



How to use the HP-UX Portable Image Tool for Integrity fluid cross-technology moves

HP-UX Portable Image Tool

Table of contents

Introduction	2
Logical Server use cases	2
Key challenges	2
High-level architecture	3
HP-UX IO Infrastructure (IOI)	3
Ethernet drivers	4
Data Link Provider Interface (DLPI)	4
Flow of operation	4
An example use case—Preserving VC-assigned MAC addresses in HP-UX by enabling Portable Image (BL8x0c i2 Field Upgrade Service)	5
Reference	7
Acronyms and glossary	7
Appendix 1—/opt/network/bin/hpuxpitool	9

Introduction

Most modern data centers are composed of heterogeneous server environments with servers from multiple vendors of different types, generations, and product families. Fluid cross-technology logical server moves allow you to choose server configurations that make the most sense for your environment without having to consider the mechanics of moving images and the underlying technologies.

A logical server is a set of configuration information that you create, activate, and move across physical and virtual machines. It describes compute resources (number of CPU cores, memory etc.) and connections to storage fabric and networks.

Logical servers can be moved between physical servers with Virtual Connect and ESX/Integrity VM virtual machines, and between physical servers with Virtual Connect with unlike configurations.

In this document, we describe the `hpuxpitol`, related modules and how to use it to make networking portable by retaining instance numbers (PPAs) between source and target. During a P2V or a V2P move from a source to a target, the instance numbers change for networking interfaces. These instance numbers are maintained by the General Input-Output (GIO) subsystem on a per-class and instance number—hardware path mappings are stored in I/O configuration files. All the existing HP-UX configuration files that contain persistent configuration rely on the original networking interface. The idea is to reassign the source instance numbers for all networking devices on the target using Kernel Registry Services (KRS) based on a MAC address match to avoid changing any configuration files.

Logical Server use cases

Some of the use cases for logical servers include:

- Reduce time to deploy new servers
 - Logical Servers can be standard blueprints for new servers
 - Reduces time and possible configuration errors
- Conveniently expand and contract application server pools
 - Logical Servers can be activated and deactivated on demand
- Reuse resources for another purpose
 - Logical Server on a given resource can be deactivated and the same resource can be activated with another Logical Server
- Move servers to different locations for maintenance purposes
 - Minimal or no downtime
 - Does not require network administrator or SAN administrator involvement

Key challenges

The following are the key challenges for Physical to Virtual (P2V) and Virtual to Physical (V2P) moves between VC Integrity blades and Integrity VM hosts:

- Making networking portable by retaining instance numbers between source and target.
- Migrating the EFI State Information (ESI) between source and target
- Support for a variety of configurations including Serviceguard and VLANs
- Pre-move validation between source and target
- N-Port ID Virtualization (NPIV) support

The HP-UX Portable Image (PI) tool resolves first challenge to make networking portable by retaining instance numbers between source and target.

High-level architecture

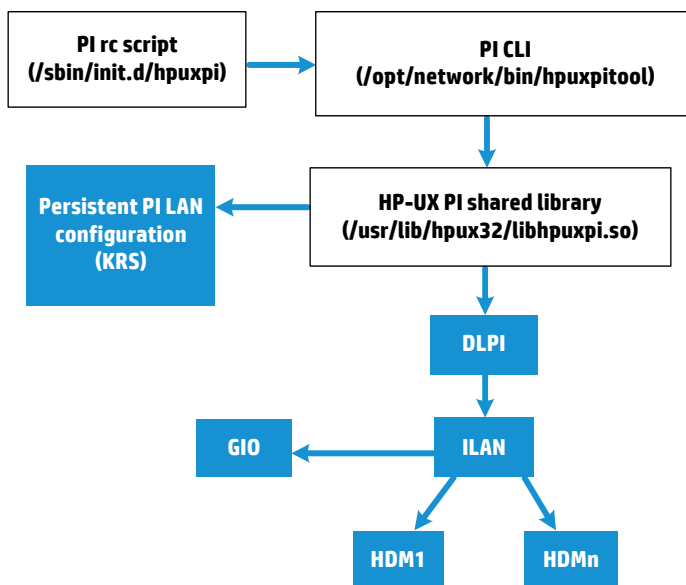
The HP-UX PI product consists of the following components:

- HP-UX PI shared library (/usr/lib/hpux32/libhpuxpi.so)—provides private API to enable/disable PI functionality, add/remove/search MAC addresses in KRS, save and update PI configuration in KRS, and fix driver configuration file. PI library interacts with DLPI to determine factory MAC address of all physical NIC and have support for message catalogs and logging messages to NetTL.
- HP-UX PI tool (/opt/network/bin/hpuxpitool)—invokes the HP-UX PI shared library to perform actions on the PI configuration database.
- Startup script (/sbin/rc2.d/S302hpuxpi -> /sbin/init.d/hpuxpi start)—invokes the HP-UX PI tool to update the PI configuration upon every boot in KRS. This will fix-up drivers configuration files.

The start-up script is invoked before any LAN interfaces are configured to ensure that driver configuration files are appropriately fixed up before LAN configuration. A dependency has been added to all LAN drivers on HP-UX PI in the parallel framework database to ensure that this ordering is strictly enforced. When the startup script is invoked during system boot, it invokes the hpuxpitool -u to update the HP-UX PI configuration. It logs any error to /etc/rc.log.

- Shutdown script (/sbin/rc1.d/K692hpuxpi -> /sbin/init.d/hpuxpi stop)—invokes the “hpuxpitool -s” to save (but not update) the HP-UX PI configuration upon every shutdown in KRS.

To know more about hpuxpitool usages, see **Appendix 1—/opt/network/bin/hpuxpitool**



HP-UX IO Infrastructure (IOI)

GIO is responsible for associating instance numbers based on MAC addresses for “lan” class devices instead of using the H/W path information in ioconfig. As part of the move when the source is shut down, the HP-UX PI tool saves the H/W path, MAC address, and instance number association for all “lan” class devices in the KRS. When the system is booted up, GIO reserves the instance numbers specified in the PI KRS. When devices are initially scanned, GIO assigns instance numbers outside of the reserved range. When drivers determine their MAC address and call into GIO to reassign the instance numbers, it returns the reserved instance number if the MAC address matches with any entry in the PI KRS. Otherwise, the initial assignment during scan is preserved.

GIO also ensures that a different mechanism to assign instance numbers do not conflict with each other (e.g., ioinit and HP-UX PI configuration).

Ethernet drivers

Hardware dependent modules (HDM) determine the MAC address of the networking device and sets the Infrastructure LAN (ILAN) interface data structure. ILAN then invokes the new GIO interface with the MAC address of the device to reassign the PI configuration based on the MAC address. If an instance number has been reserved for this MAC address, GIO reassigns the PI configuration instance number to this MAC address. ILAN then invokes HDM property interface to reset the PPA in the HDM data structures, invokes DLPI property interface to set the new PPA for the DLPI instance and then communicates the MAC address for the DLPI instance.

Data Link Provider Interface (DLPI)

When the driver to GIO call to reassign the instance number succeeds, DLPI is invoked with `dlpi_prop` call to update the instance number of the DLPI instance prior to setting the MAC address in the DLPI instance. This will be reflected by tools such as `lanscan`.

Flow of operation

A new KC tunable called `portable_image` introduced that defaults to 0 (disable). The user will have to manually set this KC tunable as part of the initial activation of LS on a V or P. The HP-UX PI startup and shutdown scripts do not do anything (restore or save) if this KC tunable is not enabled. Since there is no configuration saved, IOI will default to its hardware path based instance number assignment.

After enabling HP-UX PI by executing the following command: `kctune gio_portable_image=1` or `/opt/network/bin/hpuxpitool -e`, whenever the source is shutdown, HP-UX PI shutdown script is executed. This saves the <H/W path, class of device, instance number, driver name, MAC address1, MAC address2 ...> information in KRS for every networking device (as determined by `nwmgr` output). The HP-UX PI script also stores the driver type associated with the networking interface on the source.

When the target is booted, IOI initially reads KRS and reserves the instance numbers specified in the file for the specified class of device and frees up `ioconfig` records that have these instance numbers. If there is no KRS entry, then IOI will default the instance number assignment to the traditional hardware path model. When the devices are scanned and claimed (during attach), it assigns initial instance numbers based on the new hardware paths and `ioconfig`.

Later the driver `init()` entry point is invoked. Different Ethernet drivers (HDM) determine the MAC address at different points in the `init` sequence. The common Ethernet driver infrastructure ILAN gets the MAC address from the HDM and communicates this to DLPI (using `dlpi_prop()` interface) as part of an internal reset during `init`. Prior to communicating the MAC address to DLPI, ILAN invokes a newly created GIO routine passing in the I/O node (from `isc`) and the MAC address. IOI determines the hardware path from the I/O node, compares the ILAN provided MAC address with that in the saved `/stand/<class>_ioconfig` information on the source, updates the `ioconfig` information with the new hardware path, and returns the matching instance number, if any. If the instance number changes, ILAN updates its instance number and communicates the instance number to DLPI using a newly defined property in `dlpi_prop()` interface. Thus, devices that have `DRV_INSTANCE_REPLACEABLE` flag set and have matching MAC addresses to the source configuration will have their instance numbers reassigned by IOI. Devices that do not have a matching MAC address will not have their instance numbers reassigned, but will have instance numbers that do not conflict with the source configuration.

The only item to fix-up on the target is the driver configuration file. The HP-UX PI startup script determines the driver type associated with the networking interface on the source and the target and updates the appropriate driver configuration file on the target with any non-driver specific configuration (e.g., MTU, station address, etc.). Driver specific information (e.g., CKO, TSO, etc.) are ignored.

Example scenario:

Source has one `igelan` adapter at `lan0` with `ioconfig` record <HWP-A, lan, 0>. <HWP-A, lan, 0, igelan, MAC-A> is saved by HP-UX PI shutdown script on the source. When the target is booted up, instance number 0 is reserved by IOI. When two `lan` devices are scanned, they get instance numbers 1 and 2. `ioconfig` records now have three entries <HWP-A, lan, 0>, <HWP-1, lan, 1>, <HWP-2, lan, 2>. `lan1` driver determines its MAC address MAC-B and calls `io_node_reassign_instance()` and retains its instance number. `lan2` determines its MAC address MAC-A and calls `io_node_reassign_instance()` and gets reassigned instance number 0. `ioconfig` records now has two entries <HWP-2, lan, 0> and <HWP-1, lan, 1>.

An example use case—Preserving VC-assigned MAC addresses in HP-UX by enabling Portable Image (BL8x0c i2 Field Upgrade Service)

The HP-UX Portable Image product is installed and enabled on the blade server before the upgrade of a BL860c i2 to BL870c i2 Blade Link Domain configuration. This is also applicable in using blade additions and combinations to upgrade to a BL890c i2 server blade. After the upgrade, PI will automatically modify the HP-UX network instance numbers of the new hardware environment so that they match the pre-upgrade environment. This takes advantage of the VCM interconnect stacking links to allow all of the Monarch Server network connections to function correctly with no corrections needed in the original VC server profile.

Portable Image can be used on VC configured systems where the VC domain has selected Virtual Connect assigned MAC addresses instead of the static, factory-default MAC addresses. If you choose to use PI to assist the upgrade, there will be a pre-upgrade step to follow on the designated Monarch Server and then another step to follow after the upgrade.

Pre-upgrade procedure for PI

These instructions are to be executed on the designated Monarch Server with the OS booted.

1. You need to obtain the PI package if it's not already installed.
 - a. First, check for it being already installed by executing the following command:

```
swlist|grep HPPortableImage
```
 - b. If nothing is returned, then the package must be installed now. Go to **software.hp.com** and search for "HP Portable Image". Follow the link for HP-UX Portable Image in the product list and follow the instructions there for installing the depot.
2. If your HP-UX 11i v3 release is September 2010 or earlier, then you must install the GIO cumulative patch before proceeding.
 - a. Log into the HP IT Resource Center Web page at **itrc.hp.com** and follow the "Patch database" link under "Downloads". The site requires a quick registration.
 - b. Search for "PHKL_41525" and download the patch.
 - c. Follow the patch link and execute the installation instructions found there.

3. Enable PI by executing the following command:

```
kctune gio_portable_image=1
```

You should see the following:

Tunable	Value	Expression	Changes
gio_portable_image (before)	0	Default	Immed
(now)	1	1	

Or

```
/opt/network/bin/hpuxpitol -e
```

4. The PI state must be saved to KRS before starting the upgrade. This is done by explicitly executing the PI save command:

```
/opt/network/bin/hpuxpitol -s
```
5. Dump the saved PI state to a text file for possible use later. Obtain the dump by executing the following command:

```
/opt/network/bin/hpuxpitol -g
```

This will list out each network instance along with its pre-upgrade MAC address.
6. Shut down the OS using a "shutdown" command.

Post-upgrade procedure for PI

1. Dump the new PI state to a file for possible use later. Obtain the dump by executing the following command:

```
/opt/network/bin/hpuxpitol -g
```

This will list out each network instance along with its post-upgrade MAC address.

2. Disable PI by executing the following command:

```
kctune gio_portable_image=0
```

You should see the following:

Tunable		Value	Expression	Changes
gio_portable_image	(before)	1	1	Immed
	(now)	0	0	

Or

```
/opt/network/bin/hpuxpitol -d
```

3. The networking should be checked to make sure that the original system network instances function correctly. If the HP-UX boot reports a LAN interface configuration failure, then check the `/etc/rc.log` file for the failed instance number, which may indicate something like the following example:

```
ERROR: lan18 interface: ifconfig: no such interface
```

```
"/sbin/rc2.d/S340net start" FAILED
```

Get the MAC address for the failed instance number from the pre-upgrade PI dump and look for that MAC address in the post-upgrade PI state dump. From that entry, note the new instance number and edit the HP-UX networking configuration for the failed instance number so that it uses the new instance number.

Reference

HP Insight Management

<http://h18004.www1.hp.com/products/solutions/insightdynamics/info-library.html>

Acronyms and glossary

Term	Definition
HPVM	Hewlett Packard Virtual Machine, internal code name for Integrity VM
Hypervisor	A hypervisor is a module which allows multiple OS instances to run on a single hardware platform. In Integrity VM, the PMAN is the hypervisor.
I/O (Accelerated, Direct, Virtual)	Accelerated, Direct, and Virtual I/O refer to the three different I/O models for Integrity VM. Virtual I/O refers to standard drivers used in the guest, which rely on Integrity VM to emulate registers to allow the Virtual I/O to execute without modification in the Guest. Direct I/O refers to specially modified drivers, which run in the Guest and have direct access to the physical hardware below. This results in higher performance than VIO but does not support the sharing of the hardware between multiple guests. Accelerated I/O refers to specially modified drivers, which run in the Guest and Host and allow the sharing of the hardware between guests but perform better than VIO.
VIO, Virtual IO	VIO or Virtual IO refers to the unchanged driver stack in an Integrity VM, which relies on the host to provide register emulation to run unmodified.
Virtual NIC	A virtual NIC is the NIC which is seen by a guest in the Integrity VM environment. The virtual NIC in the guest looks like a NIC in the normal operating environment for the various applications running on the guest, but in actuality may be one of many interfaces from various guests sharing a single physical NIC in the host environment.
Virtual switch, vswitch	The vswitch, or virtual switch, is the Integrity VM module which emulates an Ethernet switch. This function is necessary in order for the Integrity VM guests to share via VIO a common physical NIC on the Host. The switch performs similar functions as a real switch with the main function being routing packets between the Guests, Host, and the wire (through the physical NIC).
VMSAN	Virtual Fibre Channel HBA on HP-UX Integrity VM 11i v3 1003 guest that provides a FC device for SAN management provisioning at the guest.

Acronym	Definition
CMS	Central Management Station
CSO	Checksum Offload
DLPI	Data Link Provider Interface
EFI	Extensible Firmware Interface
ESI	EFI State Information—special format on legacy zx2 blades that contains EFI variable store and some non-volatile information from the blade
GIO	General Input Output. Refers to the generic module in the I/O subsystem. The kernel interfaces with the I/O subsystem through this module.
HBA	Host Bus Adapter
HDM	Hardware Driver Module
ILAN	Infrastructure LAN
IOI	HP-UX IO infrastructure. Its components such as: GIO (General IO), WSIO (Work Station IO), OL infrastructure, IO Commands, LibIO and Driver products such as: Async disk, Ramdisk, Partition Filter, Tape, and Changer Drivers.
KRS	Kernel Registry Services is a kernel-specific mechanism that facilitates the maintenance of structured data. Subsystems within the kernel can use KRS to maintain static or dynamic data. This data can be either volatile or persistent across system reboot.
LAN	Local Area Network
LS	Logical Server
LSM	Logical Server Manager
MTU	Maximum Transmission Unit
NIC	Network Interface Card
NPIV	N-Port ID Virtualization
NVRAM	Non Volatile Random Access Memory
P2V	Physical to Virtual Machine
PPA	Physical Point of Attachment. It is an instance number of the NIC that is involved in sending and receiving of data. When lanscan is performed, under the “Net-Interface” column of the results, each instance of the device is listed with its name immediately followed (no spaces in between) by its PPA. The number listed for the PPA is the instance number of the device.
TSO	TCP Segmentation Offload
V2P	Virtual Machine to Physical
VLAN	Virtual Local Area Network
VMM	Virtual Machine Manager
VMSAN	Virtual Machine Storage Area Network
VNIC	Virtual NIC

Appendix 1—/opt/network/bin/hpuxpitol

`getcfg (or) -g` : Retrieve PI configuration from persistent KRS database

`savecfg (or) -s`: Save PI configuration (invoked by shutdown script) to persistent KRS database

- Loads the HP-UX PI shared library and invokes the HP-UX PI API to save HP-UX PI configuration based on the list of networking interfaces currently configured on the system

`updatecfg (or) -u`: Update PI configuration and fix driver configuration file (invoked by startup script) to persistent KRS database

- Loads the HP-UX PI shared library and invokes the HP-UX PI API to update HP-UX PI configuration based on the list of networking interfaces currently configured on the system and driver configuration files

`addcfg (or) -a`: Add equivalent MAC address to PI configuration in persistent KRS database

`hpuxpitol -a <MAC1> <MAC2>`

- Loads the HP-UX PI shared library and invokes the HP-UX PI API to add HP-UX PI configuration

`rmcfg (or) -r`: Remove one or all MAC address from PI configuration in persistent KRS database

`hpuxpitol -r [<MAC>]`

- Loads the HP-UX PI shared library and invokes the HP-UX PI API to remove HP-UX PI configuration

`enable (or) -e`: Enable Portable Image functionality

- Loads the HP-UX PI shared library and invokes the HP-UX PI API to enable HP-UX PI functionality

`disable (or) -d`: Disable Portable Image functionality

- Loads the HP-UX PI shared library and invokes the HP-UX PI API to enable HP-UX PI functionality

Get connected

hp.com/go/getconnected

Current HP driver, support, and security alerts delivered directly to your desktop

