

Advanced - Manual installation of OpenIndiana from LiveCD media

Introduction

For a number of obscure reasons (such as personal preferences, or remote console problems), an interactive installer might not do what you want.

One particular situation is when you'd like to install OpenIndiana (or another OpenSolaris derivative OS with a LiveCD/LiveUSB distro) alongside your existing OS in the same ZFS root pool.

Another situation is if you'd like to set up a "split-root" system with several filesystem datasets comprising your system's Boot Environment so that you can separately compress, quota or clone your system components. This particular task is detailed in illumos bugtracker [bug #829](#) and article [Advanced - Split-root installation](#) (however, *that* conversion can be done after a successful installation of OI to the HDD in a single rootfs dataset).

In these cases you still have a chance to use the LiveCD media to bootstrap your OpenIndiana system the way you want.

NOTE that this procedure can fail in many ways and requires understanding of the steps, of typical Solaris on-disk layout, of ZFS administration, etc. to recover from failures or seek other workarounds.

TYPOGRAPHIC NOTE

In many of the examples below, samples of commands to run as "root" (or via `pfexec`) are prefixed with a " ; " prompt.

Basically this prepends a call of "true" to the sample command and does nothing, unlike the typically used "hash" (#) prompt which acts as a comment and precludes the copped commands from running.

SITUATION BEFORE MANUAL INSTALLATION

If you want to do everything manually, you don't need to run the official installer, as in the example below. In fact, you can only either use the installer or do stuff by hand, because the installer creates the rpool as part of its procedure.

```
root@openindiana:/# DISPLAY=myotherhost:0 /usr/sbin/gui-install &
### (ran some install parts and failed due to lost virtual CD connection)
```

NOTE: This could also be used with a local X11 GUI, but my problematic box has an unsupported video card.

For me it was a starting point which failed for reasons outside its control, but illustrated what structure should be achieved (and initialized the rpool, which includes slicing my disk):

```

root@openindiana:/# df -k
Filesystem          1K-blocks      Used Available Use% Mounted on
/devices/ramdisk:a  206683         194687      11996   95% /
swap                4382104         1160    4380944    1% /etc/svc/volatile
/devices/pci@0,0/pci8086,346f@1d,2/device@2/storage@2/disk@0,0:a
                    821674         821674           0 100% /.cdrom
/dev/lofi/1         2203068        2203068           0 100% /usr
/dev/lofi/2         169460         169460           0 100% /mnt/misc
/mnt/misc/opt       169460         169460           0 100% /mnt/misc/opt
swap                4380980          36    4380944    1% /tmp
swap                4554612        173668    4380944    4% /var/run
swap                4385188          424    4380944    1% /root
swap                4385204          426    4380944    1% /jack
/dev/dsk/c3t0d0s2   821674         821674           0 100% /media/OpenIndiana_Live_X86
rpool               53452797         46   53452752    1% /rpool
rpool/ROOT/openindiana
                    56250956        2798027   53452930    5% /a
rpool/export        53452963          33   53452930    1% /a/export
/var/run/boot_archive
                    159383         147695      11688   93% /var/run/tmpkjYV1V

root@openindiana:/# lofiadm
Block Device      File                                Options
/dev/lofi/1       /.cdrom/solaris.zlib               Compressed(gzip)
/dev/lofi/2       /.cdrom/solarismisc.zlib           Compressed(lzma)
/dev/lofi/3       /var/run/boot_archive              -

root@openindiana:/# zfs list
NAME                                USED  AVAIL  REFER  MOUNTPOINT
rpool                                7.59G  51.0G   33K    /rpool
rpool/ROOT                          2.70G  51.0G   31K    legacy
rpool/ROOT/openindiana              2.70G  51.0G  2.67G    /a
rpool/dump                          2.00G  51.0G  2.00G    -
rpool/export                        794M   51.0G   33K    /a/export
rpool/export/home                   63K   51.0G   32K    /a/export/home
rpool/export/home/admin              31K   51.0G   31K    /a/export/home/admin
rpool/swap                          2.12G  53.0G  137M    -

```

Apparently the installer failed during boot-archive creation? Who knows... who cares? 🤔

Manual installation of OpenIndiana, procedure

CREATING THE RPOOL

The ZFS pool "rpool" is (in the example above) available courtesy of the gui-installer, but just as well could have been made manually, as shown below.

You may find more details in [Advanced - Creating aligned rpool partitions](#) and [Advanced - Creating an rpool manually](#) so this page just provides a short overview.

Steps below were recreated from "zpool history" and active dataset attributes:

```

### PREREQUISITE: A partition with Solaris slices must be precreated by "fdisk" and/or "format"
:; zpool create -R /a -f rpool c2t0d0s0
### This is where you can set up mirrors right away before installation

```

This is also one of the places where you can customize stuff for the whole rpool, like require "-o checksum=sha256 -o copies=3" etc. for the paranoid – or for known-bad hardware 🤔

NOTE1: The root dataset (rpool) and the BE dataset (rpool/ROOT/openindiana) **should not be compressed** (checked, this is still valid as of oi_151a).

NOTE2: Since oi_151a8, you may enable compression=lz4 on root datasets – but earlier builds of GRUB and illumos kernel (i.e. older Live Media) won't be able to mount these roots.

```

:; zfs set org.openindiana.caiman:install=busy rpool
### After a proper setup this gets set to "ready", seems needed for beadm

:; zfs create -b 4096 -V 2045m rpool/swap
:; zfs create -b 131072 -V 2045m rpool/dump
### Sizes seem like RAM/2... or arbitrary?
### Note that there may be more options to these datasets - see detailed pages
:; zfs create -o mountpoint=legacy -o canmount=off rpool/ROOT
:; zfs create -o mountpoint=/ -o canmount=noauto rpool/ROOT/openindiana
:; zfs set org.opensolaris.libbe:uuid=c2c6c968-9866-c662-aacl-86c6cc77c2c8 rpool/ROOT/openindiana
### UUID is random, individual for each BE; maintained by beadm in live systems
### This is where you can customize stuff for the root-BE dataset only; except compression.

### This step clears the way for mounting the rootfs dataset, then
### remounts other datasets (like the pool's root dataset as /a/rpool)
:; zfs umount -a
:; zfs mount rpool/ROOT/openindiana
:; zfs mount -a

```

If you used the installer, you were requested for a non-root user login credentials. Without it the system is unusable since "root" is an RBAC role (not a user account) by default, that user's home dataset is created below:

```

:; zfs create -o mountpoint=/export rpool/export
### NOTE: Compression can be enabled here, i.e.
:; zfs set compression=lzjb rpool/export

:; zfs create rpool/export/home
:; zfs create rpool/export/home/admin

```

Note that the installer also sets "mountpoint" explicitly for datasets under `rpool/export`, but this is not quite required.

This is also the time when you can create and mount sophisticated dataset hierarchies. I for example make a "`rpool/SHARED/var`" structure with some components of "`/var/*`" separated into datasets for better quota management, and they are shared between BEs by virtue of "`/etc/vfstab`". This is an advanced topic I wrote of in detail at [Advanced - Split-root installation](#), so if you don't know why or how you'd do this monstrosity – just don't do it 🙄 (though there are certain benefits, especially for space-constrained systems).

GETTING THE ISO IMAGE TO NEW HOST

In order to speed up the installation, I got the installer image stored in the HDD rpool.

Note that the copy below can also be done directly from the LiveCD or LiveUSB with "dd", but the point of my exercise was to avoid interactions with the slow unreliable virtual device provided by a jKVM applet 🙄

So I copied over network, which should get set up. If you're lucky, OpenIndiana's NWAM service and your LAN's DHCP would take care of this (and your LAN driver would be provided by the booted LiveCD image). Inserting a custom driver (as was needed for one of my two Marvell Yukon cards on another box) can be done with USB or floppy or reimaged installation ISO, that's out of scope here.

If NWAM doesn't work for you, or you have no DHCP in this network segment, you should set up the IP addresses manually (if you're following me closely, you'd note that I actually did this before remotely running the `gui-install`):

```

:; svcadm disable -s physical:nwam
:; svcadm enable physical:default
:; ifconfig e1000g0 plumb
### Plumbing can fail if the adapter is already attached by SMF, we don't care in this case.
:; ifconfig e1000g0 192.168.1.200/24 up
:; route add default 192.168.1.1

```

I created a special dataset for distribution storage:

```

:; zfs create rpool/export/ftp
:; zfs create -o compression=gzip-9 -o copies=1 rpool/export/ftp/distrib

```

Then I copied the oi_151a LiveCD ISO images into it; here it's done by `rsync` from "myotherhost" (in numerical IP address form) which is the admin workstation.

Depending on your network (in)capabilities, any other way including "cp" or "cat" over NFS, or "netcat" over raw sockets, can be used:

```
;; mkdir /a/export/ftp/distribs/openindiana-oi_151a
;; rsync myotherhost:/a/export/ftp/distribs/openindiana-oi_151a/oi-dev-151a-x86.iso* /a/export/ftp/distribs
/openindiana-oi_151a
```

Here I got the "desktop edition" livecd image. I don't need the graphics environment on the server (so I'll disable the "gdm" SMF service after boot), but many programs will need the libraries anyway.

If you can, check the checksums, i.e.:

```
;; md5sum /a/export/ftp/distribs/openindiana-oi_151a/oi-dev-151a-x86.iso
;; cat /a/export/ftp/distribs/openindiana-oi_151a/oi-dev-151a-x86.iso.md5
225508283e3013f273581a769f25a294 /a/export/ftp/distribs/openindiana-oi_151a/oi-dev-151a-x86.iso
225508283e3013f273581a769f25a294
```

REPLACING THE MOUNTED BOOD-CD PATHS WITH AN ISO IMAGE'S PATHS

Now I'll try to overlay-mount the local image over paths provided in the `gui-installer` example above.

First I added a loop device with the newly copied ISO image:

```
;; lofiadm -a /a/export/ftp/distribs/openindiana-oi_151a/oi-dev-151a-x86.iso
/dev/lofi/4
```

Then I overlay-mounted it on top of the installer's image:

```
;; mount -o ro -O -F hsfs /dev/lofi/4 /media/OpenIndiana_Live_X86
```

Then I "made local" other CD-originated paths:

```
;; lofiadm -a /media/OpenIndiana_Live_X86/solaris.zlib
/dev/lofi/5
;; lofiadm -a /media/OpenIndiana_Live_X86/solarismisc.zlib
/dev/lofi/6

;; mount -o ro -O -F hsfs /dev/lofi/5 /usr
;; mount -o ro -O -F hsfs /dev/lofi/6 /mnt/misc
```

Note that there is also a path named `"/mnt/misc/opt"` mounted from `"/mnt/misc/opt"`; this got automatically redirected for me from the local image.

Resulting mounts:

```

root@openindiana:/# df -k
Filesystem          1K-blocks      Used Available Use% Mounted on
/devices/ramdisk:a  206683         194690      11993   95% /
swap                4388596         1160      4387436    1% /etc/svc/volatile
/devices/pci@0,0/pci8086,346f@1d,2/device@2/storage@2/disk@0,0:a
  821674         821674           0 100% /.cdrom
/dev/lofi/1         2203068      2203068           0 100% /usr
/dev/lofi/2         169460       169460           0 100% /mnt/misc
/mnt/misc/opt       169460       169460           0 100% /mnt/misc/opt
swap                4387472          36      4387436    1% /tmp
swap                4561104      173668      4387436    4% /var/run
swap                4387460          24      4387436    1% /root
swap                4391696         4260      4387436    1% /jack
/dev/dsk/c3t0d0s2   821674         821674           0 100% /media/OpenIndiana_Live_X86
rpool               53452797          46     53452752    1% /rpool
rpool/ROOT/openindiana
  56250781      2798030     53452752    5% /a
rpool/export        53452785          33     53452752    1% /a/export
/var/run/boot_archive
  159383       147695       11688    93% /var/run/tmpkjYV1V
rpool/export/ftp    53452784          32     53452752    1% /a/export/ftp
rpool/export/ftp/distribs
  54265614      812863     53452752    2% /a/export/ftp/distribs
/dev/lofi/4         821674         821674           0 100% /media/OpenIndiana_Live_X86
/dev/lofi/5         2203068      2203068           0 100% /usr
/dev/lofi/6         169460       169460           0 100% /mnt/misc

root@openindiana:/# lofiadm
Block Device      File                                          Options
/dev/lofi/1       /.cdrom/solaris.zlib                       Compressed(gzip)
/dev/lofi/2       /.cdrom/solarismisc.zlib                   Compressed(lzma)
/dev/lofi/3       /var/run/boot_archive                       -
/dev/lofi/4       /a/export/ftp/distribs/openindiana-oi_151a/oi-dev-151a-x86.iso -
/dev/lofi/5       /media/OpenIndiana_Live_X86/solaris.zlib    Compressed(gzip)
/dev/lofi/6       /media/OpenIndiana_Live_X86/solarismisc.zlib Compressed(lzma)

```

COPY THE OS IMAGE

Now that you've created the storage structure and got a good source image, there's some `rsync`'ing to do. The Live-running root is a good source for general filesystem structure, as well as the device list matching your hardware. The CD image or its `/usr` might be better sources of installed software.

We use `rsync -x` to copy only the contents of current FS and not descend into sub-FSes like `/var/run` or `/tmp`.

Here goes:

```

:; rsync -xavPHK / /a/
:; rsync -xcavPHK --delete-before /mnt/misc/var/ /a/var/
:; rsync -xcavPHK --delete-before /mnt/misc/opt/ /a/opt/
:; rsync -xcavPHK --delete-before /usr/ /a/usr/

```

CUSTOMIZE THE NEW OS

Non-root user

The LiveCD clone has the `"jack:jack"` user embedded. You may want to edit `/a/etc/passwd` and `/a/etc/shadow` to replace him with your username and perhaps preferred UID (i.e. same as in your other LAN hosts).

See also RBAC `/a/etc/user_attr` and/or `/a/etc/sudoers` setup to allow privilege elevation. This is important to be able to administer your system!

If you want `"jack"` to remain in place, you should change his password after reboot, and perhaps copy his `/jack/` home directory from livecd running image to `/a/jack/` on the HDD.

Static network access

Non-DHCP networking, if required, should be set up in `/a/etc/hosts`, `/a/etc/hostname.DRIVERINSTANCE(:ALIAS)` files, `/a/etc/netmasks`, `/a/etc/defaultrouter`, `/a/etc/defaultdomain`, `/a/etc/resolv.conf` and `/a/etc/nsswitch.conf` – to name a few.

```
;; echo 192.168.1.200 > /a/etc/hostname.e1000g0
;; echo "192.168.1.0 255.255.255.0" >> /a/etc/netmasks
;; echo 192.168.1.1 > /a/etc/defaultrouter
;; echo "myhome.local" > /etc/defaultdomain
```

You might also want proper setup of DNS, maybe NTP or syslogging – but that can all be done after reboot, in a comfortable new system 🍌

Comments in `/a/etc/motd`

Modify as you like, if you like your systems to report - where have you logged into? 🍌

PREPARE THE NEW OS FOR BOOT

Prepare the new OS to (re-)detect hardware, and precreate its boot-archive:

```
;; touch /a/reconfigure
;; bootadm update-archive -R /a
```

Boot system (GRUB) support

Prepare for actual bootup:

```
;; /sbin/installgrub /boot/grub/stage1 /boot/grub/stage2 /dev/rdisk/c2t0d0s0
;; zpool set failmode=continue rpool
;; zpool set bootfs=rpool/ROOT/openindiana rpool
;; zfs set org.openindiana.caiman:install=ready rpool
```

Set up a GRUB menu in `/rpool/boot/grub/menu.lst`. I have copied this from another similar host (along with a number of GRUB resource files), but you can find them in LiveCD/s `/boot/grub/` path as well:

```
;; zfs set mountpoint=/rpool rpool
;; zfs mount rpool
### Just in case - should become /rpool
;; mkdir -p /rpool/boot/grub/bootsign
;; for F in capability menu.lst splash.xpm.gz; do cp -pf /boot/grub/$F /rpool/boot/grub/; done
```

Set up "boot signs" to point GRUB to bootable pools:

```
;; mkdir -p /rpool/etc
;; touch /rpool/boot/grub/bootsign/pool_rpool
;; echo "pool_rpool" >> /rpool/etc/bootsign
```

I customized my GRUB `menu.lst` (adding serial console support and removing splash image; sercon might require additional configuration elsewhere in the BE image), YMMV:

/rpool/boot/grub/menu.lst

```
#splashimage /boot/grub/splash.xpm.gz
#foreground 343434
#background F7FbFF
default 1
timeout 10
# Main GRUB console is serial; press a key on physiscal to grab input
serial --unit=0 --speed=115200 --word=8 --parity=no --stop=1
#serial --unit=0 --speed=9600 ---word=8 --parity=no --stop=1
terminal --timeout=10 serial console

#----- ADDED BY BOOTADM - DO NOT EDIT -----
title openindiana syscon
findroot (pool_rpool,0,a)
bootfs rpool/ROOT/openindiana
kernel$ /platform/i86pc/kernel/$ISADIR/unix -B $ZFS-BOOTFS
module$ /platform/i86pc/$ISADIR/boot_archive

title openindiana sercon
findroot (pool_rpool,0,a)
bootfs rpool/ROOT/openindiana
kernel$ /platform/i86pc/kernel/$ISADIR/unix -B $ZFS-BOOTFS,console=ttya
module$ /platform/i86pc/$ISADIR/boot_archive
#-----END BOOTADM-----

#===== End of LIBBE entry =====
```

NOTE That technically the "findroot" line is not required for "current disk" (the pool has been located anyway) and if you omit the "bootfs" line, GRUB will use the bootfs attribute value from your rpool to locate your "current" BE.

A small Linux habit – have "/etc/grub.conf" pointing to the menu. Not required/used by OpenSolaris itself:

```
;; ln -s ../rpool/boot/grub/menu.lst /a/etc/grub.conf
```

If you did any custom filesystem datasets without automount (split out "/var" or such), or added swap datasets – time to edit "/a/etc/vfstab"...

Prepare low-level SMF services

In the new root BE you should set up some SMF services. For example, livedcd variants of services should be replaced with HDD "default" instances. I embedded an init-script to do this, but still got some recovery to do manually. In hindsight, perhaps I should've disabled-enabled the services in the running livedcd BE and rsynced "/etc/svc/" SMF repository to the hdd BE, somewhat like this:

```
;; svcadm disable root:media usr:media sysidcfg:live-media root-minimal
;; svcadm enable root:default usr:default
;; svcadm enable sysidcfg:system sysidcfg:net
;; rsync -xcavPHK /etc/svc/ /a/etc/svc/
```

- NOTE: I later learned that `svccfg` (interactive or scripted via `stdin`) may be used with an explicitly specified `repository /a/etc/svc/repository.db` to manipulate the after-boot services of the installation right from the Live environment, though in a pretty bulky way (see the `restarter` property group).

I also like my SSH access to start up ASAP, i.e. if the "/export" path is unmountable due to a "polluted" (not empty) directory, the "svc://filesystem/local" service breaks and makes default SSH inaccessible, while I like all consoles to be available so as to remotely repair bugs like these. YMMV, but here's the codez:

```
;; svccfg -s ssh setprop fs-local/entities = fmri: "svc:/system/filesystem/usr"
;; svcadm refresh ssh
;; svcadm enable -r vtdaemon vt2 vt3 vt4 vt5 vt6 console-login
;; rsync -xcavPHK /etc/svc/ /a/etc/svc/
```

Finally, reboot and hope all works; have your console access ready:

```
;; init 6
```

POST-REBOOT FIXES

If you're asked for recovery login, and have not yet changed the `root` password, enter `root` as username and `openindiana` as the default (livecd) password.

If you're offered normal login, use your unprivileged account (like `jack:jack`) and elevate privileges with `pfexec`, `sudo`, `su`, `sudo su -` or such.

While debugging boot problems, you might want to use the `-v -m verbose` options to the kernel in GRUB, to report the kernel's and SMF's boot activities on `/dev/console`.

Some programs might not work in the installed environment, because the active PAM configuration contradicts the preconfigured password hashes in some way. It is recommended to redefine the `root` and unprivileged user passwords. Note that if you install other PAM modules later, notably for the kCIFS server to authenticate CIFS users, you'd need to redefine the passwords again, in order for them to work with these new mechanisms.