

Ignite-UX and MirrorDisk/UX



Abstract.....	2
Introduction.....	2
Deployment.....	2
System Recovery.....	2
How to Deploy Mirrored Disk Systems with Ignite-UX.....	3
How to mirror the root disk on a HP 9000 system.....	3
How to mirror the root disk on an HP Integrity Server.....	4
How to Recover Mirrored Disk Systems with Ignite-UX using make_tape_recovery.....	8
How to Recover Mirrored Disk Systems with Ignite-UX via make_net_recovery.....	8
An Example config.local File.....	9
Glossary.....	13
For More Information.....	14

Abstract

It is a common misconception that Ignite-UX cannot be used with systems containing mirrored disks. While it is true that Ignite-UX does not have any built-in support for mirrored disks, it is not true that Ignite-UX cannot be used on systems with mirrored disks. Ignite-UX is still a very valuable tool for use on systems configured with LVM mirroring.

This white paper has two primary goals:

- Explain benefits and limitations of using Ignite-UX with LVM mirroring.
- Explain how to use Ignite-UX with LVM mirroring.

Introduction

Ignite-UX is a tool set used for the deployment (cold installation) and recovery of HP-UX on HP systems. The `make_tape_recovery` and `make_net_recovery` tools are used to build a customized recovery image that can be used in the event of a root volume group failure.

MirrorDisk/UX must be installed on a system to enable LVM mirroring functionality. None of the Ignite-UX tools are "aware" of mirrored disks.

Deployment

In the case of cold installation (deployment), the fact that Ignite-UX is not aware of mirrored disks means:

- When describing how your system's disks will be configured using the Ignite-UX user interface, it is not possible to specify any disk mirroring information.
- There are no Ignite-UX configuration file keywords that allow you to specify any disk mirroring.

It is, of course, possible to use Ignite-UX to cold install your system. You can then set up your system's mirrored disks with the LVM commands, or by using SAM (for all volume groups except the root volume group).

System Recovery

In the case of recovery (`make_tape_recovery` and `make_net_recovery`), the fact that Ignite-UX is not aware of mirrored disks means that the system being recovered may have had mirrored disks prior to the problem that forced the system recovery. Of course, with a mirrored root disk you are much less likely to ever need to recover a failed root disk using `make_tape_recovery` or `make_net_recovery`. Nonetheless, mirroring your root disk does not cover all of the cases that system recovery can handle. For example:

- There has been a catastrophic software error on the root disk that has also affected the mirror. For example, executing `rm -rf *` while in the root (`/`) directory.
- The primary disk has a hardware failure and the mirror also fails before the primary can be replaced.
- You are about to update your system to a new operating system release or add new patches to the system. Creating a recovery archive allows you to get back to a working system if the update does not succeed.

During such a recovery, Ignite-UX will not automatically re-create the LVM mirrors. The mirrors need to be created via LVM commands or SAM after the recovery is complete.

Note:

Later in this white paper we will be discussing how you can add configuration into a recovery archive to automatically reestablish LVM mirroring of the root volume group.

How to Deploy Mirrored Disk Systems with Ignite-UX

When you're specifying how your disks should be configured in the Ignite-UX user interface, any disks that will eventually be mirrors should not be specified.

Once the cold install process has completed, you can set up your disk mirrors via the SAM command or by using the LVM commands directly. Note that SAM cannot be used to mirror the root disk.

It is important to understand the steps involved in mirroring the root volume group on HP 9000 and HP Integrity servers in order to apply these concepts to the use of Ignite-UX with mirrored disks.

How to mirror the root disk on a HP 9000 system

1. Run the `pvcreate` command on the disk that you will be mirroring to, you must use the `-B` option to make this disk bootable, it is a common mistake when mirroring the root volume group to not use the `-B` option when running `pvcreate` on the alternate boot disk.

Note:

If your setup requires more than one disk to complete mirroring, the second and any subsequent disks should not have `pvcreate` executed on them with the `-B` option. You may need to use the `-f` option to force `pvcreate` to operate on the disk if it has been used by LVM previously.

```
# pvcreate -B /dev/rdisk/<second disk>
```

2. Extend the disk into the root volume group, at this point you should also extend any other disks required into the root volume group.

```
# vextend /dev/vg00 /dev/dsk/<second disk>
```

3. The `mkboot` command must be run on the alternate boot disk so that a copy of the primary and secondary loader (and other things) are placed into the LIF¹ space that was reserved by using the `-B` option to `pvcreate` in Step 1.

```
# mkboot /dev/dsk/<second disk>
```

4. Update the boot string in the `AUTO` file in the LIF on the alternate boot disk. The `-lq` should be placed in the `AUTO` file to ensure that the system will attempt to boot without quorum. In the event that the primary boot disk fails, this will allow the alternate boot disk to boot if it is not available.

```
# mkboot -a "hpux -lq" /dev/rdisk/<second disk>>
```

5. Mirror the disks by extending all of the logical volumes, in order, onto the alternate boot disk.

¹ Logical Interchange Format, see manual page `lif(4)` for more information.

Note:

The boot, root and primary swap/dump filesystems must be contiguous. If you need to mirror to more than one disk, you may need to add a second disk or change the name of the disk being used when extending the logical volumes.

The boot, root and primary swap/dump filesystems should be mirrored before any other filesystem is mirrored.

```
# lvextend -m 1 /dev/vg00/lvol1 /dev/dsk/<second disk>
# lvextend -m 1 /dev/vg00/lvol2 /dev/dsk/<second disk>
# lvextend -m 1 /dev/vg00/lvol3 /dev/dsk/<second disk>
# lvextend -m 1 /dev/vg00/lvol4 /dev/dsk/<second disk>
# lvextend -m 1 /dev/vg00/lvol5 /dev/dsk/<second disk>
.
.
.
```

6. Reestablish the location of boot, root and primary swap/dump filesystems within LVM using (or modifying as needed) the following commands:

```
# lvlnboot -b /dev/vg00/lvol1
# lvlnboot -r /dev/vg00/lvol3
# lvlnboot -s /dev/vg00/lvol2
# lvlnboot -d /dev/vg00/lvol2
# lvlnboot -R
```

7. You must add the boot disk to the file, /stand/bootconf. A new line must be inserted in this file when using LVM and it should start with an "1" (lower case letter 1), and then followed by the block disk device of the alternate boot disk. This is to ensure that when software is installed the contents of the boot LIF both disks are updated.

```
# cat /stand/bootconf

1 /dev/dsk/<boot disk>
```

After creating an alternate boot disk using LVM, HP recommends that you test that it can boot and operates as expected.

How to mirror the root disk on an HP Integrity Server

1. Determine the how the current boot disk is currently partitioned. You must make the EFI and HPSP² partitions the same size as on the current boot disk. You can determine the size of the different partitions using the following commands:

```
# diskinfo -b /dev/rdisk/<boot disk>s1 | awk '{print $1 / (1024)}'
# diskinfo -b /dev/rdisk/<boot disk>s2 | awk '{print $1 / (1024)}'
```

² This is the HP Service Partition, you should always let this partition be created when installing an HP Integrity server as it holds the offline diagnostics for HP Integrity systems.

```
# diskinfo -b /dev/rdisk/<boot disk>s3 | awk '{print $1 / (1024)}'
```

2. The three commands in the previous step show the size in Megabytes of the EFI partition (section 1), the HP-UX partition (section 2), and the HPSP partition (section 3). Place those values into the following command:

```
# print '3\nEFI 500MB\nHPUX *****MB\nHPSP 400MB' > /tmp/partitionfile
```

Or you can choose to create the `/tmp/partitionfile` file using `vi` when performing this interactively.

Note:

If you do not have a HPSP partition, the initial 3 in the command should be 2 and the command would instead look like:

```
# print '2\nEFI 500MB\nHPUX *****MB\n' > \n/tmp/partitionfile
```

If the boot disk and the alternate boot disk are different sizes and you do not wish to calculate the size of the HP-UX partition, you could execute the `print` command as follows:

```
# print '3\nEFI 500MB\nHPUX 100%\nHPSP 400MB' > /tmp/partitionfile
```

This creates the EFI and HPSP partitions at a fixed size and allocates 100% of the remaining space to the HP-UX partition.

3. The alternate boot disk must be partitioned using the file `/tmp/partitionfile` created in the previous step. This action is performed using the `idisk` command and `echo yes` is used to answer all of the `idisk` queries. You may choose not to include this part of the command and answer the question manually when performing this task interactively. Remove the partition file created previously as follows:

```
# echo yes | idisk -wf /tmp/partitionfile /dev/rdisk/<second disk>\n# rm /tmp/partitionfile
```

4. After creating the partitions on the disk you will need to create the device files associated with the partitions:

```
# insf -e -H <hardware path to second disk>
```

5. Create the disk as a bootable LVM disk. The use of `s2` at the end of the disk device is needed since LVM uses the HP-UX partition so you must use this section of the disk as well. If you were to use just the disk device, it would overwrite the partition information that has been created on the disk. If this disk has been previously used by LVM in this way you may need to add the `-f` option to force the initialization of this disk.

```
# pvcreate -B /dev/rdisk/<second disk>s2
```

6. Now, run `mkboot` to complete the setup of the disk as a boot disk.

```
# mkboot -e -l /dev/dsk/<second disk>
```

The `-e` option sets the use of an EFI layout and the `-l` option sets this volume to be used by a volume manager (even if it is not currently used by one).

Note:

If no partition number is given to `mkboot`, the command examines the whole disk.

7. Initialize the HPSP partition with an EFI filesystem. This does not place any data into the partition; it simply initializes HPSP partition with an EFI filesystem. Proceed to the next step if you are not creating a HPSP partition or execute the following command.

```
# efi_fsinit -d /dev/rdisk/<second disk>s3
```

8. Extend the HP-UX partition into the root volume group. The HP-UX partition (section 2) is extended into the volume group *not* the whole disk.

```
# vgextend /dev/vg00 /dev/dsk/<second disk>s2
```

9. Extend the mirrors to the alternate boot disk. There may be varying numbers of logical volumes to extend and the names may be different to those indicated in the following commands so you should modify these commands as needed.

Note:

The `root`, primary `swap/dump`, and the `boot` filesystems should be mirrored before any other filesystem is mirrored.

```
# lvextend -m 1 /dev/vg00/lvol1 /dev/dsk/<second disk>s2
# lvextend -m 1 /dev/vg00/lvol2 /dev/dsk/<second disk>s2
# lvextend -m 1 /dev/vg00/lvol3 /dev/dsk/<second disk>s2
# lvextend -m 1 /dev/vg00/lvol4 /dev/dsk/<second disk>s2
# lvextend -m 1 /dev/vg00/lvol5 /dev/dsk/<second disk>s2
.
.
.
```

10. Reestablish the location of the `root`, primary `swap/dump`, and the `boot` filesystems within LVM using the following commands modifying the logical volumes if necessary:

```
# lvolboot -b /dev/vg00/lvol1
# lvolboot -r /dev/vg00/lvol3
# lvolboot -s /dev/vg00/lvol2
# lvolboot -d /dev/vg00/lvol2
# lvolboot -R
```

11. Setup the `AUTO` file in the EFI partition to ensure that if the primary boot disk fails that the system will be able to boot from the alternate boot disk without quorum:

```
# print 'boot vmunix -lq' > /tmp/AUTO
# efi_cp -d /dev/dsk/<second disk>s1 /tmp/AUTO EFI/HPUX/AUTO
# rm /tmp/AUTO
```

12. Set the path to the alternate boot device to be the disk that has just been created, enter:

```
# setboot -a <hardware path of second disk>
```

13. Ignite-UX does not preserve the Offline Diagnostics in the HPSP partition on the boot disk that it creates. That means that you must reinstall the contents of the HPSP partition from the *HP Itanium Processor Family Diagnostics and Utilities CD* that is appropriate for your system. When you choose to install the diagnostics to a HPSP partition, it will only be installed to one partition. After installation you must manually determine the partition on which the diagnostics have been installed using the following command:

```
# efi_ls -d /dev/rdisk/<device>s3 /
```

The diagnostics are present on one partition only. You can then copy the diagnostics to another HPSP partition using the following command:

```
# dd if=/dev/rdisk/<from device>s3 of=/dev/rdisk/<to device>s3 bs3=1024k
```

When you update the offline diagnostics, you must remember to update the other HPSP partition so that the updated diagnostics are available from both partitions.

14. To facilitate a recovery until `make_tape_recovery` and `make_net_recovery` are able to preserve the offline diagnostics on HP Integrity systems, you might choose to create a copy of the HPSP partition. This makes it easier to place the HPSP partition back on the same partition after a recovery. To create a copy of the diagnostics you might choose to use the following example:

```
# mkdir /var/HPSP
# dd if=/dev/dsk/<boot disk>s3 bs=1024k | gzip > \
> /var/HPSP/<boot disk>.save.gz
# dd if=/dev/dsk/<second disk>s3 bs=1024k | gzip > \
> /var/HPSP/<second disk>.save.gz
```

Run `make_tape_recovery` or `make_net_recovery` ensuring that `/var/HPSP` (or where ever you have placed the copies of the HPSP partitions) is included into the recovery archives. After recovery you can then restore the diagnostics as in the following example:

```
# gzcat < /var/HPSP/<boot disk>.save.gz | \
> dd of=/dev/dsk/<boot disk>s3 bs=1024k
# gzcat < /var/HPSP/<second disk>.save.gz | \
> dd of=/dev/dsk/<second disk>s3 bs=1024k
```

Note:

If you include configuration into Ignite-UX configuration files to perform these actions automatically after a recovery, you should ensure that this recovery archive will not be used to clone a system on which these offline diagnostics will not work.

After creating an alternate boot disk using LVM, HP recommends that you test that it can boot and operates as expected.

³ The size of the HPSP partition should be a multiple of the block size (bs) used with this command.

How to Recover Mirrored Disk Systems with Ignite-UX using `make_tape_recovery`

The `make_tape_recovery` command creates a recovery tape for a system with mirrored disks but it will not preserve the mirror configurations. After booting from the system recovery tape, disk mirrors are lost and must be reconfigured using LVM commands.

There are two methods to do this. First, you can wait until the system is recovered and then execute the necessary LVM commands manually or use SAM (note SAM cannot remirror a boot disk – see the instructions given previously for how to manually reestablish mirroring on a boot device on HP 9000 and HP Integrity systems).

Second, it is possible to make some manual changes to an Ignite-UX configuration file referenced by `make_tape_recovery` such that the LVM commands are executed as part of the recovery process itself.

Important:

If you choose to pursue the second option a recovery tape created in this way should NOT be used to clone systems unless they contain very similar hardware.

To integrate the configuration of disk mirrors into the recovery process we will be using a feature of the `make_tape_recovery` command. The `make_tape_recovery` command automatically includes the `/var/opt/ignite/recovery/config.local` file, if it exists, into any `make_tape_recovery` configuration.

Caution:

If you perform a preview when using these instructions (using the `-p` option) there is a defect in `make_tape_recovery` that prevents the `config.local` file from being included into the configuration when you resume the `make_tape_recovery` command (using the `-r` option). If you plan on performing a preview and resume operation you should upgrade to Ignite-UX version C.6.2 or later where this defect has been addressed.

After creating or modifying `/var/opt/ignite/recovery/config.local`, it's a good practice to run `"instl_adm -T -f /var/opt/ignite/recovery/config.local"`, to ensure that the syntax is correct. If the syntax is incorrect this will prevent the creation of the recovery tape.

In the section "An Example `config.local` File

" is an example `config.local` files for both HP Integrity systems and HP 9000 systems.

How to Recover Mirrored Disk Systems with Ignite-UX via `make_net_recovery`

The `make_net_recovery` command will create a network recovery archive for a system with mirrored disks, but it will not preserve the mirror configurations.

After recovering from the network recovery archive, disk mirrors are lost and must be reconfigured using LVM commands. There are two methods to reconfigure your mirrored disks:

- You can wait until the system is recovered, and then execute the necessary LVM commands manually or use SAM. If you choose to use SAM, be aware that it cannot remirror a boot disk. You must manually reestablish mirroring on a boot device as described in "How to mirror the root disk

on a HP 9000 system” on page 3 or “How to mirror the root disk on an HP Integrity Server” on page 4.

- You can manually create an Ignite-UX configuration file such that the LVM commands are executed as part of the recovery process itself.

In either case, the LVM commands are the same.

The following sections will show example configurations that can be used on HP 9000 and HP Integrity systems. To use these examples you must create a new Ignite-UX configuration file named:

```
/var/opt/ignite/clients/0x{LLA}/recovery/config.local
```

This `config.local` file is automatically included into your recovery configuration for this client each time you run the `make_net_recovery` command.

Note:

The `make_net_recovery` command is automatically executed when you use the Ignite-UX GUI for network recovery.

If you already have recovery configurations for this client and would like them to use the `config.local` file, edit the `/var/opt/ignite/clients/0x{LLA}/CINDEX` file to include a reference to “`recovery/config.local`” in all of the configuration clauses.

The commands required to configure mirrored disks can be included as a `post_config_cmd` script, which executes automatically after the system has been restored.

After creating or modifying the `config.local` file, it is recommended that you run `instl_adm -T -f config.local`, to ensure that the syntax is correct and that the file is error free.

The `config.local` file used for `make_net_recovery` is identical to is used for `make_tape_recovery`. The only difference is the location of the `config.local` file for each command.

An Example config.local File

The following example `config.local` configuration file is applicable for both HP 9000 and HP Integrity servers, and can be used with either `make_tape_recovery` or `make_net_recovery`.

On HP Integrity servers, it is assumed that the name of the root volume group is `vg00`.

On all systems, it is assumed that the root volume group contains **only** two physical volumes.

You should be aware that Ignite-UX will automatically include the alternate boot disk into the root volume group with no partitioning. This means that it is likely that the number of physical extents in the root volume group after a recovery will be different than its original state prior to the recovery. This is the expected behavior of recovery on HP Integrity systems.

Important:

In spite of efforts to minimize split lines in the following example `config.local` file, it is possible that some may exist. If you choose to use this exact text, HP recommends that you review your file for completeness and ensure that no lines have been split inappropriately.

The config.local example follows:

```
#Remirror this systems boot disk
enum _MY_REMIRROR
_MY_REMIRROR help_text "Attempt to remirror disks?"
_MY_REMIRROR = { "TRUE", "FALSE" }
init _MY_REMIRROR = "TRUE"
(( _HP_CLONING == "FALSE" ) & ( _MY_REMIRROR == "TRUE" )) {

post_config_cmd +=

# Uncomment the next line to enable debug.
# set -x

# If a command gets an error exit out of the script.

trap \"exit\" ERR

# What Kind of processor is this PA-RISC or Itanium?

cpu=$(/usr/bin/getconf CPU_VERSION)
if [ $cpu = 532 ]
then
    CPU=PA
else
    CPU=IA
fi

# Recreate a mirror of the boot disk. This assumes that the root
# volume group was mirrored when this configuration was added
# if this is an initial mirroring the disk does not have to be
# reduced from the volume group.

# CHECK how many Disks in root vg

NO_OF_00_DSKS=$(vgdisplay -v vg00 |grep \"PV Name\" |wc -l)
if [ $NO_OF_00_DSKS != 2 ]
then
    echo \" This script is designed to work with ignite.\"
    echo \" It remirrors a PV in vg00 to a second PV.\"
    echo \" It presumes the disks were previously mirrored.\"
    echo \" In so doing only works with vg00s that have 2 disks.\"
    echo \" This script will not run because this vg00 has
$NO_OF_00_DSKS disks.\"
    exit
fi

echo \"\n\n This vg00 has $NO_OF_00_DSKS disks. Continuing on.\n\n\"

# I want 1 disk with 0 lvols and another disk with more than 2.

ROOT_DSKS=$(vgdisplay -v vg00 |grep \"PV Name\" | awk ' { print $3 }
' )
echo $ROOT_DSKS|read DISK1 DISK2
NUM_OF_LVOLS1=$(/usr/sbin/pvdisplay $DISK1 |/usr/bin/grep LV|awk ' {
print $3 } ' )
NUM_OF_LVOLS2=$(/usr/sbin/pvdisplay $DISK2 |/usr/bin/grep LV|awk ' {
print $3 } ' )

# Lets Check Which Disk has lvols on it and use that as the primary.

if [ $NUM_OF_LVOLS2 -gt 2 -a $NUM_OF_LVOLS1 = 0 ]
```

```

then
    echo $ROOT_DSKS|read DISK2 DISK1
    /usr/sbin/pvdisplay $DISK1 |/usr/bin/grep LV\
    |/usr/bin/read CUR LV NUM_OF_LVOLS1
    /usr/sbin/pvdisplay $DISK2 |/usr/bin/grep LV\
    |/usr/bin/read CUR LV NUM_OF_LVOLS2
fi

echo \" Since $DISK1 has $NUM_OF_LVOLS1 lvols\"
echo \" and $DISK2 has $NUM_OF_LVOLS2 lvols,\"
echo \" Mirror-UX will be used to make a mirror \"
echo \" of $DISK1 on $DISK2.\"

# Make sure the disk is removed from the group by using vgreduce.

echo \" \n\nReducing $DISK2 from vg00.\"

vgreduce /dev/vg00 $DISK2

# Once reduced we need to section the disk with the idisk command
# the sizes of the EFI and HPSP sections need to match the mirror.
# This assumes that there is a HP Service Partition (HPSP). If the
# size of any partition is different the sizes below need to be
# changed.

PARTS1=${DISK1#/dev/dsk/}
PARTS2=${DISK2#/dev/dsk/}
PART1=$(echo $PARTS1|cut -c 0-6)
PART2=$(echo $PARTS2|cut -c 0-6)

# So we know that PART1 & 2 have no Disk sectioning in the
# device file name. We'll presume PARISC and set PARTS2
# with no disk section device file for NON IA lvm commands.

PARTS2=${PART2}

if [ $CPU = IA ]
then

# If we get here we know we're IA so ... we'll set PARTS2
# disk section device file for IA lvm commands.

PARTS2=${PART2}s2

# Let figure out the partition sizes on DISK and use those
# on DISK2.

LBA=$(idisk -p /dev/rdisk/$PART1|grep -e Starting -e Ending |\
awk ' { print $4 } ' )
echo $LBA |read EFIS EFIE HPUXS HPUXE HPSPS HPSPE
efisize=\"ibase=16;$(echo ${EFIE#0[xX]}\
|tr \\\"\\\"[abcdef]\\\\\" \\\"\\\"[ABCDEF]\\\\\"\\\"))-\\
$(echo ${EFIS#0[xX]}|tr \\\"\\\"[abcdef]\\\\\" \\\"\\\"[ABCDEF]\\\\\"\\\"))+1\"
EFISize=$(echo $efisize |bc)
EFISIZE=$(echo $EFISize\\2\\1024|bc)

hpspsize=\"ibase=16;$(echo ${HPSPE#0[xX]}\
|tr \\\"\\\"[abcdef]\\\\\" \\\"\\\"[ABCDEF]\\\\\"\\\"))-\\
$(echo ${HPSPS#0[xX]}|tr \\\"\\\"[abcdef]\\\\\" \\\"\\\"[ABCDEF]\\\\\"\\\"))+1\"
HPSPsize=$(echo $hpspsize |bc)
HPSPSIZE=$(echo $HPSPsize\\2\\1024|bc)

echo \"\n\n It has been determined that partition 1 ond 3 \"

```

```

echo \" on $PART1 are $EFISIZE MB and $HPSPSIZE MB\"
echo \" in size. These sizes will be used to partition $PART2.\"
echo \" The remainder of the disk will be assigned to partition 2.\"

print \"3\nEFI ${EFISIZE}MB\nHPUX 100%\nHPSP ${HPSPSIZE}MB\" >\
/tmp/partitionfile
echo \"\n\n Partition file has been create .... Displaying \n\n\"
cat /tmp/partitionfile

echo \"\n\n Partitioning /dev/rdisk/$PART2 \"
idisk -qwf /tmp/partitionfile /dev/rdisk/$PART2
rm /tmp/partitionfile

# After partitioning the disk it needs to have its new device
# files created for it, we cant just use insf because Ignite-UX
# has renamed the device as it probably had another name. We
# need to get the major/minor numbers from the device file
# and manually create the section devices.

echo \"\n\n Determining Major and minor Numbers\"
echo $(/usr/bin/ll /dev/dsk/${PART2} |tail -1 | \
awk ' { print $5\" \"$6 } ' ) | read BMAJNUM MINNUM
CMAJNUM=$(/usr/bin/ll /dev/rdisk/${PART2} |tail -1 \
| awk ' { print $5 } ' )
CUTMINNUM=$(print $MINNUM | cut -c 1-7 )

echo \"\n\n Removing OLD devices files.\"
ls /dev/*dsk/${PART2}s?|xargs -i -t rmfsf {} 2>&1 >/dev/null

echo \"\n\n Creating new device files.\"
mknod /dev/rdisk/${PART2}s1 c $CMAJNUM ${CUTMINNUM}1
mknod /dev/rdisk/${PART2}s2 c $CMAJNUM ${CUTMINNUM}2
mknod /dev/rdisk/${PART2}s3 c $CMAJNUM ${CUTMINNUM}3
mknod /dev/dsk/${PART2}s1 b $BMAJNUM ${CUTMINNUM}1
mknod /dev/dsk/${PART2}s2 b $BMAJNUM ${CUTMINNUM}2
mknod /dev/dsk/${PART2}s3 b $BMAJNUM ${CUTMINNUM}3

# The service partition on the disk just partitioned will not contain
# anything you may choose to copy it from the other disk. Note the
# use of the section number.

echo \"\n\n Copying service partition from ${PART1}s3 to
${PART2}s3\"
dd if=/dev/rdisk/${PART1}s3 of=/dev/rdisk/${PART2}s3 bs=1024k

fi

# Make the disk contain a boot area.

echo \"\n\n pvcreating ${PARTS2}\"
pvcreate -f -B /dev/rdisk/${PARTS2}

# Add the mirrored disk back to the group.

echo \"\n\n Adding (vgextend) mirrored disk ${PARTS2}\"
echo \" back into vg00.\"

vgextend /dev/vg00 /dev/dsk/${PARTS2}

# Copy the boot area to the disk. Note that mkboot is done to
# the whole disk for bot IA and PA but the command is not
# the same so .....

```

```

if [ $CPU = IA ]
then

    echo "\n\n Copy the boot area on ${PART2} with mkboot.\"
    mkboot -e -l /dev/rdisk/${PART2}

# Turn off the quorum requirement

    echo "\n\n adding hpux -lq to auto_file_string\"
    echo \"boot vmunix -lq\" > /tmp/AUTO.lq
    efi_cp -d /dev/rdisk/${PART2}s1 /tmp/AUTO.lq /EFI/HPUX/AUTO
    echo "\n\n repeating for the primary boot disk.\"
    efi_cp -d /dev/rdisk/${PART1}s1 /tmp/AUTO.lq /EFI/HPUX/AUTO

else
# Lets do the PA Dance

    mkboot /dev/rdisk/${PARTS2}
    mkboot -a \"hpux -lq\" /dev/rdisk/${PART1}
    mkboot -a \"hpux -lq\" /dev/rdisk/${PARTS2}
fi

# Allocate the mirrors. Mirrors must be allocated for all
# logical volumes that were previously mirrored. This
# example illustrates primary swap and root. You should
# add others as needed. If /stand is in a separate volume, you
# should do an lvextend for it first.

vgdisplay -v vg00|grep -e \"LV Name\" | awk ' { print $3 } ' \
| xargs -i -t lvextend -m 1 {} /dev/dsk/${PARTS2}

# Update the BDRA and the LABEL file.

lvlnboot -R
"
}

```

Important:

While HP has tested the above `config.local` file on some systems and configurations, it has not been tested in every possible situation and configuration. HP recommends that you test it and ensure that it works correctly in your environment before relying on it in a production situation.

Glossary

EFI	Extensible Firmware Interface is the firmware interface for HP Integrity systems. It is the name of the first partition on a HP-UX boot disk.
HPSP	HP Service Partition, which contains the Offline Diagnostics on HP Integrity systems.
LIF	Logical Interchange Format.

For More Information

The following relevant documents are available online at the HP Technical Documentation Web site at <http://www.docs.hp.com/>:

Ignite-UX Administration Guide

Successful System Cloning Using Ignite-UX

Managing HP-UX Software With SD-UX

Software Distributor Administration Guide for HP-UX 11i

HP-UX 11.0 Installation and Update Guide

Release Notes for HP-UX 11.0

HP-UX 11i v[1 | 1.6 | 2] Installation and Update Guide

HP-UX 11i v[1 | 1.6 | 2] Release Notes

Managing Systems and Workgroups: A Guide for HP-UX System Administrators

Additionally, there are a number of white papers pertinent to Ignite-UX that are located at the HP Technical Documentation Web site.

The following Web sites contain useful information that relevant to the ideas presented in this white paper:

[HP Partitioning Continuum for HP-UX 11i](#)

Product information regarding Ignite-UX for HP-UX is available at the HP Software Depot at

<http://www.docs.hp.com/en/IUX/>

© Copyright 2006 Hewlett-Packard Development Company, L.P.

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

Intel Itanium® Logo, Intel, Intel Inside and Itanium are trademarks or registered trademarks of Intel Corporation in the US and other countries and are used under license.

Intel® Itanium® Processor Family is a trademark of Intel Corporation in the US and other countries and is used under license

MPN 5991-5845, 6/2006

