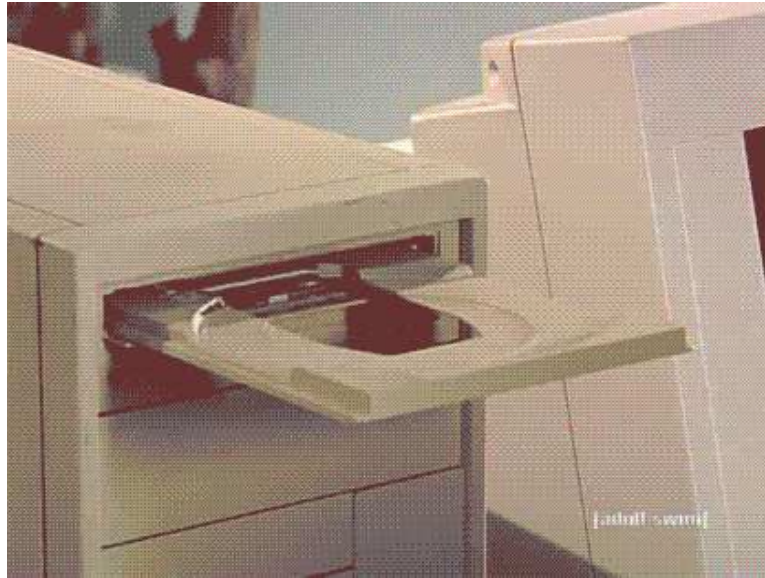


FQA 4 – 9front Installation Guide



4.1 – Pre-installation Checklist

Before you start your install, you should have some idea what you want to end up with. You will want to know the following items, at least:

Machine name, hereafter referred to as `sysname`.

Hardware installed and available. Read: *FQA 3.3 – Selecting Hardware*. Check *FQA 3.2 – Known Working Hardware* as well as the various supported hardware pages on the Bell Labs Plan 9 wiki to help determine if your hardware or VM is supported.

Network settings, if not using DHCP: Gather the requisite information for your network (IP, netmask, default gateway, etc.). If you plan to install in a VM, see *FQA 3.3 – Virtual Machines*.

In case of trouble, see: *FQA 9 – Troubleshooting*, *FQA 2.2.1 – Mailing List Archives*

4.2 – Creating bootable 9front install media

4.2.1 – ISO image

Obtain the `9front.iso`.

Read: *1.11.1 – Mirrors*

4.2.1.1 – Creating on Plan 9

Read: 8.6.2 – Burn a CD-ROM

4.2.2 – USB drive

Booting from USB disk is handled the same way as booting from internal hard drive. There are two ways to create a bootable 9front USB:

1.) Write the ISO image directly to the USB device. Example:

```
cat 9front.iso >/dev/sdUxxxxx/data
```

2.) Alternately, the boot process is able to use an ISO image stored on a FAT file system as its root file system, so all that is needed is to install the 9boot(8) bootloader, a kernel and the 9front.iso on the USB disk. The following sections describe this process on Plan 9 and Linux.

4.2.2.1 – Creating on Plan 9

The path to your USB device will look something like this: /dev/sdUxxxxx where xxxxxx is the unique name of your usb device.

Build and install the mbr and boot loader:

```
cd /sys/src/boot/pc      # 386 = pc, amd64 = pc64, etc.
mk 9bootfat mbr pbs
disk/mbr -m mbr /dev/sdUxxxxx/data
```

If it doesn't already exist, create and format a FAT partition. Don't forget to set the FAT partition active in the fdisk menu:

```
disk/fdisk -b /dev/sdUxxxxx/data      # type '?' for help
disk/format -b pbs -d -r 2 /dev/sdUxxxxx/dos
```

Mount the USB device:

```
dossrv -f /dev/sdUxxxxx/dos sdos
mount -c /srv/sdos /n/dos
```

Create a suitable /n/dos/plan9.ini:

```
bootfile=9pc      # 386 = 9pc, amd64 = 9pc64, etc.
mouseport=ask
monitor=ask
vgasize=ask
```

Note: By convention, Plan 9 usually expects text files to end with a newline. If the last line of plan9.ini is not a newline, it could fail to be parsed correctly at boot time.

Copy files to the USB device:

```
cp /386/9bootfat /n/dos
chmod +al /n/dos/9bootfat # defrag magic
cp /386/9pc /n/dos # 386 = 9pc, amd64 = 9pc64, etc.
cp /path/to/9front.iso /n/dos
```

Optional for EFI systems:

```
cd /sys/src/boot/efi; mk install
mkdir -p /n/dos/efi/boot
cp /386/boot*.efi /n/dos/efi/boot
```

Unmount the USB device:

```
umount /n/dos
rm -f /srv/sdos
```

Boot the device.

Read: `prep(8)`

4.2.2.2 – Creating on Linux

Note: There are numerous methods for creating bootable USB devices on Linux. This is one.

Obtain `mbr`, `pbs`, `9bootfat` and `9pc` binaries. Either copy them from the distributed ISO image (the files are located under `/386/`) or build up to date binaries on a `9front` system. Read: *FQA 5.2.2 – Building from source*

On the Linux system, create a `$dir` that contains the following files:

```
9front.iso
9pc # 386 = 9pc, amd64 = 9pc64, etc.
plan9.ini (as described above)
9bootfat
mbr
pbs
```

Optional for EFI systems, create the directory `boot` in `$dir` and copy the files `bootia32.efi` and `bootx64.efi` into it.

Use the `makebootfat` tool to create bootable device (ie. `/dev/sdc`):

```
$ makebootfat -m $dir/mbr -b $dir/pbs -o /dev/sdc $dir
```

Copy the ISO to the device and then boot the device.

4.2.2.3 – Bootargs

At the `bootargs` prompt you'll need to enter the path to the ISO on the USB device. It will look something like this:

```
bootargs=local!/shr/sdUxxxxx/9front.iso
```

If a USB device is not listed, escape to a shell with `!rc`, and `ls /shr` to find it.

Read: 9.5.1 – Devices not recognized or not working

4.3 – Performing a simple install

There is nothing magical about installing Plan 9. It is simply a matter of populating a Plan 9 file system (`cowfs` or `hjf`) and arranging a bootstrap to eventually load a Plan 9 kernel that can then use that file system as its root.

In most cases, the file server is the only machine that needs to have a disk. Once the initial file server is running, setting up an auth server, and enabling `bootp` and `tftp` for PXE booting, will allow Plan 9 terminals and cpu servers to load kernels from the file server and share its file system over the network.

Note: This guide describes the default installation of a terminal with disk, which is an amalgamation of a normal Plan 9 network, but is sufficient for exploring the basics of using Plan 9. Configuration of additional services (such as a file server with networking listeners, a cpu server, etc.) is outlined in *FQA 7 – System Management*.

To install 9front, boot the `9front.iso` image and follow the steps below.

4.3.1 – boot

Successfully booting the system will result in basic information about the state of the system being printed on screen:

```
Plan 9
126 holes free
00018000 0009f000 552960
00485000 0cf2c000 212496384
213049344 bytes free
cpu0: 3395MHz GenuineIntel P6
ELCR: 0E20
#10: AMD79C970: 10Mbps port 0x2000 irq 10: 000c291d0baf
mylex ctlr @ port 0x10c0: 32-bit wide SCSI host adapter
512M memory: 207M kernel data, 304M user, 929M swap
nusb/usbd: /dev/usb: no hubs
```

4.3.2 – bootargs

A list of attached storage devices is printed, followed by a prompt asking for a Plan 9 partition to boot from:

```
/dev/sdC0: VMware Virtual IDE Hard Drive
/dev/sdC0/data
/dev/sdD0: VMware Virtual IDE CDROM Drive
/dev/sdD0/data 9660
bootargs is (tcp, il, local!device) [local!/dev/sdD0/data]
```

In most cases, the suggested default will correspond to the device used to boot the ISO image.

4.3.3 – user

Next, a prompt asks for a username:

```
user[glenda]:
```

Glenda is the default `hostowner` name. Don't change this until you know what you are doing. Hit `enter` to accept the default.

Note: The `hostowner` differs from the concept of `root` on a UNIX system, where a single user `root` may take control of all processes *and* files on the system. By contrast, even the `hostowner` of a Plan 9 file server cannot violate file permissions on the file system, except when permissions checking is disabled on the console or when entering special commands at the console of the file server. The `hostowner` controls only the *processes* running on the local machine (in the case of the filserver, the file server process itself is obviously owned by the system's `hostowner`). This fundamental separation between control of processes and file permissions is exploited throughout the system, but can be confusing for users coming from a UNIX background.

4.3.4 – vgasize, monitor, mouseport

The next set of prompts deal with graphics display and the mouse:

```
vgasize is (text, 640x480x8, 1024x768x16, ...) [1024x768x16]
monitor is (vesa, xga, lcd, ...) [vesa]
mouseport is (ps2, ps2intellimouse, 0, 1, 2) [ps2]
```

The boot process prompts for the environment variables `$vgasize`, `$monitor` and `$mouseport`, and the installer will later write those values to the system's `plan9.ini`, from which they are loaded on subsequent system bootup.

Setting `$monitor` to anything besides `vesa` will bypass the emulated VESA BIOS and attempt to use a native VGA driver for the video card in question. Read: `/lib/vgadb` for a list of monitors and video cards that are already known by the system; and the man pages `vga(3)`, `vga(8)` and `vgadb(6)` for more information about how graphical displays are configured.

The default `$mouseport` of `ps2` is sufficient in most cases. Set it to `ps2intellimouse` on laptops or for mice that have a scrollwheel.

Note: Some laptops require the trackpad to be disabled in the BIOS in order for mouse button 2 (the center button) to function in Plan 9. In a pinch, mouse button 2 may be simulated by holding down the `shift` key while clicking mouse button 3 (the right button).

After the install, changes intended to persist across reboots should be added to `plan9.ini`.

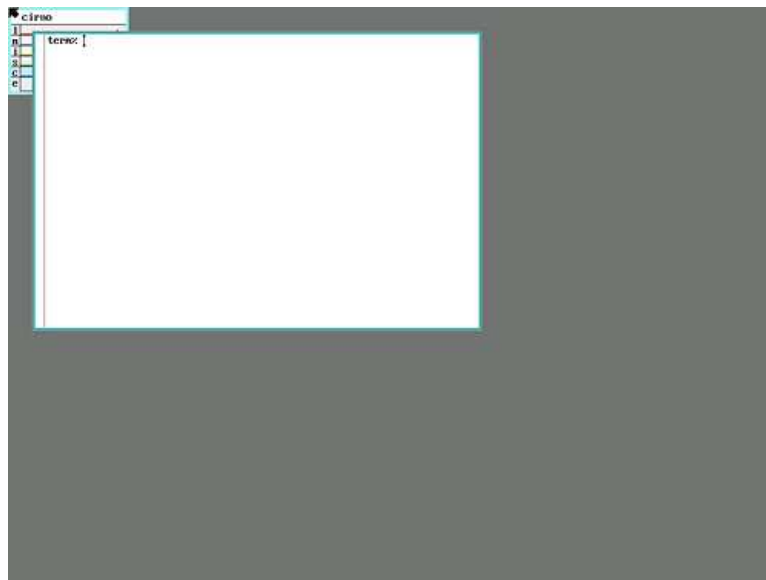
Examples:

Values as they appear in `plan9.ini`:

```
monitor=vesa  
vgasize=1024x768x16  
mouseport=ps2intellimouse
```

Read: mouse(8), vga(3), vgadb(6), plan9.ini(8), realemu(8), vga(8),
FQA 7.2.2 – How do I modify plan9.ini?

After the mouseport prompt is answered, the boot process will attempt to start the Plan 9 graphical environment, rio(1), opening a stats(8) window and a rio window on top of a gray desktop background.



4.3.4.1 – Changing screen resolution

At this point it may be desirable to change the screen resolution. To change video mode from the command line:

vesa:

Note: Only valid modes listed in the VESA BIOS may be used.

```
# obtain a list of vesa bios modes  
@{rfork n; aux/realemu; aux/vga -p}  
# configure one of the valid modes  
@{rfork n; aux/realemu; aux/vga -m vesa -l 1024x768x16}
```

vga:

```
aux/vga -m dellst2210 -l 1920x1080x32
```

4.3.5 – inst/start

Installation is performed by the rc scripts in `/rc/bin/inst`. To begin the installation, type `inst/start` in the terminal window. Follow the prompts to complete the installation, selecting the defaults where appropriate.

Note: Any task may be repeated by manually entering its name at the next `Task to do` prompt.

```
term% inst/start
Tue Jul 17 12:38:50 CET 2012 Installation process started
```

The following `Task to do` steps are handled one at a time:

4.3.6 – configfs

You can install the following types of systems:

```
cwfs64x    the cached-worm file server
hjfs       the new 9front file server (experimental!)
```

```
File system (cwfs64x, hjfs)[cwfs64x]:
```

Note: The `cwfs64x` file server uses 16KB blocks, with its cache and permanent storage locate on separate partitions. The `hjfs` file server uses 4KB blocks, with its cache and permanent storage located on the same partition. If you are installing to a disk of less than 12GB, you should choose `hjfs`.

However: The `hjfs` file system is relatively slow, and performance will degrade due to fragmentation as the disk fills up. *The valuable lesson is that you can get what you want and still not be very happy.* — Pope Cerebus

In this example we will press `enter` to accept `cwfs64x` as the default.

4.3.6.1 – cwfs no-dump configuration

A cheap VPS or an SD card lack the storage capacity for running a usable default `cwfs(4)` setup with a big WORM partition and daily dumps. The go-to solution is the `hjfs(4)` file system, which doesn't use a dedicated WORM partition and doesn't do daily dumps by default. However, it has several performance problems and is not as well tested.

The `cwfs` file system can be configured in lots of ways beyond the default cache-worm + other configuration supported by the 9front installer.

The desired configuration we're going to cover is a single `main` file server tree backed by a simple disk file system – the same type used by the `other` tree in the default setup.

To do this we're going to partition the disk appropriately and override the `mountcwfs` stage of the 9front installer with a replacement script <http://plan9.stanleylieber.com/rc/mountcwfs> that will configure `cwfs` appropriately.

The rest of the installation and most of the subsequent system operation remain unaffected.

Start by booting from the installation media. Configure networking with `ip/ipconfig(8)` and fetch the replacement `mountcwfs` script, or put the script on a flash drive:

```
% webfs
% ramfs
% hget http://plan9.stanleylieber.com/rc/mountcwfs >/tmp/mountcwfs
% chmod +x /tmp/mountcwfs
```

Override the `mountcwfs` stage using `bind(1)`:

```
bind /tmp/mountcwfs /bin/inst/mountcwfs
```

Run `inst/start` and complete the stages up to `preppart` as you would normally. At `preppart`, delete the default partitions and create one named `fsmain` with a desired size:

```
d other
d fscache
d fsworm
a fsmain 123456 .+100%
w
q
```

Make sure to "ream" the new partition at the next step.

Complete the rest of the installation, reboot. That is all.

4.3.7 – partdisk

The following disk devices were found.

```
sdC0 - VMware Virtual IDE Hard Drive
      empty                0 3916          (3916 cylinders, 29.99 GB)

sdD0 - VMware Virtual IDE CDROM Drive
```

Disk to partition (sdC0, sdD0)[no default]:

Enter the media you wish to install to.

```
Disk to partition (sdC0, sdD0)[no default]: sdC0
The disk you selected HAS NO master boot record on its first sector.
(Perhaps it is a completely blank disk.)
Shall we create a blank EFI partition table (GPT)
or install traditional DOS partition table (MBR)?
```

Assuming a blank disk image, install a fresh mbr:


```
Install mbr or gpt (mbr, gpt)[no default]: mbr
```

```
This is disk/fdisk; use it to create a Plan 9 partition.  
If there is enough room, a Plan 9 partition will be  
suggested; you can probably just type 'w' and then 'q'.
```

```
cylinder = 8225280 bytes  
>>>
```

For this example we will use the entire disk. Accept the defaults.

```
>>> w  
>>> q
```

4.3.8 – prepdisk

The following Plan 9 disk partitions were found.

```
/dev/sdC0/plan9  
empty 0 62910477 (62910477 sectors, 29.99 GB)
```

```
Plan 9 partition to subdivide (/dev/sdC0/plan9)[/dev/sdC0/plan9]:
```

Use the Plan 9 partition created in the previous step. Hit enter to select the default.

```
This is disk/prep; use it to subdivide the Plan 9 partition.  
If it is not yet subdivided, a sensible layout will be suggested;  
you can probably just type  
w  
and then  
q.
```

```
no plan9 partition table found  
9fat 204800  
nvram 1  
other 8957953  
fscache 8957953  
fsworm 44789770  
>>>
```

Again, accept the defaults.

```
>>> w  
>>> q
```

4.3.9 – mountfs

The please choose your cwfs64x partitions

Hit enter to select each partition at the prompt.

```
--rw-r----- S 0 glenda glenda 4586471936 Jul  4 13:28 /dev/sdC0/fscache  
Cwfs cache partition (/dev/sdC0/fscache)[/dev/sdC0/fscache]:  
--rw-r----- S 0 glenda glenda 22932362240 Jul  4 13:28 /dev/sdC0/fsworm  
Cwfs worm partition (/dev/sdC0/fsworm)[/dev/sdC0/fsworm]:  
--rw-r----- S 0 glenda glenda 4586471936 Jul  4 13:28 /dev/sdC0/other  
Cwfs other partition (/dev/sdC0/other)[/dev/sdC0/other]:
```

Since this is a fresh install, we choose **yes** to ream (format) the file system:

```
Ream the file system? (yes, no)[no]: yes  
Starting cwfs64x file server for /dev/sdC0/fscache  
Reaming file system  
bad nvram key  
bad authentication id  
bad authentication domain  
nvrcheck: can't read nvram  
config: config: config: auth is now disabled  
config: config: config: config: config: config: current fs is "main"  
cmd_users: cannot access /adm/users  
63-bit cwfs as of Wed Jul  4 00:59:30 2012  
    last boot Tue Jul 17 13:34:57 2012  
Configuring cwfs64x file server for /dev/sdC0/fscache  
Mounting cwfs64x file server for /dev/sdC0/fscache  
% mount -c /srv/cwfs /n/newfs  
Mounting cwfs64x file server for /dev/sdC0/other  
% mount -c /srv/cwfs /n/other other
```

4.3.10 – configdist

Are you going to download the distribution
from the internet or do you have it on local media?

Distribution is from (local, net)[local]:

We are installing from local media, so, `local`. Hit enter to accept the default.

Note: The `net` option attempts to download the ISO from the Internet and install it directly to disk. Since the installer is most likely already running from an ISO image, this option may prove to be of limited utility. The option is somewhat experimental, and may work. Attempt it at your own risk.

[Continued on next page]

4.3.11 – confignet

You can connect to the internet via a local ethernet or a dial-up PPP connection.

Interface to use (ether, ppp)[ether]:

Please choose a method for configuring your ethernet connection.

 manual – specify IP address, network mask, gateway IP address
 dhcp – use DHCP to automatically configure

Configuration method (manual, dhcp)[dhcp]:

4.3.11.1 – dhcp

Hit enter to move on to the next task.

4.3.11.2 – manual

If you chose manual, enter values that are appropriate for your network.

```
Configuration method (manual, dhcp)[dhcp]: manual
ip address [no default]: 10.0.2.15
network mask [no default]: 255.255.255.0
gateway address [no default]: 10.0.2.2
```

4.3.12 – mountdist

Please wait... Scanning storage devices...

```
    /dev/sdC0/9fat
    /dev/sdC0/data
    /dev/sdC0/fscache
    /dev/sdC0/fsworm
    /dev/sdC0/other
    /dev/sdD0/data
```

The following storage media were detected.
Choose the one containing the distribution.

```
    /dev/sdD0/data (iso9660 cdrom)
```

Distribution disk (/dev/sdD0/data, /dev/sdC0/fscache, /)[/]:

The CD-ROM is already mounted at /, so we hit enter to choose the default.

```
% mount /srv/boot /n/distmedia
```

```
Which directory contains the distribution?  
Any of the following will suffice (in order of preference):  
- the root directory of the cd image  
- the directory containing 9front.iso  
- the directory containing 9front.iso.bz2
```

```
Location of archives [/]:
```

And again, the root directory of the CD-ROM is already mounted at /, so hit enter to choose the default.

```
% mount /srv/boot /n/distmedia
```

```
Which directory contains the distribution?  
Any of the following will suffice (in order of preference):  
- the root directory of the cd image  
- the directory containing 9front.iso  
- the directory containing 9front.iso.bz2
```

```
Location of archives [/]:
```

Hit enter.

4.3.13 – copydist Hit enter at the copydist prompt to begin the process of copying the distribution files from the install media to the hard disk.

```
processing /sys/lib/sysconfig/proto/allproto
```

ACHTUNG! Currently, there is no progress meter. For systems without DMA, the copydist task may exceed one hour in duration. Disk activity may be verified by inspecting the stats(8) window.

Eventually, you should see the following:

```
file system made
```

This indicates that the system files have completed copying to the install target.

4.3.14 – sysname

```
Setup network configuration
```

```
sysname [cirno]:
```

Use a system name of your choice, or just hit enter to select the default.

4.3.15 – tzsetup

Setup Time Zone

Time Zone (Argentina, Australia_ACT, Australia_Broken-Hill, Australia_LHI, Australia_NSW, Australia_North, Australia_Queensland, Australia_South, Australia_Sturt, Australia_Tasmania, Australia_Victoria, Australia_West, Australia_Yancowinna, Brazil_Acre, Brazil_DeNoronha, Brazil_East, Brazil_West, CET, Canada_Atlantic, Canada_Central, Canada_East-Saskatchewan, Canada_Eastern, Canada_Mountain, Canada_Newfoundland, Canada_Pacific, Canada_Yukon, Chile_Continental, Chile_EasterIsland, Cuba, EET, Egypt, GB-Eire, GMT, HST, Hongkong, Iceland, Iran, Israel, Jamaica, Japan, Libya, Mexico_BajaNorte, Mexico_BajaSur, Mexico_General, NZ, NZ_CHAT, Navajo, PRC, Poland, ROC, ROK, Singapore, Turkey, US_Alaska, US_Arizona, US_Central, US_East-Indiana, US_Eastern, US_Hawaii, US_Michigan, US_Mountain, US_Pacific, US_Yukon, W-SU, WET)[US_Eastern]:

Daylight saving time (DST) is handled automatically. Type your chosen time zone and hit enter.

4.3.16 – bootsetup

Setup Plan 9 FAT boot partition (9fat)

Plan 9 FAT partition (/dev/sdC0/9fat)[/dev/sdC0/9fat]:

Hit enter to accept the default. Any environment variables entered at the > prompt during boot, as well as settings configured during install will now be written to /n/9fat/plan9.ini and the kernel will be copied to the 9fat partition.

[Continued on next page]

```
dosrv: serving #s/dos
Initializing Plan 9 FAT partition.
% disk/format -r 2 -d -b /386/pbs /dev/sdC0/9fat
Initializing FAT file system
type hard, 12 tracks, 255 heads, 63 sectors/track, 512 bytes/sec
used 4096 bytes
% mount -c /srv/dos /n/9fat /dev/sdC0/9fat
% rm -f /n/9fat/9bootfat /n/9fat/plan9.ini /n/9fat/9pc
% cp /n/newfs/386/9bootfat /n/9fat/9bootfat
% chmod +al /n/9fat/9bootfat
% cp /tmp/plan9.ini /n/9fat/plan9.ini
% cp /n/newfs/386/9pc /n/9fat/9pc
```

If you use the Windows NT/2000/XP master boot record or a master boot record from a Unix clone (e.g., LILO or FreeBSD bootmgr), it is probably safe to continue using that boot record rather than install the Plan 9 boot record.

Since we are not installing on a disk with a pre-existing Windows installation, we choose to install the Plan 9 master boot record and mark the partition active:

```
Install the Plan 9 master boot record (y, n)[no default]: y  
Mark the Plan 9 partition active (y, n)[no default]: y
```

The Plan 9 partition is now marked as active.

4.3.17 – finish

The final task is to remove (or disable) the CD-ROM and finish by hitting enter to reboot the machine.

Congratulations, you've just installed a 9front system!

Now your cat goes to sleep on the keyboard.



4.4 – Encrypted Partitions



9front supports booting from encrypted `fsworm`, `fscache` and other partitions. The following assumes a hard drive `/dev/sdC0`:

- Begin the installation process as normal by booting `9front.iso`.
- During `partdisk` and `prepdisk`, create the `fsworm`, `fscache` and other partitions as normal.
- At the `mountfs` prompt, enter `!rc` to drop to a shell prompt.
- Format and activate the encrypted partitions (generates new encryption key):

```
disk/cryptsetup -f /dev/sdC0/fsworm /dev/sdC0/fscache /dev/sdC0/other  
disk/cryptsetup -i /dev/sdC0/fsworm /dev/sdC0/fscache /dev/sdC0/other
```

- Type `exit` to resume the installation.
- During `mountfs`, select the partitions under `/dev/fs` instead of the ones under `/dev/sdC0`.
- During the rest of the installation, select the defaults, where appropriate.
- During post-installation boot, at the `bootargs` prompt, type `!rc` to drop to a shell prompt.
- Activate the encrypted partitions (prompts for password):

```
disk/cryptsetup -i /dev/sdC0/fsworm /dev/sdC0/fscache /dev/sdC0/other
```

- Type `exit` to resume booting.
- At the `bootargs` prompt, enter `local!/dev/fs/fscache` to continue booting from the encrypted partition.

Note: Encrypted partitions are largely untested with `cwfs`. Data integrity is not

guaranteed. Use at your own risk.

Encrypted partitions have not been tested at all with `hjf`s.

Read: `cryptsetup(8)`