphrase AC4M-H920-TEAY-KR2F-HLVT-8H5N`:

```
troutont-lion-lm1:~ rtrouton$ diskutil corestorage unlockVolume C6CAE4D4-DBFB-42A5-981D-8623C98DE7A7 -passphrase AC4M-H920-TEAY-KR2F-HLVT-8H5N
Started CoreStorage operation
Logical Volume successfully unlocked
Logical Volume successfully attached as disk5
Logical Volume successfully mounted as /Volumes/Lion HD
Core Storage disk: disk5
Finished CoreStorage operation
troutont-lion-lm1:~ rtrouton$
```

This command would be used only if FileVault 2 generated the recovery key. This would not apply if you're using FileVaultMaster.keychain as an institutional recovery key.

You can also unlock or decrypt using the individual recovery key using the commands shown.

Recovering from Terminal

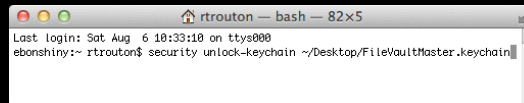
Unlocking or decrypting with a institutional recovery key

First, copy the FileVaultMaster.keychain with the private key in the keychain from its secured place to a convenient place on the Mac.

Next, to allow the Mac to get access to both the keys inside, unlock the FileVaultMaster.keychain.

To unlock FileVaultMaster.keychain:

security unlock-keychain /path/to/FileVaultMaster.keychain



You'll be prompted a password. Please enter the Master Password at this point.

With the institutional recovery key, it's a two step process. First, you'll need to retrieve your keychain that has both your private and public keys from the safe place that you put it in. Second, you'll need to unlock that keychain so that the encrypted Mac has access to both keys.

Recovering from Terminal

Unlocking or decrypting with an institutional recovery key

Once you've unlocked FileVaultMaster.keychain, you can unlock or decrypt the disk.

To unlock:

```
diskutil corestorage unlockVolume UUID -recoveryKeychain /path/to/FileVaultMaster.keychain
```

To decrypt:

```
diskutil corestorage revert UUID -recoveryKeychain /path/to/FileVaultMaster.keychain
```

As long as FileVaultMaster.keychain is unlocked, you should not be prompted for a password.

Once you've unlocked the keychain with both the private and public keys inside, you'll run the commands shown to either unlock or decrypt the disk. Since FileVault 2 now has access to both keys, you shouldn't need to provide any other passwords.

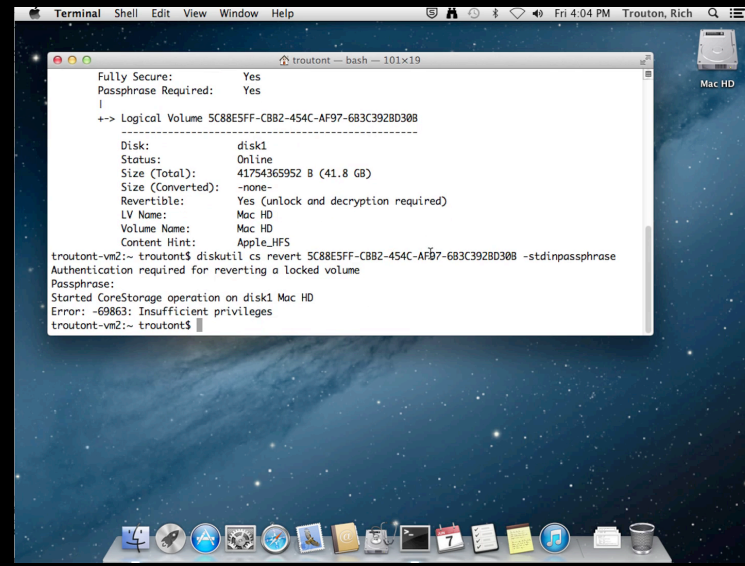


During the demo, the following points will be covered:

1. Booting to Recovery HD
2. Opening Terminal and using the FileVaultMaster.keychain on a separate drive to unlock an encrypted drive.

10.8.4 Decryption Changes

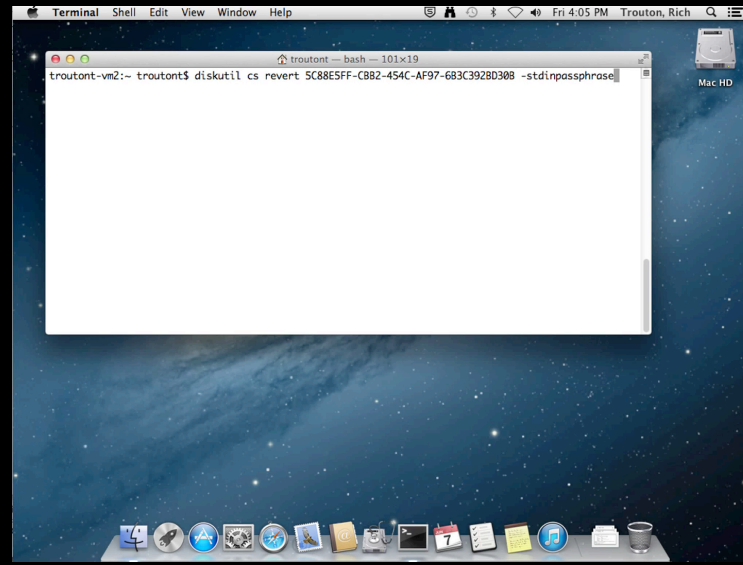
Standard User

A screenshot of a macOS desktop environment. The desktop background is a dark blue space-themed wallpaper. In the center, a terminal window titled 'Terminal' is open, showing a series of commands and their outputs. The user is 'troutont' and the shell is 'bash'. The terminal displays the output of 'diskutil cs revert' for a specific logical volume, which includes details about the disk, its status, size, and reversion requirements. It then prompts for a passphrase, which is entered as 'stdin:passphrase'. Finally, it shows an error message: 'Error: -69863: Insufficient privileges'. The dock at the bottom contains various application icons, and the top status bar shows the time as 'Fri 4:04 PM' and the user as 'Trouton, Rich'.

Starting in 10.8.4, Apple changed diskutil so that it now requires an administrator password before you can decrypt a FileVault 2-encrypted boot drive while booted from it. Here's what happens when a standard user tries to decrypt.

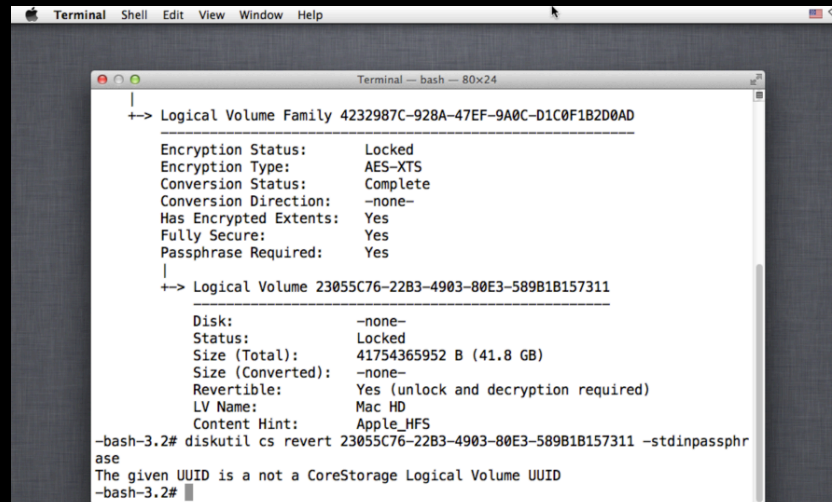
10.8.4 Decryption Changes

Admin User



Here's what happens when a admin user tries to decrypt.

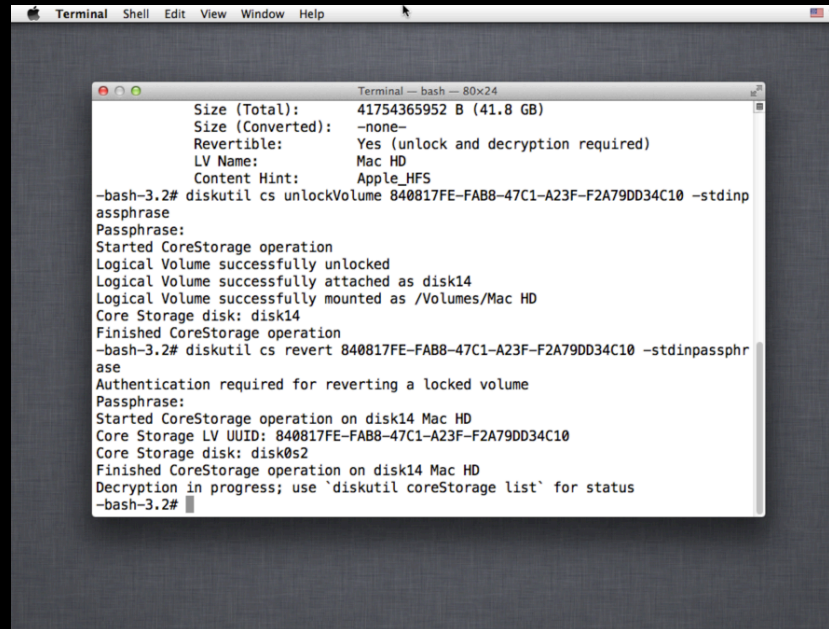
10.8.4 Decryption Changes



```
Terminal  Shell  Edit  View  Window  Help
Terminal — bash — 80x24
↔ Logical Volume Family 4232987C-928A-47EF-9A0C-D1C0F1B2D0AD
Encryption Status:      Locked
Encryption Type:        AES-XTS
Conversion Status:      Complete
Conversion Direction:   -none-
Has Encrypted Extents:  Yes
Fully Secure:           Yes
Passphrase Required:    Yes
|
↔ Logical Volume 23055C76-22B3-4903-80E3-589B1B157311
Disk:                    -none-
Status:                  Locked
Size (Total):            41754365952 B (41.8 GB)
Size (Converted):        -none-
Revertible:              Yes (unlock and decryption required)
LV Name:                 Mac HD
Content Hint:            Apple_HFS
-bash-3.2# diskutil cs revert 23055C76-22B3-4903-80E3-589B1B157311 -stdinpassphr
ase
The given UUID is a not a CoreStorage Logical Volume UUID
-bash-3.2#
```

There were some side-effects to this change. The most notable was that you can no longer just decrypt a machine while booted from Recovery HD. If you try, you will get a confusing error.

10.8.4 Decryption Changes



```
Terminal — bash — 80x24
Size (Total):      41754365952 B (41.8 GB)
Size (Converted):  -none-
Revertible:        Yes (unlock and decryption required)
LV Name:           Mac HD
Content Hint:      Apple_HFS
-bash-3.2# diskutil cs unlockVolume 840817FE-FAB8-47C1-A23F-F2A79DD34C10 -stdinpassphrase
Passphrase:
Started CoreStorage operation
Logical Volume successfully unlocked
Logical Volume successfully attached as disk14
Logical Volume successfully mounted as /Volumes/Mac HD
Core Storage disk: disk14
Finished CoreStorage operation
-bash-3.2# diskutil cs revert 840817FE-FAB8-47C1-A23F-F2A79DD34C10 -stdinpassphrase
Authentication required for reverting a locked volume
Passphrase:
Started CoreStorage operation on disk14 Mac HD
Core Storage LV UUID: 840817FE-FAB8-47C1-A23F-F2A79DD34C10
Core Storage disk: disk0s2
Finished CoreStorage operation on disk14 Mac HD
Decryption in progress; use `diskutil coreStorage list` for status
-bash-3.2#
```

The answer here is that Apple now requires that the encrypted volume be unlocked first. Once it's unlocked, then you can decrypt.



With the exception of that last bit about decryption, everything I've talked about up to this point has applied equally to Lion and later versions of OS X. Before we dive into fdsetup on Mountain Lion and later, let's take a look at what FileVault 2 in 10.7 does not have.

You can monitor, unlock or decrypt a FileVault 2–encrypted boot drive using command line tools, but you can't start the encryption process from the command line using Apple's native tools. Instead, the encryption needs to be enabled from System Preference's FileVault preference pane.

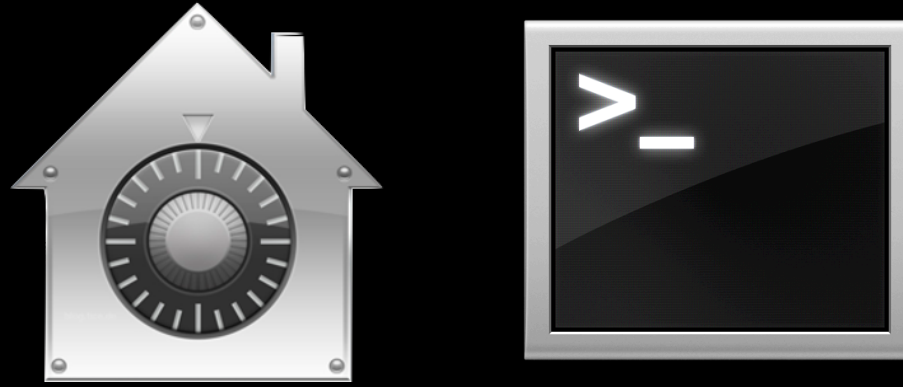
It is not possible to see who has FileVault 2–enabled accounts without looking at the pre–boot login screen.

It can be difficult to add an account to the list of enabled accounts without using the FileVault preference pane.

It is not possible to remove an account from the list of enabled accounts without deleting the account or setting the account password to be blank.

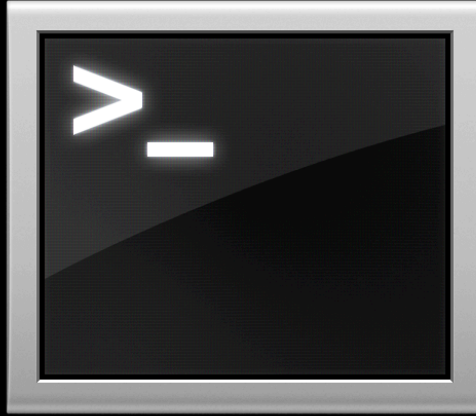
You have to choose between using the individual alphanumeric recovery key or using the institutional recovery key using FileVaultMaster.keychain.

fdsetup overview



fdsetup allows FileVault 2 administration from the command line and solves all of those problems with its various functions. It will turn on FileVault 2 encryption using a variety of options, disable encryption, allow addition and removal of FileVault 2 enabled users from the command line, supply a current list of authorized users, provide encryption status and much more.

fdsetup commands



- › fdsetup enable
- › fdsetup disable
- › fdsetup add
- › fdsetup list
- › fdsetup remove
- › fdsetup sync

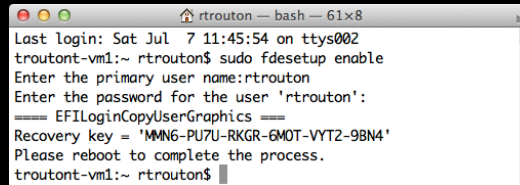
fdsetup has a number of verbs associated with it. The ones that may be most commonly used are enable, disable, add, list, remove and sync.

fdsetup enable

- Activates FileVault 2 Encryption
 - Can set FileVault 2 encryption to use:
 - Individual alphanumeric recovery key
 - Institutional recovery key using FileVaultMaster.keychain
 - Both kinds of recovery key simultaneously
 - Can enable multiple user accounts at time of encryption activation
 - Can import user and certificate information

fdsetup is amazingly flexible when it comes to enabling FileVault 2 encryption from the command-line.

fdsetup enable

A terminal window titled 'rtrouton — bash — 61x8' showing the execution of the 'fdsetup enable' command. The output includes the last login time, prompts for the primary user name and password, the EFI login copy user graphics, a recovery key, and a prompt to reboot.

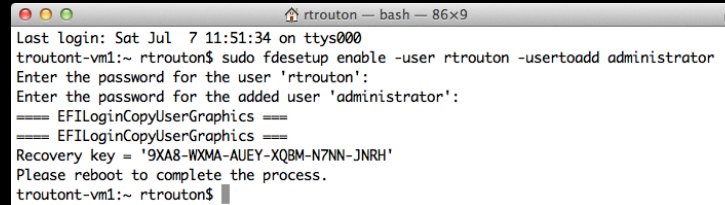
```
rtrouton — bash — 61x8
Last login: Sat Jul 7 11:45:54 on ttys002
troutont-vm1:~ rtrouton$ sudo fdsetup enable
Enter the primary user name:rtrouton
Enter the password for the user 'rtrouton':
===== EFILoginCopyUserGraphics =====
Recovery key = 'MMN6-PU7U-RKGR-GMOT-VYTZ-9BN4'
Please reboot to complete the process.
troutont-vm1:~ rtrouton$
```

fdsetup enable

To start with the simplest method, you would run the command shown on the screen to enable FileVault 2 encryption. Next, you'll be prompted for the username and password of the primary user, which is the account you want to have appear at the FileVault 2 pre-boot login screen once the encryption is turned on. If everything's working properly, you'll next be given an alphanumeric individual recovery key and prompted to restart.

One thing that's very important to know is that the individual recovery key is not saved anywhere. You will need to make a record of it when it's displayed or you will not have it later.

fdsetup enable

A terminal window titled 'rtrouton - bash - 86x9' showing the execution of the 'fdsetup enable' command. The output includes the last login time, the command being run, prompts for passwords for 'rtrouton' and 'administrator', progress bars for EFI login copy, a recovery key, and a reboot prompt.

```
rtrouton - bash - 86x9
Last login: Sat Jul  7 11:51:34 on ttys000
troutont-vm1:~ rtrouton$ sudo fdsetup enable -user rtrouton -usertoadd administrator
Enter the password for the user 'rtrouton':
Enter the password for the added user 'administrator':
==== EFILoginCopyUserGraphics ====
==== EFILoginCopyUserGraphics ====
Recovery key = '9XA8-WXMA-AUEY-XQBM-N7NN-JNRH'
Please reboot to complete the process.
troutont-vm1:~ rtrouton$
```

fdsetup enable -user username -usertoadd username

You can also enable additional user accounts at the time of encryption, as long as the accounts are either local or mobile network accounts. You would run the command as shown on the screen and specify the accounts you want. As part of this, you will be prompted for the account passwords.

After that, you'll be given an individual recovery key and prompted to restart. All of the accounts specified should appear at the FileVault 2 pre-boot login screen.

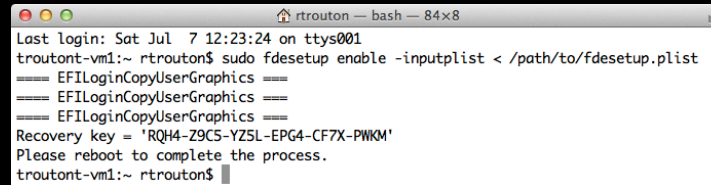
fdsetup enable

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
  <dict>
    <key>Username</key>
    <string>localadmin</string>
    <key>Password</key>
    <string>password</string>
    <key>AdditionalUsers</key>
    <array>
      <dict>
        <key>Username</key>
        <string>tom</string>
        <key>Password</key>
        <string>password</string>
      </dict>
      <dict>
        <key>Username</key>
        <string>harry</string>
        <key>Password</key>
        <string>password</string>
      </dict>
    </array>
  </dict>
</plist>
```

Note: All account passwords need to be supplied in cleartext.

For those who want to automate the process, fdsetup also supports importing a property list file via standard input (stdin). The plist file needs to follow the format shown up on the screen and more users can be added by appending their information under the AdditionalUsers plist key.

fdsetup enable

A terminal window titled 'rtrouton - bash - 84x8' showing the execution of the 'fdsetup enable' command. The output includes the last login time, the command being run, three lines of 'EFILoginCopyUserGraphics' progress bars, a recovery key, and a prompt to reboot.

```
rtrouton - bash - 84x8
Last login: Sat Jul 7 12:23:24 on ttys001
troutont-vm1:~ rtrouton$ sudo fdsetup enable -inputplist < /path/to/fdsetup.plist
==== EFILoginCopyUserGraphics ====
==== EFILoginCopyUserGraphics ====
==== EFILoginCopyUserGraphics ====
Recovery key = 'RQH4-Z9C5-YZ5L-EPG4-CF7X-PWKM'
Please reboot to complete the process.
troutont-vm1:~ rtrouton$
```

fdsetup enable -inputplist < plistfile.plist

Once the plist has been set up, you would run the command shown on the screen to enable FileVault 2 encryption and reference the information in the plist file.

Since the accounts and passwords are in the plist file, fdsetup does not need to prompt for passwords. Instead, the individual recovery key is displayed and the user is prompted to restart. All of the accounts specified in the plist file should appear at the FileVault 2 pre-boot login screen.

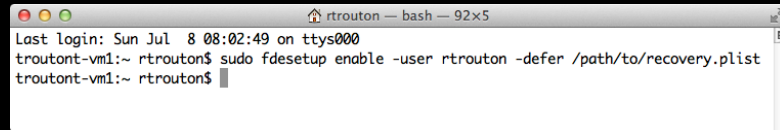
fdsetup enable

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>EnabledDate</key>
  <string>2013-08-25 21:17:26 -0400</string>
  <key>EnabledUser</key>
  <string>username</string>
  <key>HardwareUUID</key>
  <string>00000000-0000-1000-8000-000C29B31139</string>
  <key>LVGUID</key>
  <string>5A28D2AA-126E-4C1D-A7D6-78FA57868B3A</string>
  <key>LVUUID</key>
  <string>CC1349A8-ACD6-4FF8-BCEA-7C607D0A45EA</string>
  <key>PVUUID</key>
  <string>82128581-777F-411F-84C3-8C11A35D8803</string>
  <key>RecoveryKey</key>
  <string>RKT4-CAZU-D76B-XR9Q-OXXU-TT88</string>
  <key>SerialNumber</key>
  <string>VMWk3cJE4T1G1L1nyken7/kg</string>
</dict>
</plist>
```

To avoid the need to enter a password, fdsetup also has a defer flag that can be used with the enable verb to delay enabling FileVault 2 until after the user logs out. With the defer flag, the user will be prompted for their password at their next logout. The recovery key information is not generated until the user password is obtained, so the defer option requires a file location where this information will be written to as a plist file.

The plist file will be created as a root-only readable file and contain information similar to what's shown on the screen. For security reasons, this plist file should not stay on the encrypted system. It should be copied to a safe location and then securely deleted from the system.

fdsetup enable

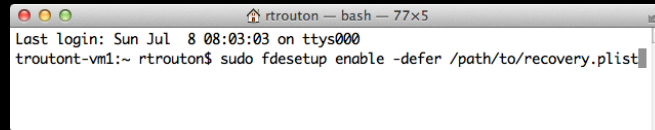
A terminal window titled 'rtrouton — bash — 92x5'. The window shows the following text: 'Last login: Sun Jul 8 08:02:49 on ttys000', 'troutont-vm1:~ rtrouton\$ sudo fdsetup enable -user rtrouton -defer /path/to/recovery.plist', and 'troutont-vm1:~ rtrouton\$' with a cursor at the end.

```
rtrouton — bash — 92x5
Last login: Sun Jul 8 08:02:49 on ttys000
troutont-vm1:~ rtrouton$ sudo fdsetup enable -user rtrouton -defer /path/to/recovery.plist
troutont-vm1:~ rtrouton$
```

fdsetup enable -user username -defer plistfile.plist

If you have a particular user account that you want to enable, you would run the command shown on the screen to defer enabling FileVault 2 and specify the account you want.

fdsetup enable

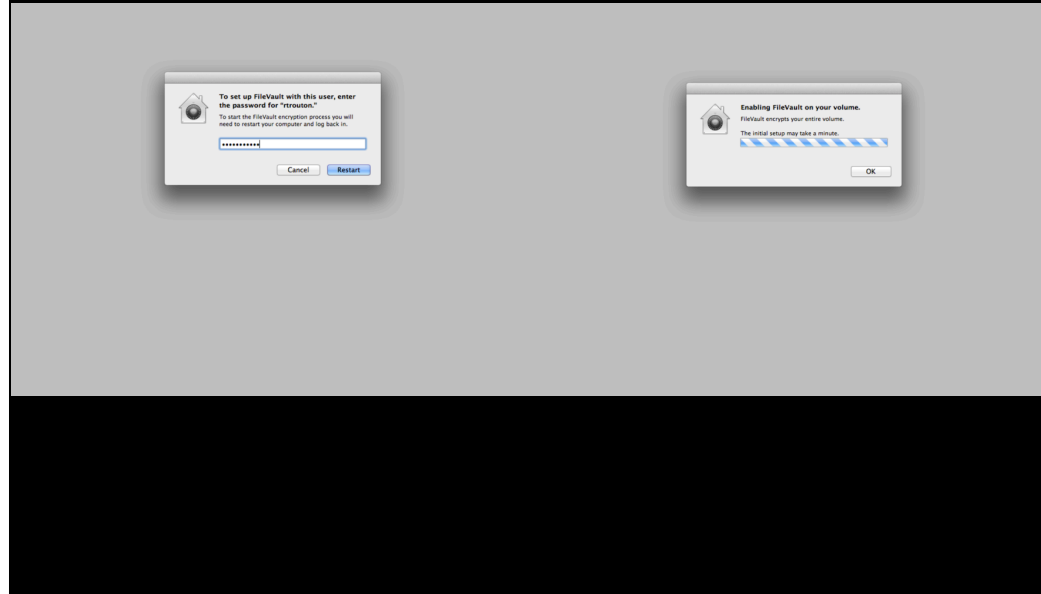
A screenshot of a terminal window with a title bar that reads 'rtrouton - bash - 77x5'. The terminal content shows 'Last login: Sun Jul 8 08:03:03 on ttys000' followed by the command 'troutont-vm1:~ rtrouton\$ sudo fdsetup enable -defer /path/to/recovery.plist' with a cursor at the end of the line.

```
rtrouton - bash - 77x5
Last login: Sun Jul 8 08:03:03 on ttys000
troutont-vm1:~ rtrouton$ sudo fdsetup enable -defer /path/to/recovery.plist
```

fdsetup enable -defer plistfile.plist

If you don't want to specify the account, you would use the command shown on the screen. If there is no account specified, then the current logged-in user will be enabled for FileVault 2. If there is no user specified and no users are logged in when the command is run, then the next user that logs in will be chosen and enabled.

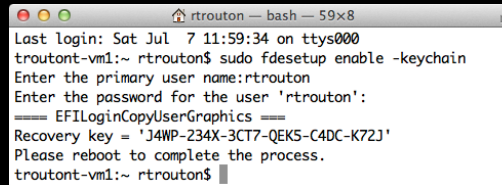
fdsetup enable



On logout, the user will be prompted to enter their account password. Once entered, FileVault 2 will be enabled and the recovery information plist file will be created. Once the enabling process is complete, the Mac will restart.

An important thing to keep in mind about the defer option is that it enables one single user account at the time of turning on FileVault 2 encryption. The defer option does not enable multiple user accounts and cannot be used to enable accounts once FileVault 2 encryption has been turned on.

fdsetup enable

A terminal window titled 'rtrouton — bash — 59x8' showing the execution of the 'fdsetup enable -keychain' command. The output includes the last login time, the command being run, prompts for user name and password, and the display of a recovery key.

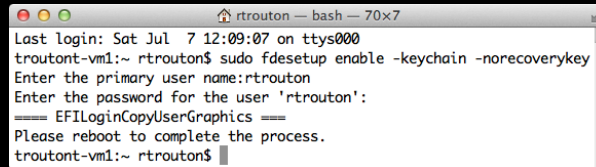
```
rtrouton — bash — 59x8
Last login: Sat Jul 7 11:59:34 on ttys000
troutont-vm1:~ rtrouton$ sudo fdsetup enable -keychain
Enter the primary user name:rtrouton
Enter the password for the user 'rtrouton':
===== EFILoginCopyUserGraphics =====
Recovery key = 'J4WP-234X-3CT7-QEK5-C4DC-K72J'
Please reboot to complete the process.
troutont-vm1:~ rtrouton$
```

fdsetup enable -keychain

Another capability of FileVault 2 in Mountain Lion and later is the ability to use the alphanumeric individual recovery key, an institutional recovery key using FileVaultMaster.keychain, or both kinds of recovery key at the same time.

As seen in the earlier examples, fdsetup will provide the individual recovery key by default. To use the institutional recovery key, the `-keychain` flag needs to be used as shown on the screen. The individual recovery key is displayed, but the encryption will also use the FileVaultMaster.keychain institutional recovery key. In case recovery is needed, either recovery key will work to unlock or decrypt the encrypted drive.

fdsetup enable

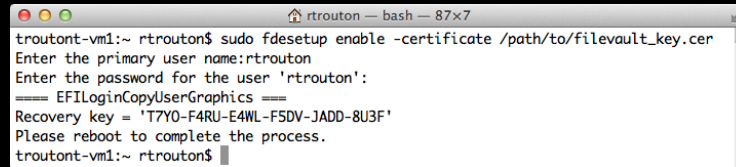
A terminal window titled 'rtrouton — bash — 70x7' showing the execution of the 'fdsetup enable' command. The output indicates the last login time, the command being run, and prompts for the primary user name and password. It also shows the EFI login copy user graphics and a message to reboot to complete the process.

```
rtrouton — bash — 70x7
Last login: Sat Jul 7 12:09:07 on ttys000
troutont-vm1:~ rtrouton$ sudo fdsetup enable -keychain -norecoverykey
Enter the primary user name:rtrouton
Enter the password for the user 'rtrouton':
==== EFILoginCopyUserGraphics ====
Please reboot to complete the process.
troutont-vm1:~ rtrouton$
```

fdsetup enable -keychain -norecoverykey

If you want to specify that only the FileVaultMaster keychain be used, both the `-keychain` and `-norecoverykey` flags need to be used when enabling encryption

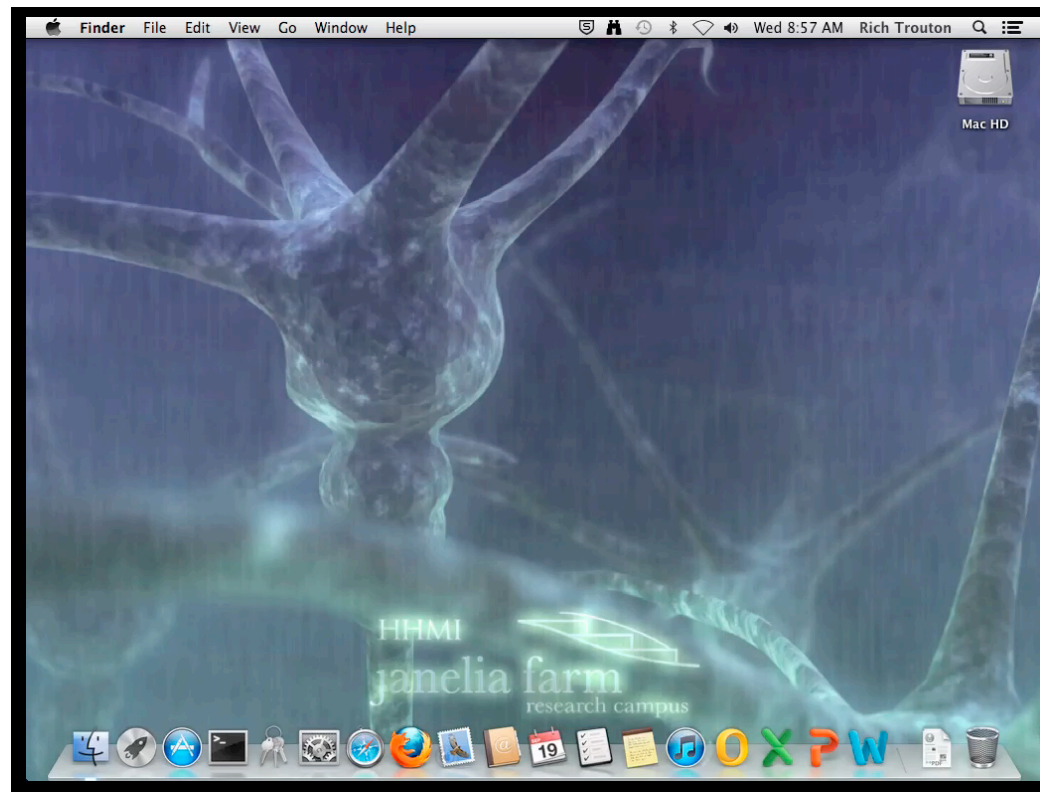
fdsetup enable

A terminal window titled 'rtrouton — bash — 87x7' showing the execution of the 'fdsetup enable' command. The command is run with the '-certificate' option pointing to a file path. The terminal prompts for the primary user name and password. It then displays the EFI login copy user graphics, a recovery key, and a message to reboot to complete the process.

```
troutont-vm1:~ rtrouton$ sudo fdsetup enable -certificate /path/to/filevault_key.cer
Enter the primary user name:rtrouton
Enter the password for the user 'rtrouton':
==== EFILoginCopyUserGraphics ====
Recovery key = 'T7Y0-F4RU-E4WL-FSDV-JADD-8U3F'
Please reboot to complete the process.
troutont-vm1:~ rtrouton$
```

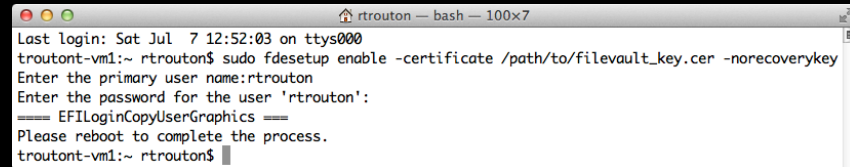
fdsetup enable -certificate cert.cer

fdsetup is also capable of creating a FileVaultMaster keychain and automatically storing it in /Library/Keychains. To do this, an existing FileVault 2 public key needs to be available as a DER encoded certificate file. Once that's available, the command shown on the screen will enable FileVault 2, automatically create the institutional recovery key with the supplied certificate file and store it as /Library/Keychains/FileVaultMaster.keychain



Let's take a look at how you would create a DER encoded certificate file from an existing public key. In this case, we're assuming that there is not a pre-existing recovery key so we'll be creating one with the create FileVaultMaster keychain tool.

fdsetup enable

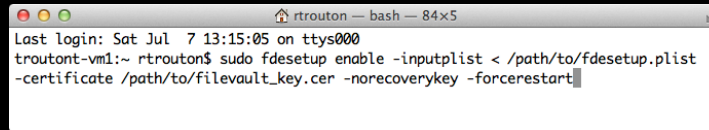
A terminal window titled 'rtrouton — bash — 100x7' showing the execution of the 'fdsetup enable' command. The output indicates the command was successful and prompts for a reboot.

```
rtrouton — bash — 100x7
Last login: Sat Jul 7 12:52:03 on ttys000
troutont-vm1:~ rtrouton$ sudo fdsetup enable -certificate /path/to/filevault_key.cer -norecoverykey
Enter the primary user name:rtrouton
Enter the password for the user 'rtrouton':
==== EFILoginCopyUserGraphics ====
Please reboot to complete the process.
troutont-vm1:~ rtrouton$
```

fdsetup enable -certificate cert.cer -norecoverykey

To specify that only the FileVaultMaster keychain be used as the recovery key, you would add the `norecoverykey` flag to the command.

fdsetup enable

A terminal window titled 'rtrouton - bash - 84x5' showing the command 'fdsetup enable' being executed with various options. The command is: 'fdsetup enable -inputplist < /path/to/fdsetup.plist -certificate /path/to/filevault_key.cer -norecoverykey -forcerestart'. The output shows the last login time and the successful execution of the command.

```
rtrouton - bash - 84x5
Last login: Sat Jul 7 13:15:05 on ttys000
trouton-vm1:~ rtrouton$ sudo fdsetup enable -inputplist < /path/to/fdsetup.plist
-certificate /path/to/filevault_key.cer -norecoverykey -forcerestart
```

**fdsetup enable -inputplist < plistfile.plist-certificate
cert.cer -norecoverykey
-forcerestart**

Along with the various options for enabling, it's also possible to force a restart of the Mac once FileVault 2 has been successfully configured. This can help automate the process of enabling FileVault 2 on a Mac if no input from a logged-in user is needed.

For example, an organization may want to pre-configure its Macs to automatically encrypt with FileVault 2 at first boot with a local admin account enabled. It also wants to use only the institutional recovery key. If a plist with the desired account information and a certificate file to create the institutional recovery key is available, the command shown on the screen could be run to enable FileVault 2 and force a restart at the first boot.



Since this combines three different enable options, let's take a look at how it works when you run that command to automatically encrypt. In this case, I'm going to be enabling three accounts via a plist file and setting the institutional key as the sole recovery key.

fdsetup disable

Disables FileVault 2 Encryption

In contrast to all of the various options available for enabling FileVault 2 using fdsetup, the command to turn off FileVault 2 encryption is fdsetup disable. There are no additional flags associated with this command.

fdsetup add

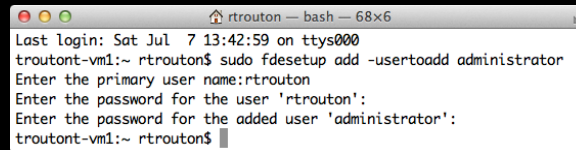
- Enables additional accounts after FileVault 2 encryption is complete
 - Can enable multiple user accounts
 - Can import user information

Once the Mac has been fully encrypted with FileVault 2, you can add additional users using `fdsetup`. To do so, you will need to provide both the username and password of either a previously enabled account or an admin account, as well as the password of the account you want to add.

There's something that's interesting to know about this method: the admin user in question does not themselves need to be enabled for FileVault 2. In my testing, I found that an admin user can authorize the enabling of other accounts even if the admin account wasn't enabled. An admin account can also enable itself using this process, by being both the authorizing admin account and the account being enabled. This is similar to the System Preferences behavior, where an admin account could enable itself by logging in and clicking the lock in the FileVault preference pane.

Since a key has to be involved somewhere, I've got an inquiry open with Apple as to why this works but I haven't heard back yet.

fdsetup add

A terminal window titled 'rtrouton — bash — 68x6' showing the execution of the 'fdsetup add' command. The output shows the last login time, the command being run, and prompts for the primary user name and passwords for both the existing user and the new user.

```
rtrouton — bash — 68x6
Last login: Sat Jul 7 13:42:59 on ttys000
troutont-vm1:~ rtrouton$ sudo fdsetup add -usertoadd administrator
Enter the primary user name:rtrouton
Enter the password for the user 'rtrouton':
Enter the password for the added user 'administrator':
troutont-vm1:~ rtrouton$
```

fdsetup add -usertoadd username

The command shown on the screen will enable a specified user on this encrypted Mac. The primary user can be any account on the Mac that's already been enabled for use with FileVault 2, or any account with admin privileges.

fdsetup add

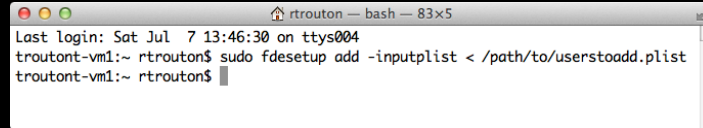
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
  <dict>
    <key>Username</key>
    <string>rtrouton</string>
    <key>Password</key>
    <string>password</string>
    <key>AdditionalUsers</key>
    <array>
      <dict>
        <key>Username</key>
        <string>fcheeryble</string>
        <key>Password</key>
        <string>password</string>
      </dict>
      <dict>
        <key>Username</key>
        <string>nnickleby</string>
        <key>Password</key>
        <string>password</string>
      </dict>
    </array>
  </dict>
</plist>
```

Note: All account passwords need to be supplied in cleartext.

For those who want to automate the process, fdsetup also supports importing a plist file via standard input (stdin). The plist needs to follow the format shown up on the screen.

When adding additional users using a plist file, the top level Username key is ignored, and the Password key value should be an admin account's password. More users can be added by appending their information under the AdditionalUsers plist key.

fdsetup add

A terminal window titled 'rtrouton - bash - 83x5' showing the execution of the 'fdsetup add' command. The window displays the last login time, the command being run, and the prompt for the next command.

```
rtrouton - bash - 83x5
Last login: Sat Jul 7 13:46:30 on ttys004
troutont-vm1:~ rtrouton$ sudo fdsetup add -inputplist < /path/to/userstoadd.plist
troutont-vm1:~ rtrouton$
```

fdsetup add -inputplist /path/to/plistname.plist

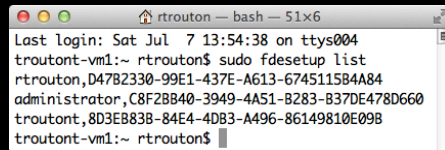
Once the plist has been set up, you can run the command shown on the screen to add additional users by referencing the account information in the plist file.

fdsetup list

- Displays enabled accounts
 - List includes the accounts' usernames and UUIDs

To list all accounts enabled for FileVault 2, fdsetup includes the list verb.

fdsetup list

A terminal window titled 'rtrouton — bash — 51x6' showing the output of the 'fdsetup list' command. The output lists three accounts: 'rtrouton', 'administrator', and 'troutont', each with their respective UUIDs.

```
rtrouton — bash — 51x6
Last login: Sat Jul 7 13:54:38 on ttys004
troutont-vm1:~ rtrouton$ sudo fdsetup list
rtrouton,D4782330-99E1-437E-A613-674511584A84
administrator,C8F28B40-3949-4A51-B283-B37DE478D660
troutont,8D3EB838-84E4-4DB3-A496-86149810E098
troutont-vm1:~ rtrouton$
```

fdsetup list

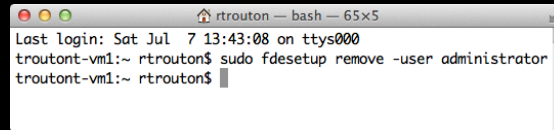
To get a list of all FileVault 2 enabled accounts on your Mac, you would run the command shown on the screen. All enabled accounts will be listed with both the accounts' username and UUID.

fdsetup remove

- Removes accounts from the list of FileVault 2 enabled accounts
 - Can disable using account username
 - Can disable using account UUID

To remove accounts from the list of FileVault 2 enabled accounts, fdsetup includes the remove verb. You can remove users by using either the username or the account's UUID.

fdsetup remove

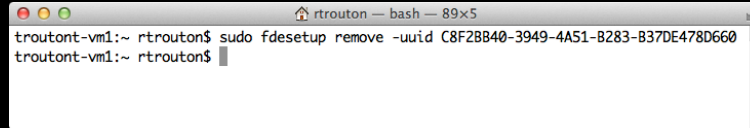
A terminal window titled 'rtrouton - bash - 65x5' showing the command 'fdsetup remove -user administrator' being executed. The output shows the last login time and the command being run.

```
rtrouton - bash - 65x5
Last login: Sat Jul 7 13:43:08 on ttys000
troutont-vm1:~ rtrouton$ sudo fdsetup remove -user administrator
troutont-vm1:~ rtrouton$
```

fdsetup remove -user username

To remove the account by username, you would run the command as shown on the screen and provide the account's username.

fdsetup remove

A terminal window titled 'rtrouton - bash - 89x5' showing the command 'fdsetup remove -uuid C8F2BB40-3949-4A51-B283-B37DE478D660' being executed with 'sudo' in a 'troutont-vm1' environment.

```
troutont-vm1:~ rtrouton$ sudo fdsetup remove -uuid C8F2BB40-3949-4A51-B283-B37DE478D660
troutont-vm1:~ rtrouton$
```

fdsetup remove -uuid uuid_here

To remove the account using the UUID, you would run the command as shown on the screen and provide the account's UUID.

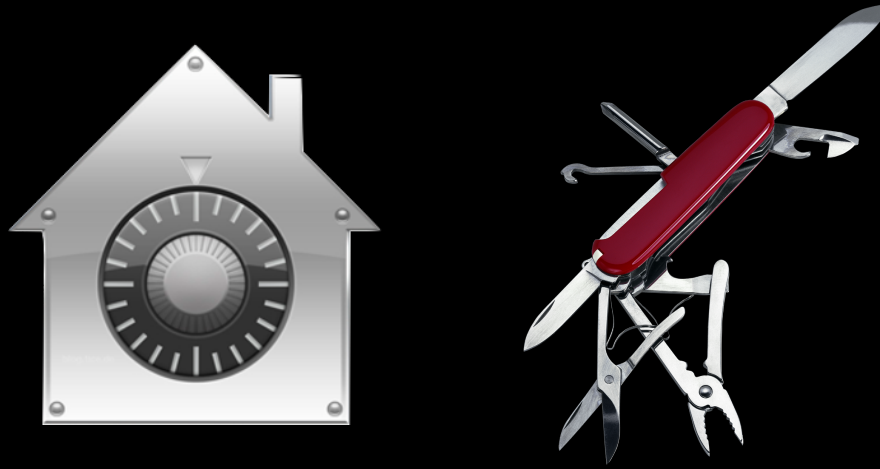
fdsetup sync

- Compares directory service account information with Mac's list of FileVault 2 enabled accounts
 - Removes users that have been removed from the directory service
 - Does not add directory service accounts to list of FileVault 2 enabled accounts

fdsetup also has the sync verb, which allows FileVault 2 to check with the Mac's directory service and see which accounts have been changed. Its main use currently is to automate the disabling of FileVault 2-enabled accounts by checking the directory service to see which accounts have been removed. If an account has been removed from the directory service, running fdsetup sync on an encrypted Mac will automatically remove the account from the list of FileVault 2 enabled accounts. The sync only affects the account's FileVault 2 status and will not remove the account or account home folder from the Mac.

One important thing to know is that sync does not allow accounts to be automatically added, only removed.

fdesetup = FileVault 2 Multitool








fdesetup is a Swiss Army knife for managing FileVault 2 on Mountain Lion and later. It can enable FileVault 2, add and remove users, report on FileVault 2's status and more. If you're managing FileVault 2 encryption in your own environment, I recommend using this tool. Properly used, it will save you time and give encryption options available with no other software.

FileVault 2 Enterprise Management Solutions



There are a number of FileVault 2 management solutions for the enterprise, available from JAMF Software, Dell and open source projects.

	Name	FileVault 2 management	Recovery Key Support	Vendor
	Cauliflower Vest	10.7.x and 10.8.x	Individual	Open Source
	The Casper Suite	10.8.x	Individual and Institutional	JAMF Software
	Dell Credant Enterprise Edition for Mac	10.7.x and 10.8.x	Institutional	Dell
	Crypt	10.7.x and 10.8.x	Individual	Open Source
	FileVault Setup.app	10.8.x	Individual and Institutional	Open Source

All have their strengths, so evaluate them carefully to find the one that meets your needs. Describing each tools' capabilities would be its own session as long as this one, so I recommend that you take a look at the YouTube videos I link to later in the talk, as I've previously talked in detail on both Cauliflower Vest and on the Casper Suite's FileVault 2 management capabilities.

For those interested, the Cauliflower Vest information was given as part of a longer FileVault 2 session at the 2012 Penn State MacAdmins conference. The Casper Suite's ability to manage FileVault 2 was covered at the 2012 JAMF National User Conference and also at the 2013 Penn State MacAdmins conference.

FileVault Setup.app



Since I haven't previously spoken about it, I want to take some time to talk about an open source project called FileVault Setup.app. This is an application that is designed to be a user-friendly interface for Apple's `fdesetup` tool and supports turning on FileVault 2 encryption and enabling a single user account.

One great thing about this tool from my perspective is that it's designed to be independent of any server-based resources. To the best of my knowledge, this is the first tool that allows FileVault encryption to be enforced on a machine entirely from the machine's local resources.

FileVault Setup.app

- Allows FileVault 2 encryption to be force-enabled on a Mac
- Enforcement requires a loginhook
- Can be configured to enable fdesetup in a variety of ways
- By default, uses both individual and institutional keys

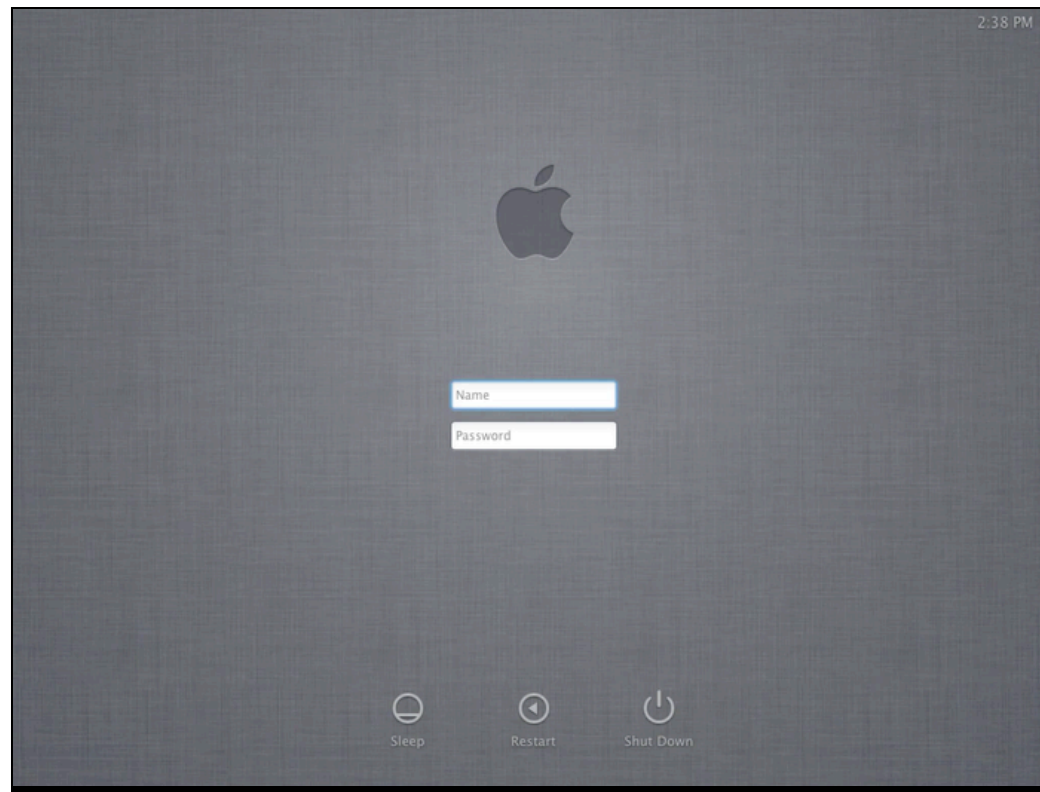
The application was designed to be run by a Mac OS X loginhook. This allows it to be launched when a user logs in, but also runs the application with root privileges. Running this application with root privileges is important because fdesetup requires root privileges to run.

FileVault Setup.app

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>EnabledDate</key>
  <string>2013-04-29 22:17:00 -0400</string>
  <key>HardwareUUID</key>
  <string>00000000-0000-1000-8000-000C29CEF923</string>
  <key>HasMasterKeychain</key>
  <true/>
  <key>LVGUID</key>
  <string>9807169C-24E6-4DDC-975A-71D078D73390</string>
  <key>LVUUID</key>
  <string>2BF1F4CA-5E97-4A6B-820A-A87F1DEA5B1D</string>
  <key>PVUUID</key>
  <string>0B0DE25B-8D24-4E31-B1B0-0831455C3A65</string>
  <key>RecoveryKey</key>
  <string>QFDA-9W5V-K2W3-93MR-Y7T8-DPZ5</string>
  <key>SerialNumber</key>
  <string>VMNVk2F+NYrG/tkLIgnnJaiw</string>
</dict>
</plist>
```

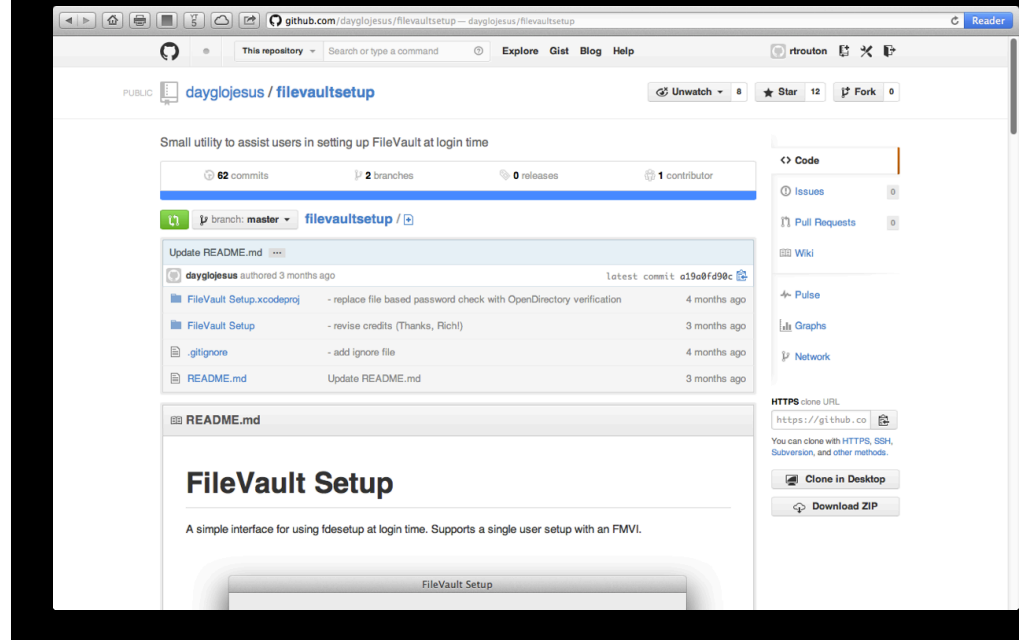
In its default configuration, FileVault Setup will try to set up two recovery keys by using a properly configured /Library/Keychains/FileVaultMaster.keychain as the institutional recovery key and also generate an alphanumeric individual recovery key.

To make sure that the individual recovery key is recorded for later reference, FileVault Setup will generate a plist file similar to what's shown up on the screen. This plist will contain the individual recovery key and is stored in /var/root/ as a file named fdesetup_output.plist. To make sure that your recovery key is not stored on the encrypted system, my recommendation is to plan and implement a mechanism for recovering then removing this information from the system.



Here's how the process looks from the user's perspective. In this case, we're enforcing that FileVault 2 encryption be enabled on a Mac that doesn't have FileVault 2 turned on.

FileVault Setup.app



FileVault Setup.app is available on GitHub. It was written by Brian Warsing, who works for Simon Fraser University in Canada, for use with Simon Fraser's Mac population.

Limitations of FileVault 2

- FileVault 2 is an overall better solution than FileVault 1 is, but it does not necessarily cover all workplaces' requirements for full disk encryption.
 - Can't use remote management tools at the pre-boot login screen
 - Can't use authentication methods other than password/passphrases
 - Pre-boot login screen can't display username / password blanks

FileVault 2 is an overall better solution than FileVault 1 is, but it does not necessarily cover all workplaces' requirements for full disk encryption.

Among the things you cannot currently do with FileVault 2 are the following:

1. Use remote management tools at the pre-boot login screen – All current remote management tools require the operating system to be running. The OS is not running at the pre-boot login screen, so there's no way to run these tools.
2. Use authentication methods other than password/passphrases – At this time, the EFI boot environment does not support the use of encryption tokens such as smart cards or USB encryption dongles to unlock FileVault 2's encryption.
3. You cannot set the FileVault pre-boot login screen to display username and password blanks. It only allows for the account icons.

FileVault 2 and the Law

- For folks who need to satisfy regulatory requirements for encryption:
 - FileVault 2 is FIPS 140-2 Compliant for Mountain Lion.
 - FileVault 2's underlying low level encryption uses Apple's new Common Crypto implementation
 - It is Apple's intention to continue its FIPS validation for shipping products.

For those folks who want to use FileVault 2 in a government or other heavily-regulated environment, FileVault 2 on 10.8 is certified as being FIPS 140-2 Compliant by the US Government's FIPS 140-2 encryption standard.

Apple is working on FIPS 140-2 certification for Maverick's Common Crypto cryptography foundation, which would also cover FileVault 2 on 10.9, but the certification process itself can only be begun once Mavericks has been released. At this point, there is no plan to certify Lion's Common Crypto, so Lion's FileVault 2 will not be FIPS validated within the foreseeable future.

For More Information



July 2011 - FileVault Decrypted
August 2011 - FileVault Decrypted for Enterprise
May 2012 - Getting Started with Cauliflower Vest
June 2012 - Cauliflower Vest - Unusual Name, Serious Security
July 2012 - Managing Mountain Lion's FileVault 2 with fdesetup

If you want more information, I recommend checking out the following issues of MacTech. The 2012 issues are available via the MacTech iPad app and print copies of all issues should be available for ordering.

FileVault 2 talk videos

- Penn State MacAdmins 2012 - <http://youtu.be/rw7fcJcmlnI>
- JAMF National User Conference 2012 - <http://www.jamfsoftware.com/news/2012/11/30/video-managing-filevault-2-on-os-x-mountain-lion-with-the-casper-suite/>
- Penn State MacAdmins 2013 - http://youtu.be/fsxtNHj_IY8

As previously mentioned, I had covered Cauliflower Vest and the Casper Suite in earlier talks. The video links for these talks are available here.

Useful Links

- Apple Best Practices for Deploying FileVault 2 - <http://training.apple.com/osx>
- OS X: How to create and deploy a recovery key for FileVault 2 - <http://support.apple.com/kb/HT5077>
- Using a login banner with FileVault 2 - <http://derflounder.wordpress.com/2011/08/04/using-a-login-banner-with-filevault-2/>
- Displaying expiring password notifications when using FileVault 2 with Active Directory accounts - <http://derflounder.wordpress.com/2011/09/12/displaying-expiring-password-notifications-when-using-filevault-2-with-active-directory-accounts/>

Useful Links

- Cauliflower Vest project - <http://code.google.com/p/cauliflowervest/>
- Cauliflower Vest Introduction – <http://code.google.com/p/cauliflowervest/wiki/Introduction>
- Csfde - <http://code.google.com/p/cauliflowervest/wiki/Csfde>
- User Admin - <http://code.google.com/p/cauliflowervest/wiki/UserAdmin>
- Interactive FileVault 2 initialization script (uses csfde)
- <http://derflounder.wordpress.com/2012/03/13/interactive-filevault-2-initialization-script/>

Useful Links

- Administering FileVault 2 on OS X Mountain Lion with the Casper Suite - <http://www.jamfsoftware.com/resources/white-papers>
- Using fdesetup with Mountain Lion's FileVault 2 - <http://derflounder.wordpress.com/2012/07/25/using-fdesetup-with-mountain-lions-filevault-2/>
- Embedding certificate data into a fdesetup plist file - <http://derflounder.wordpress.com/2012/08/22/embedding-certificate-data-into-a-fdesetup-plist-file/>
- Crypt:A FileVault 2 Escrow Solution: <http://grahamgilbert.com/blog/2013/01/18/crypt-a-filevault-2-escrow-solution/>

Useful Links

- FileVault Setup.app – local FileVault 2 encryption setup and enforcement - <http://derflounder.wordpress.com/2013/04/29/filevault-setup-app-local-filevault-2-encryption-setup-and-enforcement/>
- fdesetup authrestart – FileVault 2's one-time encryption bypass feature - <http://derflounder.wordpress.com/2012/09/22/fdesetup-authrestart-filevault-2s-one-time-encryption-bypass-feature/>
- Upgrading your FileVault 2 encrypted Mac to Mountain Lion - <http://derflounder.wordpress.com/2012/07/28/upgrading-your-filevault-2-encrypted-mac-to-mountain-lion/>

Useful Links

- Use an admin password when importing users for FileVault access from a file - <http://support.apple.com/kb/HT5710>
- Enabling FileVault 2 pre-boot login screen functions from the command line - <http://derflounder.wordpress.com/2013/06/19/enabling-filevault-2-pre-boot-login-screen-functions-from-the-command-line/>
- Erasing a FileVault 2-encrypted Volume - <http://derflounder.wordpress.com/2013/06/29/erasing-a-filevault-2-encrypted-volume/>

Downloads

PDF available from the following link:

<http://tinyurl.com/MacSysAd2013PDF>

Keynote slides available from the
following link:

<http://tinyurl.com/MacSysAd2013key>