

ZFS Troubleshooting Guide

From Siwiki

Contents

- 1 Resolving Hardware Problems
 - 1.1 Diagnosing Potential Problems
 - 1.2 Disk Replacement Example
 - 1.3 Notify FMA That Device Replacement is Complete
 - 1.4 Solving Mirrored {Root} Pool Problems (zpool attach)
 - 1.5 Creating a Pool or Attaching a Disk to a Pool (I/O error)
 - 1.6 Panic/Reboot/Pool Import Problems
 - 1.6.1 Boot From Milestone=None Recovery Method
 - 1.6.2 Boot From OpenSolaris Live CD Recovery Method
 - 1.7 Changing Disk Capacity Sizes
- 2 Resolving ZFS Disk Space Issues
- 3 Resolving Software Problems
 - 3.1 RAID-Z Checksum Errors in Nevada Builds, 120-123
 - 3.2 Unsupported CIFS properties in Solaris 10 Releases
 - 3.3 Cache Device Support Starts in the Solaris 10 10/09 Release
- 4 Resolving NFS Problems
- 5 Pool Migration Issues
- 6 ZFS Installation Issues
 - 6.1 Review Solaris Installation Requirements
 - 6.2 Before You Start
 - 6.3 Solaris/ ZFS Initial Installation
- 7 ZFS Root Pool Recommendations and Requirements
- 8 Solaris Live Upgrade Migration Scenarios
 - 8.1 Review LU Requirements
 - 8.2 Live Upgrade Issues
 - 8.2.1 Live Upgrade Issues (Solaris 10 10/09)
 - 8.3 Live Upgrade with Zones (Solaris 10 10/08)
 - 8.3.1 Migrate a UFS Root File System with Zones Installed to a ZFS Root File System
 - 8.3.2 Configure a ZFS Root File System With Zone Roots on ZFS
 - 8.3.3 Upgrade or Patch a ZFS Root File System With Zone Roots on ZFS
 - 8.4 Recover from BE Removal Failure (ludelete)
- 9 ZFS Root Pool and Boot Issues
 - 9.1 Recover from Lost Root Password or Similar Problem
 - 9.2 Resolving ZFS Mount Point Problems That Prevent Successful Booting
 - 9.2.1 Resolve ZFS Mount Point Problems
 - 9.3 Boot From a Alternate Disk in a Mirrored ZFS Root Pool
 - 9.4 Activating a BE Fails
 - 9.5 ZFS Root Pool Recovery
 - 9.5.1 Complete Solaris ZFS Root Pool Recovery
 - 9.5.1.1 Before You Begin
 - 9.5.1.2 Create and Send Root Pool Snapshots
 - 9.5.1.3 Bare Metal Receive Root Pool Snapshots
 - 9.5.2 Replacing/Relabeling the Root Pool Disk
 - 9.5.3 Rolling Back a Root Pool Snapshot From a Failsafe Boot
 - 9.6 Primary Mirror Disk in a ZFS Root Pool is Unavailable or Fails
 - 9.7 ZFS Swap and Dump Devices
 - 9.7.1 Resizing ZFS Swap and Dump Devices
 - 9.7.1.1 Adjusting the Size of the Swap Volume on an Active System

9.7.2 Destroying an Inactive Root Pool With an Active Dump/Swap Device

Resolving Hardware Problems

Diagnosing Potential Problems

- General commands for diagnosing hardware problems are:

```
zpool status
zpool status -v
fmdump
fmdump -ev or fmdump -eV
format or rmformat
```

- Identify hardware problems with the `zpool status` commands. If a pool is in the `DEGRADED` state, use the `zpool status` command to identify if a disk is unavailable. For example:

```
# zpool status -x
pool: zeepool
state: DEGRADED
status: One or more devices could not be opened. Sufficient replicas exist for
the pool to continue functioning in a degraded state.
action: Attach the missing device and online it using 'zpool online'.
see: http://www.sun.com/msg/ZFS-8000-D3
scrub: resilver completed after 0h12m with 0 errors on Thu Aug 28 09:29:43 2008
config:

    NAME                STATE          READ WRITE CKSUM
    zeepool              DEGRADED      0     0     0
      mirror            DEGRADED      0     0     0
        c1t2d0          ONLINE        0     0     0
        spare           DEGRADED      0     0     0
          c2t1d0        UNAVAIL       0     0     0  cannot open
          c2t3d0        ONLINE        0     0     0
    spares
      c1t3d0             AVAIL
      c2t3d0             INUSE         currently in use

errors: No known data errors
```

- See the disk replacement example to recover from a failed disk.
- Identify potential data corruption with the `zpool status -v` command. If only one file is corrupted, then you might choose to deal with it directly, without needing to restore the entire pool.

```
# zpool status -v rpool
pool: rpool
state: DEGRADED
status: One or more devices has experienced an error resulting in data
corruption. Applications may be affected.
action: Restore the file in question if possible. Otherwise restore the
entire pool from backup.
see: http://www.sun.com/msg/ZFS-8000-8A
scrub: scrub completed after 0h2m with 1 errors on Tue Mar 11 13:12:42 2008
config:

    NAME                STATE          READ WRITE CKSUM
    rpool              DEGRADED      0     0     9
      c2t0d0s0          DEGRADED      0     0     9

errors: Permanent errors have been detected in the following files:
    /mnt/root/lib/amd64/libc.so.1
```

- Display the list of suspected faulty devices using the `fmdump` command. It is also useful to know the diagnosis engines available on your system and how busy they have been, which is obtained via the `fmstat` command. Similarly, `fmadm` will show the status of the diagnosis engines. You can also see that there are 4 diagnosis

engines which are appropriate to devices and ZFS: disk-transport, io-retire, zfs-diagnosis, and zfs-retire. Check your OS release for the available FMA diagnosis engine capability.

```
# fmdump
TIME                UUID                SUNW-MSG-ID
Aug 18 18:32:48.1940 940422d6-03fb-4ea0-b012-aec91b8dafd3 ZFS-8000-D3
Aug 21 06:46:18.5264 692476c6-a4fa-4f24-e6ba-8edf6f10702b ZFS-8000-D3
Aug 21 06:46:18.7312 45848a75-eae5-66fe-a8ba-f8b8f81deae7 ZFS-8000-D3
# fmstat
module              ev_rcv  ev_acpt  wait   svc_t   %w   %b   open  solve  memsz  bufisz
cpumem-retire       0        0  0.0    0.0    0   0    0     0     0     0     0
disk-transport      0        0  0.0    55.9   0   0    0     0     0    32b    0
eft                 0        0  0.0    0.0    0   0    0     0     0    1.2M   0
fabric-xlate        0        0  0.0    0.0    0   0    0     0     0     0     0
fmd-self-diagnosis  0        0  0.0    0.0    0   0    0     0     0     0     0
io-retire           0        0  0.0    0.0    0   0    0     0     0     0     0
snmp-trapgen        0        0  0.0    0.0    0   0    0     0     0    32b    0
sysevent-transport 0        0  0.0    4501.8 0   0    0     0     0     0     0
syslog-msgs         0        0  0.0    0.0    0   0    0     0     0     0     0
zfs-diagnosis       0        0  0.0    0.0    0   0    0     0     0     0     0
zfs-retire          0        0  0.0    0.0    0   0    0     0     0     0     0
# fmadm config
MODULE              VERSION  STATUS  DESCRIPTION
cpumem-retire       1.1     active  CPU/Memory Retire Agent
disk-transport      1.0     active  Disk Transport Agent
eft                 1.16    active  eft diagnosis engine
fabric-xlate        1.0     active  Fabric Ereport Translator
fmd-self-diagnosis  1.0     active  Fault Manager Self-Diagnosis
io-retire           2.0     active  I/O Retire Agent
snmp-trapgen        1.0     active  SNMP Trap Generation Agent
sysevent-transport 1.0     active  SysEvent Transport Agent
syslog-msgs         1.0     active  Syslog Messaging Agent
zfs-diagnosis       1.0     active  ZFS Diagnosis Engine
zfs-retire          1.0     active  ZFS Retire Agent
```

- Display more details about potential hardware problems by examining the error reports with `fmdump -ev`. Display even more details with `fmdump -eV`.

```
# fmdump -eV
TIME                CLASS
Aug 18 2008 18:32:35.186159293 ereport.fs.zfs.vdev.open_failed
nvlist version: 0
  class = ereport.fs.zfs.vdev.open_failed
  ena = 0xd3229ac5100401
  detector = (embedded nvlist)
  nvlist version: 0
    version = 0x0
    scheme = zfs
    pool = 0x4540c565343f39c2
    vdev = 0xcba57455fe08750b
  (end detector)

  pool = whoo
  pool_guid = 0x4540c565343f39c2
  pool_context = 1
  pool_failmode = wait
  vdev_guid = 0xcba57455fe08750b
  vdev_type = disk
  vdev_path = /dev/ramdisk/rdx
  parent_guid = 0x4540c565343f39c2
  parent_type = root
  prev_state = 0x1
  __ttl = 0x1
  __tod = 0x48aa22b3 0xb1890bd
```

- If expected devices can't be displayed with the `format` or `rmformat` utility, then those devices won't be visible to ZFS.

Disk Replacement Example

Replacing Devices in a Pool

- In the following example, we are assuming that `c4t60060160C166120099E5419F6C29DC11d0s6` is the faulty disk and it is replaced with `c4t60060160C16612006A4583D66C29DC11d0s6` in the `z-mirror`

pool.

```
# zpool status z-mirror
pool: z-mirror
state: ONLINE
scrub: resilver completed with 0 errors on Tue Sep 11 09:05:44 2007
config:

    NAME                                STATE      READ  WRITE  CKSUM
    z-mirror                             ONLINE    0     0     0
    mirror
      c4t60060160C166120064F22DA86C29DC11d0s6  ONLINE    0     0     0
      c4t60060160C166120099E5419F6C29DC11d0s6  FAULTED   0     0     0
```

- If you are replacing the faulted disk with a replacement disk in the same physical location, physically replace the new disk.
- Let ZFS know that the faulted disk has been replaced by using this syntax: `zpool replace [-f] <pool> <device> [new_device]`

```
# zpool replace z-mirror c4t60060160C166120099E5419F6C29DC11d0s6 c4t60060160C16612006A4583D66C29DC11d0s6
```

If you are replacing a disk in the same physical location, then you only need identify the original device. For example:

```
# zpool replace z-mirror c4t60060160C166120099E5419F6C29DC11d0s6
```

- Review the pool status.

```
# zpool status z-mirror
pool: z-mirror
state: ONLINE
scrub: resilver completed with 0 errors on Tue Sep 11 09:08:44 2007
config:

    NAME                                STATE      READ  WRITE  CKSUM
    z-mirror                             ONLINE    0     0     0
    mirror
      c4t60060160C166120064F22DA86C29DC11d0s6  ONLINE    0     0     0
      c4t60060160C16612006A4583D66C29DC11d0s6  ONLINE    0     0     0

errors: No known data errors
```

- If necessary, clear any existing errors that resulted from the faulted disk.

```
# zpool clear z-mirror
```

- Scrub the pool to ensure the new device is functioning properly.

```
# zpool scrub z-mirror
```

Notify FMA That Device Replacement is Complete

Template:Draft Follow these steps after the device is replaced, the pool errors are cleared, and the pool is scrubbed.

- Review the FMA/zfs diagnostic counters from the previous fault.

```
# fmstat
```

- Reset the ZFS diagnostic counters and determine whether any new fault activity is occurring.

```
# fmadm reset zfs-diagnosis
# fmadm reset zfs-retire
# fmstat
```

- Determine the FMA fault event from the failed device.

```
# fmadm faulty -a
-----
TIME                EVENT-ID                MSG-ID                SEVERITY
-----
Jul 17 11:03:56 378924d1-840b-c4dd-c8e2-a5491d4047ff  ZFS-8000-D3        Major
...
Fault class : fault.fs.zfs.device
Affects      : zfs://pool=rzpool/vdev=70f7855d9f673fcc
               faulted but still in service
Problem in   : zfs://pool=rzpool/vdev=70f7855d9f673fcc
               faulted but still in service
...
```

- Let FMA know that the ZFS fault event is cleared.

```
# fmadm repair zfs://pool=rzpool/vdev=70f7855d9f673fcc
fmadm: recorded repair to zfs://pool=rzpool/vdev=70f7855d9f673fcc
```

- Confirm that no new faults have occurred.

```
# fmadm faulty
```

Solving Mirrored {Root} Pool Problems (zpool attach)

- If you cannot attach a disk to create a mirrored root or non-root pool with the zpool attach command, and you see messages similar to the following:

```
# zpool attach rpool c1t1d0s0 c1t0d0s0
cannot attach c1t0d0s0 to c1t1d0s0: new device must be a single disk
```

- You might be running into CR 6852962, which was seen in an LDOM environment.
- If the problem is not 6852962 and the system is booted under a virtualization product, make sure the devices are accessible by ZFS outside of the virtualization product.

Creating a Pool or Attaching a Disk to a Pool (I/O error)

If you attempt to create a pool or attach a disk or a disk slice to an existing pool and you see the following error:

```
# zpool attach rpool c4t0d0s0 c4t1d0s0
cannot open '/dev/dsk/c4t1d0s0': I/O error
```

This error means that the disk slice doesn't have any disk space allocated to it or possibly that a Solaris fdisk partition and the slice doesn't exist on an x86 system. Use the format utility to allocate disk space to a slice. If the x86 system doesn't have a Solaris fdisk partition, use the fdisk utility to create one.

Panic/Reboot/Pool Import Problems

During the boot process, each pool must be opened, which means that pool failures might cause a system to enter into a panic-reboot loop. In order to recover from this situation, ZFS must be informed not to look for any pools on startup.

Boot From Milestone=None Recovery Method

- Boot to the none milestone by using the -m milestone=None boot option.

```
ok boot -m milestone=None
```

- Remount your root file system as writable.

- Rename or move the `/etc/zfs/zpool.cache` file to another location.

These actions cause ZFS to forget that any pools exist on the system, preventing it from trying to access the pools. If you have multiple pools on the system, do these additional steps:

- * Determine which pool might have issues by using the `fmddump -eV` command to display the pools with their status.
- * Import the pools one-by-one, skipping the pools that are having issues, as described in the `fmddump` command.

- If the system is back up, issue the `svcadm milestone all` command.

Boot From OpenSolaris Live CD Recovery Method

Template:Draft

If you are running a Solaris SXCE or Solaris 10 release, you might be able to boot from the OpenSolaris Live CD and fix whatever is causing the pool import to fail.

- Boot from the OpenSolaris Live CD
- Import the pool
- Resolve the issue that causes the pool import to fail, such as replace a failed disk
- Export the pool
- Boot from the original Solaris release
- Import the pool

Changing Disk Capacity Sizes

- You can use the `autoexpand` property to expand a disk's size in the Nevada release, build 117. For example, two 16-GB disks in a mirrored pool are replaced with two 72-GB disks. The `autoexpand` property is enabled after the disk replacements to expand the full LUN sizes.

```
# zpool list pool
NAME      SIZE  USED  AVAIL    CAP  HEALTH  ALTROOT
pool     16.8G  76.5K  16.7G    0%  ONLINE  -
# zpool replace pool c1t16d0 c1t1d0
# zpool replace pool c1t17d0 c1t2d0
# zpool list pool
NAME      SIZE  USED  AVAIL    CAP  HEALTH  ALTROOT
pool     16.8G  88.5K  16.7G    0%  ONLINE  -
# zpool set autoexpand=on pool
# zpool list pool
NAME      SIZE  USED  AVAIL    CAP  HEALTH  ALTROOT
pool     68.2G  117K  68.2G    0%  ONLINE  -
```

- If you resize a LUN from a storage array and the `zpool status` command doesn't display the LUN's expected capacity, export and import the pool to see expected capacity. This problem is related to CRs 6475340/6606879, fixed in the Nevada release, build 117.
- If `zpool status` doesn't display the array's LUN expected capacity, confirm that the expected capacity is visible from the format utility. For example, the format output below shows that one LUN is configured as 931.01 GB and one is configured as 931.01 MB.

```
2. c6t600A0B800049F93C0000030A48B3EA2Cd0 <SUN-LCSM100_F-0670-931.01GB>
   /scsi_vhci/ssd@g600a0b800049f93c0000030a48b3ea2c
3. c6t600A0B800049F93C0000030D48B3EAB6d0 <SUN-LCSM100_F-0670-931.01MB>
   /scsi_vhci/ssd@g600a0b800049f93c0000030d48b3eab6
```

- You will need to reconfigure the array's LUN capacity with the array sizing tool to correct this sizing problem.
- When the LUN sizes are resolved, export and import the pool if the pool has already been created with these LUNs.

Resolving ZFS Disk Space Issues

- The `zpool list` command reports physical disk capacity in the pool and `zfs list` command reports usable space

that is available to file systems, which is disk space minus ZFS redundancy metadata overhead, if any. If the `zpool list` command reports additional capacity, but the `zfs list` command says the available file system is 0, then the file system is full.

- We recommend using the `zpool list` and `zfs list` commands rather than the `df` command to identify available pool space and available file system space. The `df` doesn't understand descendent datasets, whether snapshots exist, nor is `df` dedup-aware. If any ZFS properties, such as compression and quotas, are set on file systems, reconciling the space consumption that is reported by `df` might be difficult.
- For more information about the ZFS deduplication feature impacts space accounting, see [ZFS Deduplication \(http://hub.opensolaris.org/bin/view/Community+Group+zfs/faq#HWhatisZFSdeduplication\)](http://hub.opensolaris.org/bin/view/Community+Group+zfs/faq#HWhatisZFSdeduplication) .
- The `zpool list` and `zpool import` command output has changed starting in Nevada, builds 125-129. For more information, see [Why has the `zpool` command changed? \(http://hub.opensolaris.org/bin/view/Community+Group+zfs/faq#HWhyhashezpoolcommandchanged\)](http://hub.opensolaris.org/bin/view/Community+Group+zfs/faq#HWhyhashezpoolcommandchanged) .

Resolving Software Problems

RAID-Z Checksum Errors in Nevada Builds, 120-123

Update: This bug is fixed in Nevada, build 124.

- A bug (6869090) in NV build 120 causes checksum errors in RAID-Z configurations as follows:
 - Data corruption on a RAID-Z system of any sort (`raidz1`, `raidz2`, `raidz3`) can lead to spurious checksum errors being reported on devices that were not used as part of the reconstruction. These errors are harmless and can be cleared safely (`zpool clear <pool>`).
 - A far more serious problem with single-parity RAID-Z configurations can lead to data corruption. This data corruption is recoverable as long as no additional data corruption or drive failure occurs. That is to say, data is fine provided no additional problems occur. The problem is present on all RAIDZ-1 configurations that use an odd number of children (disks) for example, 4+1, or 6+1. Note that RAIDZ-1 configurations with an even number of children (for example, 3+1), `raidz2`, and `raidz3` are unaffected.
- Mirrored storage pools are not impacted by this bug.
- Solaris 10 releases are not impacted by this bug.
- OpenSolaris releases are not impacted by this bug unless you have updated to build 120-123.
- The recommended course of action is to roll back to build `snv_119` or earlier. If for some reason this is impossible, please email `zfs-discuss`, and we can discuss the best course of action for you. After rolling back, initiate a scrub. ZFS will identify and correct these errors, but if enough accumulate it will (incorrectly) identify drives as faulty (which they likely aren't). You can clear these failures (`zpool clear <pool>`).
- Without rolling back, repeated scrubs will eventually remove all traces of the data corruption. You may need to clear checksum failures as they're identified to ensure that enough drives remain online.

Unsupported CIFS properties in Solaris 10 Releases

- The Solaris 10 10/08 release includes modifications to support the Solaris CIFS environment as described in `zfs.1m`. However, the CIFS features are not supported in the Solaris 10 release. Therefore, these properties are set to read-only values. If you attempt to reset the CIFS-related properties, you will see a message similar to the following:

```
-----
# zfs set utf8only=on rpool
cannot set property for 'rpool': 'utf8only' is readonly
-----
```

Cache Device Support Starts in the Solaris 10 10/09 Release

- Cache devices are supported starting in the Solaris 10 10/09 release. Starting in the Solaris 10 10/08 release, cache device support is identified as available by using the `zpool upgrade -v` command. For example:

```
-----
# zpool upgrade -v
This system is currently running ZFS pool version 10.
The following versions are supported:
-----
```

VER	DESCRIPTION
1	Initial ZFS version
2	Ditto blocks (replicated metadata)
3	Hot spares and double parity RAID-Z
4	zpool history
5	Compression using the gzip algorithm
6	bootfs pool property
7	Separate intent log devices
8	Delegated administration
9	refquota and reservation properties
10	Cache devices

- However, cache devices are not supported in Solaris 10 releases prior to the Solaris 10 10/09 release.
- If you attempt to add a cache device to a ZFS storage pool when the pool is created, the following message is displayed:

```
# zpool create pool mirror c0t1d0 c0t2d0 cache c0t3d0
cannot create 'pool': operation not supported on this type of pool
```

- If you attempt to add a cache device to a ZFS storage pool after the pool is created, the following message is displayed:

```
# zpool create pool mirror c0t1d0 c0t2d0
# zpool add pool cache c0t3d0
cannot add to 'pool': pool must be upgraded to add these vdevs
```

Resolving NFS Problems

- Current Solaris releases run NFSv4 by default, which provides access to ZFS style ACLs. If you are sharing data with Linux systems or older Solaris systems, which might be running NFSv3, the NFSv4 ACL info will be available but not viewable or modifiable.
- If you have Linux clients and you want to traverse nested mount points, you'll need to use NFSv4 on these clients by specifying the "mount -t nfs4" option.
- If you are sharing UIDs/GIDs across systems, make sure the UIDs/GIDs are translated correctly. If you see "nobody" instead of a UID, then check that the NFSMAPID_DOMAIN is set correctly. If you update the domain name, restart the svc:/network/nfs/mapid:default service.
- Check basic network connectivity and NFS sharing first before testing ACL behavior. Create and share a test file system with no ACLs first and see if the data is accessible on the NFS client.
- Use getent to check that hostnames and usernames are available across systems.
- Current OpenSolaris releases provide the mirror mount feature, which means that when file systems are created on the NFS server, the NFS client can automatically discover these newly created file systems within their existing mount of a parent file system. This feature is not available in Solaris 10 releases.
- Current OpenSolaris releases require fully-qualified hostnames when you use the sharenfs property to share ZFS file systems.

Pool Migration Issues

- Moving a ZFS storage pool from a FreeBSD system to a Solaris system:
 - A ZFS storage pool that is created on a FreeBSD system is created on a disk's p* devices with an EFI label. On a FreeBSD system, you can boot from a disk that has an EFI label.
 - A ZFS storage pool that is created on a Solaris system is either created on a disk's d* devices with an EFI label or if the pool is used for booting, the pool must be created on disk slices (s* devices) with an SMI label. You cannot boot from an disk that has an EFI label in Solaris releases.
 - If you are migrating a pool from a FreeBSD system that was used for booting, you will need to unset the bootfs property before the migration if the migrated pool will no longer be used for booting on the Solaris system. An existing bug in the Solaris releases does not allow you to unset the bootfs property on a pool that has disks with an EFI label. If you don't unset the bootfs property before the migration, you will not be able to add whole disks to this pool.

- A workaround is to boot the system under FreeBSD's Fixit mode and use the `zpool set bootfs=pool-name` command to unset the `bootfs` property. Then, import the pool on a Solaris system.
- A pool that is used for booting on a FreeBSD system can't be used for booting on a Solaris system. You would need to relabel the disks with SMI labels and allocate space in the disk slices. This would cause any data on the existing pool to be removed.
- A larger issue is that a ZFS storage pool should not be created on a disk's `p*` (partition) devices. In Solaris releases, the `p*` devices represent the larger `fdisk` partitions. These partitions are available for a Solaris installation and other possible OSes. A ZFS storage pool should be created on a disk's `d*` devices that represent the whole disk. Creating a pool on a disk's `s*` devices represent a slice within the disk and should be used when creating a ZFS root pool or for some other specialized purpose.
- The problem with creating a storage pool on a disk's `p*` devices is that you could conceivably create pools with overlapping disk partitions, such as creating one pool with `c0t1d0p0` and another pool with `c0t1d0`, for example. This overlap could cause data loss or a system crash. This is CR 6884447.

ZFS Installation Issues

Review Solaris Installation Requirements

- 768 MB is the minimum amount of memory required to install a ZFS root file system
- 1 GB of memory is recommended for better overall ZFS performance
- At least 16 GB of disk space is recommended
- ZFS boot and install support is provided starting in the Solaris 10 10/08 release, SXCE, build 90, and the OpenSolaris releases

Before You Start

- Due to an existing boot limitation, disks intended for a bootable ZFS root pool must be created with disk slices and must be labeled with a VTOC (SMI) disk label.
- If you relabel EFI-labeled disks with VTOC labels, be sure that the desired disk space for the root pool is in the disk slice that will be used to create the bootable ZFS pool.

Solaris/ ZFS Initial Installation

- For the OpenSolaris releases, a ZFS root file system is installed by default and there is no option to choose another type of root file system.
- For SXCE, build 90 and Solaris 10 10/08 releases, you can only install a ZFS root file system from the text installer.
- You cannot use the standard upgrade option to install or migrate to a ZFS root file system. For information on creating ZFS flash archive in the Solaris 10 release by installing patches, go to ZFS flash support (<http://opensolaris.org/os/community/zfs/boot/flash/>).
- Starting in SXCE, build 90 and the Solaris 10 10/08 release, you can use Solaris LiveUpgrade to migrate a UFS root file system to a ZFS root file system.
- Access the text installer as follows:
 - On SPARC based system, use the following syntax from the Solaris installation DVD or the network:

```
-----
ok boot cdrom - text
ok boot net - text
**On an x86 based system, select the text-mode install option when presented.
-----
```

- If you accidentally start the GUI install method, do the following:

```
-----
* Exit the GUI installer
* Expand the terminal window to 80 x 24
* Unset the DISPLAY, like this:
# DISPLAY=
# export DISPLAY
# install-solaris
-----
```

ZFS Root Pool Recommendations and Requirements

- During an initial installation, select two disks to create a mirrored root pool. Or, you can also attach a disk to create a mirrored root pool after installation. See the ZFS Administration Guide for details.
- Disks intended for the root pool must contain a slice and have an SMI label. Disks with EFI labels cannot be booted. Several factors are at work here, including BIOS support for booting from EFI labeled disks.
- Minimum root pool version is 10.
- If you attach another disk to create a mirrored root pool later, make sure you specify a bootable slice and not the whole disk because the latter may try to install an EFI label.
- You cannot use a RAID-Z configuration for a root pool. Only single-disk pools or pools with mirrored disks are supported. You will see the following message if you attempt to use an unsupported pool for the root pool:

```
ERROR: ZFS pool <pool-name> does not support boot environments
```

- Root pools cannot have a separate log device. For example:

```
# zpool add -f rpool log c0t6d0s0
cannot add to 'rpool': root pool can not have multiple vdevs or separate logs
```

- The lzjb compression property is supported for root pools but the other compression types are not supported.
- Keep a second ZFS BE for recovery and patching purposes. This process is safer than patching a live boot environment for two reasons:
 - You're applying the patches to an inactive boot environment, which is safer than patching a live environment
 - Your previous environment isn't modified at all, which makes it a secure environment to go back to if the new one fails
- The patching or upgrading process is as follows:
 - Clone your current boot environment, using either LiveUpgrade or beadm (beadm is part of the OpenSolaris release).
 - Apply the patches or perform the upgrade to the clone BE.
 - Activate and boot the clone BE.
 - If the patched or upgraded boot environment fails, reboot the old one.
- Keep root pool snapshots on a remote system. See the steps below for details.

Solaris Live Upgrade Migration Scenarios

- You can use the Solaris Live Upgrade feature to migrate a UFS root file system to a ZFS root file system.
- You can't use Solaris Live Upgrade to migrate a ZFS boot environment (BE) to UFS BE.
- You can't use Solaris Live Upgrade to migrate non-root or shared UFS file systems to ZFS file systems.

Review LU Requirements

- You must be running the SXCE, build 90 release or the Solaris 10 10/08 release to use LU to migrate a UFS root file system to a ZFS root file system.
- You must create a ZFS storage pool that contains disk slices before the LU migration.
- The pool must exist either on a disk slice or on disk slices that are mirrored, but not on a RAID-Z configuration or on a nonredundant configuration of multiple disks. If you attempt to use an unsupported pool configuration during a Live Upgrade migration, you will see a message similar to the following:

```
ERROR: ZFS <pool-name> does not support boot environments
```

- If you see this message, then either the pool doesn't exist or its an unsupported configuration.

Live Upgrade Issues

- The Solaris installation GUI's standard-upgrade option is not available for migrating from a UFS to a ZFS root file system. To migrate from a UFS file system, you must use Solaris Live Upgrade. You cannot use Solaris Live Upgrade to create a UFS BE from a ZFS BE.
- Do not rename your ZFS BEs with the `zfs rename` command because the Solaris Live Upgrade feature is unaware of the name change. Subsequent commands, such as `ludelete`, will fail. In fact, do not rename your ZFS pools or file systems if you have existing BEs that you want to continue to use.
- Solaris Live Upgrade creates the datasets for the BE and ZFS volumes for the swap area and dump device but does not account for any existing dataset property modifications. Thus, if you want a dataset property enabled in the new BE, you must set the property before the `lucreate` operation. For example:

```
# zfs set compression=on rpool/ROOT
```

- When creating an alternative BE that is a clone of the primary BE, you cannot use the `-f`, `-x`, `-y`, `-Y`, and `-z` options to include or exclude files from the primary BE. You can still use the inclusion and exclusion option set in the following cases:

UFS -> UFS UFS -> ZFS ZFS -> ZFS (different pool)

- Although you can use Solaris Live Upgrade to upgrade your UFS root file system to a ZFS root file system, you cannot use Solaris Live Upgrade to upgrade non-root or shared file systems.
- On a SPARC system that runs the Solaris 10 5/09 release, set the `BOOT_MENU_FILE` variable before activating the ZFS BE with `luactivate`, due to CR 6824589.

```
# BOOT_MENU_FILE="menu.lst"
# export BOOT_MENU_FILE
```

- LiveUpgrade is not supported on a system in diagnostic mode, where the `diag` switch is set to `true`.

Live Upgrade Issues (Solaris 10 10/09)

Template:Draft Review the following information before upgrading to the Solaris 10 10/09 release.

- If you have a separate `/var` dataset in a previous Solaris 10 release and you want to use `luupgrade` to upgrade to the Solaris 10 10/09 release, and you have applied patch 121430-42 (on SPARC systems) or 121431-43 (on x86 systems), apply IDR143154-01 (SPARC) or IDR143155-01 (x86) to avoid the issue described in CR 6884728. Symptoms of this bug are that entries for the ZFS BE and ZFS `/var` dataset are incorrectly placed in the `/etc/vfstab` file. The workaround is to boot in maintenance mode and remove the erroneous `vfstab` entries.
- If you attempt to use `lucreate` and you have a ZFS dataset with a mountpoint set to `legacy` in a non-global zone, the `luactivate` might fail. This is CR 6837400.
- If the `luactivate` fails, you might need to reinstall the boot blocks to boot from the previous BE.

Live Upgrade with Zones (Solaris 10 10/08)

Review the following supported ZFS and zones configurations. These configurations are upgradeable and patchable.

Migrate a UFS Root File System with Zones Installed to a ZFS Root File System

This ZFS zone root configuration can be upgraded or patched. See the ZFS Administration Guide for information about supported zones configurations that can be upgraded or patched in the Solaris 10 release.

1. Upgrade the system to the Solaris 10 10/08 release if it is running a previous Solaris 10 release.
2. Create the root pool.

```
# zpool create rpool mirror c1t0d0s0 c1t1d0s0
```

3. Confirm that the zones from the UFS environment are booted.
4. Create the new boot environment.

```
# lucreate -n S10BE2 -p rpool
```

5. Activate the new boot environment.
6. Reboot the system.
7. Migrate the UFS zones to the ZFS BE.
 1. Boot the zones.
 2. Create another BE within the pool.

```
# lucreate S10BE3
```

3. Activate the new boot environment.

```
# luactivate S10BE3
```

4. Reboot the system.

```
# init 6
```

8. Resolve any potential mount point problems, due to a Solaris Live Upgrade bug.
 1. Review the zfs list output and look for any temporary mount points.

```
# zfs list -r -o name,mountpoint rpool/ROOT/s10u6

NAME                                MOUNTPOINT
rpool/ROOT/s10u6                    /.alt.tmp.b-VP.mnt/
rpool/ROOT/s10u6/zones              /.alt.tmp.b-VP.mnt//zones
rpool/ROOT/s10u6/zones/zonerootA   /.alt.tmp.b-VP.mnt/zones/zonerootA
```

The mount point for the root ZFS BE (rpool/ROOT/s10u6) should be /.

2. Reset the mount points for the ZFS BE and its datasets.

```
# zfs inherit -r mountpoint rpool/ROOT/s10u6
# zfs set mountpoint=/ rpool/ROOT/s10u6
```

3. Reboot the system. When the option is presented to boot a specific boot environment, either in the GRUB menu or at the OpenBoot Prom prompt, select the boot environment whose mount points were just corrected.

Configure a ZFS Root File System With Zone Roots on ZFS

Set up a ZFS root file system and ZFS zone root configuration that can be upgraded or patched. In this configuration, the ZFS zone roots are created as ZFS datasets.

1. Install the system with a ZFS root, either by using the interactive initial installation method or the Solaris JumpStart installation method.
2. Boot the system from the newly-created root pool.
3. Create a dataset for grouping the zone roots.

```
zfs create -o canmount=noauto rpool/ROOT/S10be/zones
```

Setting the noauto value for the canmount property prevents the dataset from being mounted other than by the explicit action of Solaris Live Upgrade and system startup code.

4. Mount the newly-created zones container dataset.

```
# zfs mount rpool/ROOT/S10be/zones
```

The dataset is mounted at /zones.

5. Create and mount a dataset for each zone root.

```
# zfs create -o canmount=noauto rpool/ROOT/S10be/zones/zonerootA
# zfs mount rpool/ROOT/S10be/zones/zonerootA
```

6. Set the appropriate permissions on the zone root directory.

```
# chmod 700 /zones/zonerootA
```

7. Configure the zone, setting the zone path as follows:

```
# zonecfg -z zoneA
zoneA: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:zoneA> create
zonecfg:zoneA> set zonepath=/zones/zonerootA
```

8. Install the zone.

```
# zoneadm -z zoneA install
```

9. Boot the zone.

```
# zoneadm -z zoneA boot
```

Upgrade or Patch a ZFS Root File System With Zone Roots on ZFS

Upgrade or patch a ZFS root file system with zone roots on ZFS. These updates can either be a system upgrade or the application of patches.

1. Create the boot environment to upgrade or patch.

```
# lucreate -n newBE
```

The existing boot environment, including all the zones, are cloned. New datasets are created for each dataset in the original boot environment. The new datasets are created in the same pool as the current root pool.

2. Select one of the following to upgrade the system or apply patches to the new boot environment.

- Upgrade the system.

```
# luupgrade -u -n newBE -s /net/install/export/s10u7/latest
```

Where the -s option is the location of a Solaris installation medium.

- Apply patches to the new boot environment.

```
# luupgrade -t -n newBE -t -s /patchdir 139147-02 157347-14
```

3. Activate the new boot environment after the updates to the new boot environment are complete.

```
# luactivate newBE
```

4. Boot from newly-activated boot environment.

```
# init 6
```

5. Resolve any potential mount point problems, due to a Solaris Live Upgrade bug.
 1. Review the zfs list output and look for any temporary mount points.

```
# zfs list -r -o name,mountpoint rpool/ROOT/newBE

NAME                                MOUNTPOINT
rpool/ROOT/newBE                    /.alt.tmp.b-VP.mnt/
rpool/ROOT/newBE/zones              /.alt.tmp.b-VP.mnt//zones
rpool/ROOT/newBE/zones/zonerootA    /.alt.tmp.b-VP.mnt/zones/zonerootA
```

The mount point for the root ZFS BE (rpool/ROOT/newBE) should be /.

2. Reset the mount points for the ZFS BE and its datasets.

```
# zfs inherit -r mountpoint rpool/ROOT/newBE
# zfs set mountpoint=/ rpool/ROOT/newBE
```

3. Reboot the system. When the option is presented to boot a specific boot environment, either in the GRUB menu or at the OpenBoot Prom prompt, select the boot environment whose mount points were just corrected.

Recover from BE Removal Failure (ludelete)

- If you use ludelete to remove an unwanted BE and it fails with messages similar to the following:

```
# ludelete -f c0t1d0s0
System has findroot enabled GRUB
Updating GRUB menu default setting
Changing GRUB menu default setting to <0>
ERROR: Failed to copy file </boot/grub/menu.lst> to top level dataset for BE <c0t1d0s0>
ERROR: Unable to delete GRUB menu entry for deleted boot environment <c0t1d0s0>.
Unable to delete boot environment.
```

- You might be running into the following bugs: 6718038 or 6715220. These issues are fixed in the Solaris 10 10/08 release.
 - The workaround is as follows:
1. Edit /usr/lib/lu/lulib and in line 2934, replace the following text:

```
lulib_copy_to_top_dataset "$BE_NAME" "$ldme_menu" "/${BOOT_MENU}"
```

with this text:

```
lulib_copy_to_top_dataset ` /usr/sbin/lucurr ` "$ldme_menu" "/${BOOT_MENU}"
```

2. Rerun the ludelete operation.
 - For more discussion on ludelete failures and how they can be addressed, see this Bob Netherton's blog: http://blogs.sun.com/bobn/entry/getting_rid_of_pesky_live

ZFS Root Pool and Boot Issues

- The boot process can be slow if the boot archive is updated or a dump device has changed. Be patient.
- Do not physically remove a root pool disk without following steps for root pool replacement. See the steps below.
- [CR 6668666 (http://bugs.opensolaris.org/view_bug.do?bug_id=6668666)] - If you attach a disk to create a mirrored root pool after an initial installation, you will need to apply the boot blocks to the secondary disks in the Solaris 10 release. For example:

```
sparc# installboot -F zfs /usr/platform/`uname -i`/lib/fs/zfs/bootblk /dev/rdisk/c0t1d0s0
x86# installgrub /boot/grub/stage1 /boot/grub/stage2 /dev/rdisk/c0t1d0s0
```

- CR 6741743 - The boot -L command doesn't work if you migrated from a UFS root file system. Copy the bootlst command to the correct location. This CR is fixed in the Solaris 10 5/09 release.
- [CR 6696372 (http://bugs.opensolaris.org/view_bug.do?bug_id=6696372)] - The krtld messages from the boot -Z command are harmless. This CR is fixed in the Solaris 10 9/10 release.

Recover from Lost Root Password or Similar Problem

If you need to recover the root password or some similar problem that prevents successful login in a ZFS root environment, you will need to boot failsafe mode on a Solaris 10 system or boot from alternate media, depending on the severity of the error.

Select either of the following recovery methods:

1. On a Solaris 10 system, boot failsafe mode by using the following steps:

- Boot failsafe mode.

```
ok boot -F failsafe
```

- Mount the ZFS BE on /a when prompted:

```
..
..
..
ROOT/zfsBE was found on rpool.
Do you wish to have it mounted read-write on /a? [y,n,?] y
mounting rpool on /a
Starting shell.
```

- Change to the /a/etc directory.

```
# cd /a/etc
```

- Correct the passwd or shadow file.

```
# vi passwd
```

- Reboot the system.

```
# init 6
```

2. On an OpenSolaris system, boot from alternate media by using the following steps:

- Boot from an installation CD or from the network:

```
ok boot cdrom -s
ok boot net -s
```

If you don't use the -s option, you will need to exit the installation program.

- Import the root pool and specify an alternate mount point:

```
# zpool import -R /a rpool
```

- Mount the ZFS BE specifically because canmount is set to noauto by default.

```
# zfs mount rpool/ROOT/zfsBE
```

- Change to the /a/etc directory.

```
# cd /a/etc
```

- Correct the passwd or shadow file.

```
# vi shadow
```

- Reboot the system.

```
# init 6
```

Resolving ZFS Mount Point Problems That Prevent Successful Booting

The best way to change the active boot environment is to use the luactivate command. If booting the active environment fails, due to a bad patch or a configuration error, the only way to boot a different environment is by selecting that environment at boot time. You can select an alternate BE from the GRUB menu on an x86 based system or by booting it explicitly from the PROM on an SPARC based system.

Due to a bug in the Live Upgrade feature in the Solaris 10 10/08 release, the non-active boot environment might fail to boot because the ZFS datasets or the zone's ZFS dataset in the boot environment has an invalid mount point.

The same bug also prevents the BE from mounting if it has a separate /var dataset.

The mount points can be corrected by taking the following steps.

Resolve ZFS Mount Point Problems

1. Boot the system from a failsafe archive.
2. Import the pool.

```
# zpool import rpool
```

3. Review the zfs list output after the pool is imported, looking for incorrect temporary mount points. For example:

```
# zfs list -r -o name,mountpoint rpool/ROOT/s10u6

NAME                                MOUNTPOINT
rpool/ROOT/s10u6                    /.alt.tmp.b-VP.mnt/
rpool/ROOT/s10u6/zones               /.alt.tmp.b-VP.mnt//zones
rpool/ROOT/s10u6/zones/zonerootA    /.alt.tmp.b-VP.mnt/zones/zonerootA
```

The mount point for the root BE (rpool/ROOT/s10u6) should be /.

4. Reset the mount points for the ZFS BE and its datasets.

```
# zfs inherit -r mountpoint rpool/ROOT/s10u6
# zfs set mountpoint=/ rpool/ROOT/s10u6
```

5. Reboot the system. When the option is presented to boot a specific boot environment, either in the GRUB menu or at the OpenBoot Prom prompt, select the boot environment whose mount points were just corrected.

Boot From a Alternate Disk in a Mirrored ZFS Root Pool

You can boot from different devices in a mirrored ZFS root pool.

- Identify the device pathnames for the alternate disks in the mirrored root pool by reviewing the zpool status output. In the example output, disks are c0t0d0s0 and c0t1d0s0.

```
# zpool status
pool: rpool
state: ONLINE
scrub: resilver completed after 0h6m with 0 errors on Thu Sep 11 10:55:28 2008
config:

    NAME          STATE          READ WRITE CKSUM
    rpool         ONLINE         0     0     0
    mirror
      c0t0d0s0    ONLINE         0     0     0
      c0t1d0s0    ONLINE         0     0     0

errors: No known data errors
```

- If you attached the second disk in the mirror configuration after an initial installation, apply the bootblocks after the second disk has resilvered. For example, on a SPARC system:

```
# installboot -F zfs /usr/platform/`uname -i`/lib/fs/zfs/bootblk /dev/rdisk/c0t1d0s0
```

- Depending on the hardware configuration, you might need to update the OpenBoot PROM configuration or the BIOS to specify a different boot device. For example, on a SPARC system:

```
ok setenv boot-device /pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/disk@1
ok boot
```

Activating a BE Fails

- ZFS root pool disks must contain a VTOC label. You will be warned about adding a disk with an EFI label to the root pool.
- See the steps below to detach and relabel the disk with a VTOC label. These steps are also applicable to the Solaris Nevada (SXCE) and Solaris 10 releases.

1. Detach the disk. For example:

```
# zpool detach rpool c0t0d0s0
```

2. Relabel the disk.

```
# format -e c0t1d0s0
format> label
[0] SMI Label
[1] EFI Label
Specify Label type[1]: 0
Ready to label disk, continue? yes
format> quit
```

Make sure all the disk space is in s0. The relabeling process might go back to the default sizing so check to see that all the disk space is where you want it.

3. Attach the disk. For example:

```
# zpool attach rpool c0t0d0s0 c0t1d0s0
```

4. Confirm that the newly attached disk has resilvered completely by using the zpool status command to watch the progress.

```
# zpool status
pool: rpool
state: ONLINE
scrub: resilver completed after 0h8m with 0 errors on Mon Jan 26 10:39:11 2009
config:
```

NAME	STATE	READ	WRITE	CKSUM	
rpool	ONLINE	0	0	0	
mirror	ONLINE	0	0	0	
c0t0d0s0	ONLINE	0	0	0	67.9M resilvered
c0t1d0s0	ONLINE	0	0	0	6.55G resilvered

5. Install the bootblock on the newly attached disk.

```
x86# installgrub /boot/grub/stage1 /boot/grub/stage2 /dev/rdisk/c0t1d0s0
```

6. Confirm that you can boot from the newly attached disk by selecting this disk from the BIOS level.

ZFS Root Pool Recovery

Complete Solaris ZFS Root Pool Recovery

The section describes how to create and restore root pool snapshots starting with Solaris 10 10/08 and recent SXCE releases. The following issues can complicate the root pool snapshot process:

- CR 6462803 - Fixed in the SXCE (Nevada, build 111) release but still open in the Solaris 10 5/09 release, and means that a recursive snapshot of the root pool can fail if more than one BE exists. The workaround is to create a recursive snapshot of the root pool prior to the creation of additional BEs.
- CR 6794452 - Fixed in the SXCE (Nevada, build 107) release and the Solaris 10 5/09 release, but still open in the Solaris 10 10/08 release, which means that you must send and receive individual root pool dataset snapshots. Apply kernel patch 139555-08 for SPARC and 139556-08 for x86 to resolve this CR in the Solaris 10 10/08 release. The resolution of this CR is to use the `zfs receive -u` option when restoring the root pool snapshots even when sending and receiving the entire recursive root pool snapshot.

Different ways exist to send and receive root pool snapshots:

- You can send the snapshots to be stored as files in a pool on a remote system. Confirm that the remote snapshot files can be received on the local system in a test pool. See the first procedure below.
- You can send the snapshots to be stored in a pool on a remote system. You might need to enable `ssh` for root on the remote system if you send the snapshots as root. During a root pool recovery, you will also need to enable remote services so that you can use `rsh` to receive the snapshots back to local system while booted from the miniroot. See the second procedure below.

Validating remotely stored snapshots as files or snapshots is an important step in root pool recovery. In either method, snapshots should be recreated on a routine basis, such as when the pool configuration changes or the Solaris OS is upgraded.

The following procedures have been tested with one ZFS BE.

Before You Begin

For several reasons, it is advisable **not** to use `zfs send` to create backup files for later use. It is advisable instead, to pipe `zfs send` directly into `zfs receive` on another disk, or another machine. This implies that you have another machine, or external storage, with ZFS filesystem(s) on them, which are suitable to receive onto. Furthermore, if you are restoring from bare metal, you will be restoring in a minimal CD boot environment, which does not include `ssh`. So the bare metal recovery process is fundamentally more complex if your backup image is piped into `zfs receive` immediately, as advised, on a separate machine. It is beyond the scope of this document. This document is written as if you will save your `zfs send` datastream to a file. There are many solutions (design your own) which enable you to pipe `zfs send` directly into `zfs receive` ... You could use a locally attached removable disk ... You could send to a remote system, and later if needed, the remote system could temporarily send to a file for immediate consumption ... Or many other options.

- A `zfs receive` operation is all-or-nothing. If there is any bit error in the datastream, the whole datastream is invalid. If such an error occurs, you want to discover it immediately. You want to avoid a false sense of security, assuming your data stream is safely stored on disk or tape, if it was silently

corrupted. Therefore, it's desirable for your "zfs send" to pipe directly into "zfs receive" for instant checksum verification.

- A "zfs receive" operation is all-or-nothing. You cannot select individual files from the datastream to recover. In most recovery situations, you do not need to restore onto bare metal. Usually it's useful if you can just get that one file or directory that was accidentally deleted. Therefore it's desirable for your "zfs send" to pipe directly into "zfs receive"

Because you will restore from a boot CD, there are only a few compression options available to you: 7z, bzip2, and gzip. The general purpose clear winner is 7z with compression level 1. It's about as fast as gzip, with compression approx 2x stronger than bzip2

In the rpool recovery instructions that follow, we are omitting the compression/decompression commands because it simplifies the documentation, and is not applicable in all situations. (You probably won't use gzip/bzip2/7z if you're piping "zfs send" directly into "zfs receive" as advised.) At your own discretion, you may use compression commands similar to the following examples, to enhance your experience:

```
-----
zfs send rpool@uniquebackupstring | 7z a -mx=1 -si /mnt/rpool.zfssend.7z
7z x /mnt/rpool.zfssend.7z -so | zfs receive -F rpool

zfs send rpool@uniquebackupstring | gzip --fast > /mnt/rpool.zfssend.gz
gunzip -c /mnt/rpool.zfssend.gz | zfs receive -F rpool

zfs send rpool@uniquebackupstring | bzip2 > /mnt/rpool.zfssend.bz2
bunzip2 /mnt/rpool.zfssend.bz2 | zfs receive -F rpool
-----
```

Because you will restore from a boot CD, the media needs to be located somewhere that can be accessed from a CD boot environment, which does not include ssh/scp. The obvious choice is NFS. Be aware that solaris NFS clients often have compatibility problems with linux NFS servers. You can do it, but it takes more effort. It is easier to configure a Solaris NFS server to serve files to a Solaris NFS client.

I am assuming there is a solaris NFS server available because it makes my job easy while I'm writing this. ;-)

Note: You could just as easily store the backup on removable disk in something like a removable zfs drive. Just make sure whatever way you store it, it's accessible from the CD boot environment, which might not support a later version of zpool etc.

Create and Send Root Pool Snapshots

Suppose the machine "hansel" will send snapshots to its partner "gretel." Suppose a directory "/backupdir" has already been created on gretel, with enough empty space to store hansel's snapshots.

Be sure reverse-DNS resolves hansel's IP address to "hansel" or else NFS will fail. You can easily solve this by editing the /etc/hosts file on gretel.

1. Share space from gretel to hansel. Do this on gretel:

```
-----
gretel# share -F nfs -o rw=hansel,root=hansel /backupdir
-----
```

2. Mount that space to make it available on hansel:

```
-----
hansel# mount gretel:/backupdir /mnt
-----
```

3. For the heck of it, you might as well capture some extra information:

```
-----
echo "" > /mnt/zpool-properties.txt
for pool in `zpool list | grep -v '^NAME ' | sed 's/ .*//'\` ; do
  echo "-----" | tee -a /mnt/zpool-properties.txt
  echo "zpool get all $fs" | tee -a /mnt/zpool-properties.txt
done
echo "" > /mnt/zfs-properties.txt
for fs in `zfs list | grep -v '@' | grep -v '^NAME ' | sed 's/ .*//'\` ; do
  echo "-----" | tee -a /mnt/zfs-properties.txt
  echo "zfs get all $fs" | tee -a /mnt/zfs-properties.txt
done
-----
```

```
done
```

4. Create a recursive snapshot of the root pool. Do this on hansel:

```
hansel# zfs snapshot -r rpool@uniqueSnapshotName
hansel# zpool set listsnapshots=on rpool
hansel# zfs list
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
rpool	17.6G	116G	67K	/rpool
rpool@uniqueSnapshotName			0	- 67K -
rpool/ROOT	5.43G	116G	21K	legacy
rpool/ROOT@uniqueSnapshotName			0	- 21K -
rpool/ROOT/osolBE	5.43G	116G	4.68G	/
rpool/ROOT/osolBE@uniqueSnapshotName			290K	- 4.68G -
rpool/dump	4.00G	116G	4.00G	-
rpool/dump@uniqueSnapshotName			0	- 4.00G -
rpool/export	69.5K	116G	23K	/export
rpool/export@uniqueSnapshotName			0	- 23K -
rpool/export/home	46.5K	116G	23K	/export/home
rpool/export/home@uniqueSnapshotName			0	- 23K -
rpool/export/home/admin	23.5K	116G	23.5K	/export/home/admin
rpool/export/home/admin@uniqueSnapshotName			0	- 23.5K -
rpool/swap	8.20G	124G	15.2M	-
rpool/swap@uniqueSnapshotName			0	- 15.2M -

5. You may either send the individual snapshots, or send recursive snapshots.

- Send the individual snapshots to the remote system. This method is required for a Solaris 10 10/08 (sol_10u6) and earlier systems that do not have the above mentioned kernel patch level. Furthermore, this method provides the following advantages:
 - Allows recovery of individual datasets rather than the entire root pool should that be desired.
 - Does not send the dump and swap device snapshots, which is not necessary and adds time to the recursive method. But it does require manual creation of these datasets as outlined in the steps below.

```
hansel# zfs send rpool@uniqueSnapshotName > /mnt/rpool_uniqueSnapshotName
hansel# zfs send rpool/ROOT@uniqueSnapshotName > /mnt/rpool_ROOT_uniqueSnapshotName
hansel# zfs send rpool/ROOT/osolBE@uniqueSnapshotName > /mnt/rpool_ROOT_osolBE_uniqueS
hansel# zfs send rpool/export@uniqueSnapshotName > /mnt/rpool_export_uniqueSnapshotName
hansel# zfs send rpool/export/home@uniqueSnapshotName > /mnt/rpool_export_home_uniqueS
hansel# zfs send rpool/export/home/admin@uniqueSnapshotName > /mnt/rpool_export_home_a
```

- If you are running the SXCE build 107 or the Solaris 10 5/09 (sol_10u7) or later release, you can send the entire recursive snapshot.

```
# zfs send -R rpool@uniqueSnapshotName > /mnt/rpool.uniqueSnapshotName
```

Bare Metal Receive Root Pool Snapshots

In this scenario, assume the following conditions:

- ZFS root pool cannot be recovered
- ZFS root pool snapshots are stored on a remote system and shared over NFS
- The system is booted from an equivalent Solaris release to the root pool version so that the Solaris release and the pool version match. Otherwise, you will need to add the `-o version=version-number` property option and value when you recreate the root pool in step 5 below.

1. Boot from CD/DVD or the network.

- Sparc

```
ok boot net
```

or

```
ok boot cdrom
```

Then, exit out of the installation program.

- x86

Boot from the solaris installation disc.

Select "6. Single user shell"

2. You probably need to bring up the network interface, something like this:

```
ifconfig -a plumb
ifconfig -a
```

(Notice the name of the network adapter. In this example, it's e1000g0)

```
ifconfig e1000g0 192.168.1.100/24 up
```

3. Mount the remote snapshot dataset. (Assuming the server which has the snapshots is 192.168.1.100)

```
# mount -F nfs 192.168.1.100:/backupdir /mnt
```

4. If the root pool disk is replaced and does not contain a disk label that is usable by ZFS, you will have to relabel the disk. For more information, see [#Replacing/Relabeling_the_Root_Pool_Disk](#).
5. Recreate the root pool. For example:

```
# zpool create -f -o failmode=continue -R /a -m legacy rpool c1t1d0s0
```

6. If you had to replace or relabel the disk, then you might need to reinstall the boot blocks. For example:
 - Sparc:

```
# installboot -F zfs /usr/platform/`uname -i`/lib/fs/zfs/bootblk /dev/rdisk/c1t1d0s0
```

- x86:

```
# installgrub /boot/grub/stage1 /boot/grub/stage2 /dev/rdisk/c1t1d0s0
```

7. Select one of the following if you are running the Solaris 10 10/08, Solaris 10 5/09, or the Nevada, build 107 or later release:

- Solaris 10 10/08 or Nevada, prior to build 107: Receive the individual root pool snapshots. This step might take some time. Notice, the first one requires -F because it already exists. The rest do not need -F. For example:

```
# cat /mnt/rpool_uniqueSnapshotName | zfs receive -F rpool
cat /mnt/rpool_ROOT_uniqueSnapshotName | zfs receive rpool/ROOT
cat /mnt/rpool_ROOT_osolBE_uniqueSnapshotName | zfs receive rpool/ROOT/osolBE
cat /mnt/rpool_export_uniqueSnapshotName | zfs receive rpool/export
cat /mnt/rpool_export_home_uniqueSnapshotName | zfs receive rpool/export/home
cat /mnt/rpool_export_home_admin_uniqueSnapshotName | zfs receive rpool/export/home/ad
```

- Solaris 10 10/08 with above-mentioned kernel patch, Solaris 10 5/09 or later, Nevada, build 107 or later: Receive the recursive root pool snapshot. This step might take some time. For example:

```
# cat /mnt/rpool.uniqueSnapshotName | zfs receive -Fd rpool
```

8. Set the bootfs property on the root pool BE.

```
# zpool set bootfs=rpool/ROOT/osolBE rpool
```

9. Recreate the dump device. This step is not necessary when using the recursive restore method because it is recreated with the rpool.

```
# zfs create -V 2G rpool/dump
```

10. Recreate the swap device. This step is not necessary when using the recursive restore method because it is recreated with the rpool.

```
SPARC# zfs create -V 2G -b 8k rpool/swap
x86# zfs create -V 2G -b 4k rpool/swap
```

11. Reboot the system.

```
# init 6
```

Replacing/Relabeling the Root Pool Disk

You might need to replace a disk in the root pool for the following reasons:

- The root pool is too small and you want to replace it with a larger disk
- The root pool disk is failing. If the disk is failing so that the system won't boot, you'll need to boot from an alternate media, such as a CD or the network, before you replace the root pool disk.

Part of recovering the root pool might be to replace or relabel the root pool disk. Follow the steps below to relabel and replace the root pool disk.

1. Physically attach the replacement disk.
2. If the replacement disk has an EFI label, the fdisk output looks similar to the following on an x86 system.

```
# fdisk /dev/rdisk/c1t1d0p0
selecting c1t1d0p0
Total disk size is 8924 cylinders
Cylinder size is 16065 (512 byte) blocks

          Cylinders
Partition  Status   Type          Start   End   Length   %
=====  =====  =====
          1          EFI           0   8924   8925   100

Enter Selection: 6
```

Use fdisk to change this to a Solaris partition.

3. Select one of the following to create a Solaris fdisk partition for a disk on an x86 system or create an SMI label for a disk on a SPARC system.
 - On an x86 system, create a Solaris fdisk partition that can be used for booting by selecting 1=SOLARIS2. You can create a Solaris partition by using the fdisk -B option that creates one Solaris partition that uses the whole disk. Beware that the following command uses the whole disk.

```
# fdisk -B /dev/rdisk/c1t1d0p0
```

Display the newly created Solaris partition. For example:

```
Total disk size is 8924 cylinders
```

```

                Cylinder size is 16065 (512 byte) blocks

                Cylinders
Partition      Status   Type           Start   End   Length   %
=====      =====
                1      Active   Solaris2      1     8923    8923    100

Enter Selection: 6

```

- On a SPARC based system, make sure you have an SMI label. Use the format -e command to determine if the disk label is EFI or SMI and relabel the disk, if necessary. In the output below, the disk label includes sectors and not cylinders. This is an EFI label.

```

# format -e
Searching for disks...done
AVAILABLE DISK SELECTIONS:
  0. c1t0d0 <SUN36G cyl 24620 alt 2 hd 27 sec 107>
     /pci@8,600000/SUNW,qlc@4/fp@0,0/ssd@w21000004cf7fac8a,0
  1. c1t1d0 <SUN36G cyl 24620 alt 2 hd 27 sec 107>
     /pci@8,600000/SUNW,qlc@4/fp@0,0/ssd@w21000004cf7fad21,0
Specify disk (enter its number): 1
Selecting c1t1d0
[disk formatted]
format> p
partition> p
Current partition table (original):
Total disk sectors available: 71116541 + 16384 (reserved sectors)

Part      Tag      Flag      First Sector      Size      Last Sector
-----
  0        usr      wm          34      33.91GB      71116541
  1 unassigned  wm          0         0             0
  2 unassigned  wm          0         0             0
  3 unassigned  wm          0         0             0
  4 unassigned  wm          0         0             0
  5 unassigned  wm          0         0             0
  6 unassigned  wm          0         0             0
  7 unassigned  wm          0         0             0
  8 reserved   wm      71116542      8.00MB      71132925

partition> label
[0] SMI Label
[1] EFI Label
Specify Label type[1]: 0
Auto configuration via format.dat[no]?
Auto configuration via generic SCSI-2[no]?
partition>

```

4. Create a slice in the Solaris partition for the root pool. Creating a slice on x86 and SPARC system is similar except that an x86 system has a slice 8. In the example below, a slice 0 is created and the disk space is allocated to slice 0 on an x86 system. For a SPARC system, just ignore the slice 8 input.

```

# format
Specify disk (enter its number): 1
Selecting c1t1d0
[disk formatted]
FORMAT MENU:
  disk          - select a disk
  type          - select (define) a disk type
  partition     - select (define) a partition table
  current       - describe the current disk
  format        - format and analyze the disk
  fdisk        - run the fdisk program

format> p
PARTITION MENU:
  0 - change `0' partition
  1 - change `1' partition
  2 - change `2' partition
  3 - change `3' partition
  4 - change `4' partition
  5 - change `5' partition
  6 - change `6' partition
  7 - change `7' partition
select - select a predefined table
modify - modify a predefined partition table

```

```

    name - name the current table
    print - display the current table
    label - write partition map and label to the disk
    !<cmd> - execute <cmd>, then return
    quit
partition> p
Current partition table (original):
Total disk cylinders available: 8921 + 2 (reserved cylinders)

Part      Tag      Flag      Cylinders      Size      Blocks
0 unassigned wm        0              0          (0/0/0)      0
1 unassigned wm        0              0          (0/0/0)      0
2 backup  wu        0 - 8920      68.34GB    (8921/0/0) 143315865
3 unassigned wm        0              0          (0/0/0)      0
4 unassigned wm        0              0          (0/0/0)      0
5 unassigned wm        0              0          (0/0/0)      0
6 unassigned wm        0              0          (0/0/0)      0
7 unassigned wm        0              0          (0/0/0)      0
8 boot    wu        0 - 0         7.84MB    (1/0/0)     16065
9 unassigned wm        0              0          (0/0/0)      0
partition> modify
Select partitioning base:
  0. Current partition table (original)
  1. All Free Hog
Choose base (enter number) [0]? 1

Part      Tag      Flag      Cylinders      Size      Blocks
0 root     wm        0              0          (0/0/0)      0
1 swap    wu        0              0          (0/0/0)      0
2 backup  wu        0 - 8920      68.34GB    (8921/0/0) 143315865
3 unassigned wm        0              0          (0/0/0)      0
4 unassigned wm        0              0          (0/0/0)      0
5 unassigned wm        0              0          (0/0/0)      0
6 usr     wm        0              0          (0/0/0)      0
7 unassigned wm        0              0          (0/0/0)      0
8 boot    wu        0 - 0         7.84MB    (1/0/0)     16065
9 alternates wm        0              0          (0/0/0)      0

Do you wish to continue creating a new partition
table based on above table[yes]?
Free Hog partition[6]? 0
Enter size of partition '1' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '3' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '4' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '5' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '6' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '7' [0b, 0c, 0.00mb, 0.00gb]:

Part      Tag      Flag      Cylinders      Size      Blocks
0 root     wm        1 - 8920      68.33GB    (8920/0/0) 143299800
1 swap    wu        0              0          (0/0/0)      0
2 backup  wu        0 - 8920      68.34GB    (8921/0/0) 143315865
3 unassigned wm        0              0          (0/0/0)      0
4 unassigned wm        0              0          (0/0/0)      0
5 unassigned wm        0              0          (0/0/0)      0
6 usr     wm        0              0          (0/0/0)      0
7 unassigned wm        0              0          (0/0/0)      0
8 boot    wu        0 - 0         7.84MB    (1/0/0)     16065
9 alternates wm        0              0          (0/0/0)      0

Okay to make this the current partition table[yes]?
Enter table name (remember quotes): "disk1"

Ready to label disk, continue? yes
partition> q
format> q

```

5. Attach the replacement disk.

```

# zpool attach rpool c1t0d0s0 c1t1d0s0
Please be sure to invoke installgrub(1M) to make 'c1t1d0s0' bootable.

```

6. Check the resilvering status of the newly attached disk.

```

# zpool status rpool

```

7. After the disk resilvering is complete, install the boot blocks.


```
x86# installgrub /boot/grub/stage1 /boot/grub/stage2 /dev/rdsk/clt1d0s0
```

```
sparc# installboot -F zfs /usr/platform/`uname -i`/lib/fs/zfs/bootblk /dev/rdsk/clt1d0s0
```

8. Confirm that you can boot from the replacement disk.
9. Detach the smaller or unneeded disk.

```
# zpool detach rpool clt0d0s0
```

Rolling Back a Root Pool Snapshot From a Failsafe Boot

This procedure assumes that existing root pool snapshots are available. In this example, the root pool snapshots are available on the local system.

```
# zfs snapshot -r rpool@0730
# zfs list
NAME                                USED  AVAIL  REFER  MOUNTPOINT
rpool                               6.87G 60.1G   37K    /rpool
rpool@0730                          18K   -       37K    -
rpool/ROOT                          3.87G 60.1G   18K    legacy
rpool/ROOT@0730                     0     -       18K    -
rpool/ROOT/zfs1008BE                3.87G 60.1G   3.82G  /
rpool/ROOT/zfs1008BE@0730          52.3M -       3.81G  -
rpool/dump                          1.00G 60.1G  1.00G  -
rpool/dump@0730                    16K   -       1.00G  -
rpool/export                        52K   60.1G   19K    /export
rpool/export@0730                   15K   -       19K    -
rpool/export/home                   18K   60.1G   18K    /export/home
rpool/export/home@0730              0     -       18K    -
rpool/swap                          2.00G 62.1G   16K    -
rpool/swap@0730                     0     -       16K    -
```

1. Shutdown the system and boot failsafe mode.

```
ok boot -F failsafe
Multiple OS instances were found. To check and mount one of them
read-write under /a, select it from the following list. To not mount
any, select 'q'.

  1  /dev/dsk/c0t1d0s0                Solaris 10 xx SPARC
  2  rpool:5907401335443048350       ROOT/zfs1008

Please select a device to be mounted (q for none) [?,??,q]: 2
mounting rpool on /a
```

2. Roll back the individual root pool snapshots.

```
# zfs rollback rpool@0730
# zfs rollback rpool/ROOT@0730
# zfs rollback rpool/ROOT/zfs1008BE@0730
# zfs rollback rpool/export@0730
.
```

Current ZFS snapshot rollback behavior is that recursive snapshots are not rolled back with the `-r` option. You must roll back the individual snapshots from the recursive snapshot.

3. Reboot back to multiuser mode.

```
# init 6
```

Primary Mirror Disk in a ZFS Root Pool is Unavailable or Fails

- If the primary disk in the pool fails, you might need to boot from the secondary disk by

specifying the boot path. For example, on a SPARC system, a `devalias` is available to boot from the second disk as `disk1`.

```
ok boot disk1
```

- In some cases, you might need to remove the failed boot disk from the system to boot successfully from the other disk in the mirror.
- If the system supports hot-plugging you can attempt to do a live replacement of a failed disk. On some systems, you might have to offline and unconfigure the disk before you physically replace it.

```
# zpool offline rpool c0t0d0s0
# cfgadm -c unconfigure c1::dsk/c0t0d0
```

- Physically replace the primary disk. For example, `c0t0d0s0`.
- Put a VTOC label on the new disk with `format -e`.
- Reconfigure the disk and bring it online, if necessary. For example:

```
# cfgadm -c configure c1::dsk/c0t0d0
# zpool online rpool c0t0d0
```

- Let ZFS know the primary disk was physically replaced at the same location.

```
# zpool replace rpool c0t0d0s0
```

- If the `zpool replace` step fails, detach and attach the primary mirror disk:

```
# zpool detach rpool c0t0d0s0
# zpool attach rpool c0t1d0s0 c0t0d0s0
```

- Confirm that the disk is available and the disk is resilvered.

```
# zpool status rpool
```

- After the resilvering is complete, replace the bootblocks on the primary disk.

```
SPARC# installboot -F zfs /usr/platform/`uname -i`/lib/fs/zfs/bootblk /dev/rdisk/c0t0d0s0
x86# installgrub /boot/grub/stage1 /boot/grub/stage2 /dev/rdisk/c0t0d0s0
```

- Confirm that you can boot from the primary disk.

ZFS Swap and Dump Devices

During an initial installation or a Live Upgrade migration, a swap volume and dump volume are created. The default sizes of the swap and dump volumes that are created by the Solaris installation program are as follows:

- Swap volume size - Systems with 4 GB-128 GB memory, swap volume size is calculated in the 1-4 GB range.
- Dump volume size - Generally calculated as 1/2 of total memory size. A busy system with many active ZFS file systems might use 1/2 to 3/4 the size of physical memory for the size of the dump device.

You can adjust the sizes of your swap and dump volumes in a JumpStart profile or during an initial installation to sizes of your choosing as long as the new sizes support system operation.

```
# zfs list
```

```

NAME                USED  AVAIL  REFER  MOUNTPOINT
rpool                5.66G  27.6G  21.5K  /rpool
rpool/ROOT           4.65G  27.6G   18K   /rpool/ROOT
rpool/ROOT/zfs1008BE 4.65G  27.6G  4.65G  /
rpool/dump           515M  27.6G  515M   -
rpool/swap           513M  28.1G   16K   -

```

Resizing ZFS Swap and Dump Devices

- You can adjust the size of your swap and dump volumes during an initial installation.
- You can create and size your swap and dump volumes before you do a Solaris Live Upgrade operation. ZFS dump volume performance is better when the volume is created with a 128-KB block size. In SXCE, build 102, ZFS dump volumes are automatically created with a 128-KB block size (CR 6725698). For example:

```

# zpool create rpool mirror c0t0d0s0 c0t1d0s0
/* The Solaris 10 10/08 dump creation syntax would be:
# zfs create -V 2G -b 128k rpool/dump
/* The SXCE build 102 dump creation syntax would be:
# zfs create -V 2G rpool/dump
SPARC# zfs create -V 2G -b 8k rpool/swap
x86# zfs create -V 2G -b 4k rpool/swap

```

- Activate your swap device, if necessary:

```
# swap -a /dev/zvol/dsk/rpool/swap
```

- Add the swap entry to the /etc/vfstab file similar to the following, if necessary:

```
/dev/zvol/dsk/rpool/swap    -            -            swap    -            no
```

- Activate your dump device, if necessary:

```
# dumpadm -d /dev/zvol/dsk/rpool/dump
```

- Solaris Live Upgrade does not resize existing swap and dump volumes. You can reset the volsize property of the swap and dump devices after a system is installed. For example:

```

# zfs set volsize=2G rpool/dump
# zfs get volsize rpool/dump
NAME                PROPERTY  VALUE      SOURCE
rpool/dump          volsize   2G         -

```

- You can adjust the size of the swap and dump volumes in a JumpStart profile by using profile syntax similar to the following:

```

install_type initial_install
cluster SUNWCXall
pool rpool 16g 2g 2g c0t0d0s0

```

In this profile, the 2g and 2g entries set the size of the swap area and dump device a

- You can adjust the size of your dump volume, but it might take some time, depending on the size of the dump volume. For example:

```

# zfs set volsize=2G rpool/dump
# zfs get volsize rpool/dump
NAME                PROPERTY  VALUE      SOURCE
rpool/dump          volsize   2G         -

```

Adjusting the Size of the Swap Volume on an Active System

If you need to adjust the size of the swap volume after installation on an active system, review the following steps. See CR 6765386 for more information.

1. If your swap device is in use, then you might not be able to delete it. Check to see if the swap area is in use. For example:

```
# swap -l
swapfile          dev  swaplo  blocks  free
/dev/zvol/dsk/rpool/swap 182,2      8  4194296  4194296
```

In the above output, blocks == free, so the swap device is not actually being used.

2. If the swap area is not in use, remove the swap area. For example:

```
# swap -d /dev/zvol/dsk/rpool/swap
```

3. Confirm that the swap area is removed.

```
# swap -l
No swap devices configured
```

4. Resize the swap volume. For example:

```
# zfs set volsize=1G rpool/swap
```

5. Activate the swap area.

```
# swap -a /dev/zvol/dsk/rpool/swap
# swap -l
swapfile          dev  swaplo  blocks  free
/dev/zvol/dsk/rpool/swap 182,2      8  2097136  2097136
```

The swap -a attempt might fail if the swap area is already listed in /etc/vfstab or is in use by Live Upgrade. In this case, use the swapadd feature instead.

```
# /sbin/swapadd
# swap -l
swapfile          dev  swaplo  blocks  free
/dev/zvol/dsk/rpool/swap 256,1     16  2097136  2097136
```

If you need to add swap space but removing an existing swap device is difficult on a busy system, add another swap volume. For example:

```
# zfs create -V 2G rpool/swap1
# swap -a /dev/zvol/dsk/rpool/swap1
# swap -l
swapfile          dev  swaplo  blocks  free
/dev/zvol/dsk/rpool/swap 256,1     16  2097136  2097136
/dev/zvol/dsk/rpool/swap1 256,5     16  4194288  4194288
Add an entry for the second swap volume to the /etc/vfstab file.
```

Destroying an Inactive Root Pool With an Active Dump/Swap Device

If you want to destroy a ZFS root pool that is no longer needed, but it still has an active dump device and swap area, consider the following:

- Boot from alternate media, import the inactive pool and destroy it
- Or, use the swap command to create a second swap volume for the new root pool and use the dumpadm to reset the dump device to the new pool. After the previous dump and swap devices are no longer in use, you can remove them.

```
# zfs create -V 2G rpool2/swap
# swap -a /dev/zvol/dsk/rpool2/swap
# zfs create -V 2G rpool2/dump
# dumpadm -d /dev/zvol/dsk/rpool2/dump
< destroy the inactive root pool >
```

- CR 6910925 prevents you from removing an existing dump device.

Retrieved from "http://www.solarisinternals.com/wiki/index.php/ZFS_Troubleshooting_Guide"

- This page was last modified 21:56, 3 October 2011.