Driver for Linux Release Notes

Versions: Driver for Linux, FC/FCoE Version 10.0.803.2203 (RHEL 5.6-5.9, RHEL 6.1-6.4 / SLES 11 SP1, SP2)
Driver for Linux, NIC Version 10.0.803.12
Driver for Linux, iSCSI Version 10.0.719.1006 (RHEL 5)
Driver for Linux, iSCSI Version 10.0.803.7 (RHEL 6/SLES 11)

Date: February 2014

Purpose and Contact Information

These release notes describe the resolved known issues and current known issues associated with these Emulex® driver for Linux releases.

For the latest product documentation, go to www.Emulex.com. If you have questions or require additional information, contact an authorized Emulex technical support representative at tech.support@emulex.com, 800-854-7112 (US/Canada toll free), +1 714-885-3402 (US/International), or +44 1189-772929 (Europe, Middle East, and Africa).

Recommended Firmware Versions

- OCe1100x: 10.0.803.13
- OCe14000: 10.0.803.2202
- LPe1600x: 10.0.803.8
- LPe1200x: Firmware 2.01a11, Boot Code 5.13a0

Resolved Issues

Note: SLES 10 drivers have been frozen and no future support is planned.

FC/FCoE Version 10.0.803.2203 Resolved Issues

Added support for the OCe14000 family of adapters.

NIC Version 10.0.803.12 Resolved Issues

Added support for the OCe14000 family of adapters.

iSCSI Versions 10.0.719.1006 and 10.0.803.7 Resolved Issues

1. **Added support for the OCe14000 family of adapters.**
2. **Previously, Emulex provided a proprietary iSCSI driver as the out-of-box driver for all Linux operating systems. The Open-iSCSI driver was available inbox only. For RHEL 6.4 and SLES 11 SP2 and later drivers, Emulex is transitioning the Open-iSCSI driver to be the out-of-box driver. For RHEL 5.x, the out-of-box driver will continue to be the proprietary driver.**

Known Issues

FC/FCoE Version 10.0.803.2203 Known Issues

1. **Link Aggregation Control Protocol (LACP) cannot be used on the same port as FCoE or iSCSI.**

   Workaround

   None.

2. **PCI Hot Plug may cause applications such as the OneCommand Manager application or third party applications that use the Emulex libraries (for example, HBAAPI), to misbehave or malfunction.**

   Workaround

   a) Stop all applications that are accessing LPFC's HBAAPI interface (OneCommand Manager application or third party applications) before performing PCI Hot Plug of an LPFC adapter.

   b) Use the following command to stop the OneCommand Manager application:

   ```
   #/usr/sbin/hbanyware/stop_ocmanager
   ```

   c) After performing PCI Hot Plug of the adapter, you can restart the applications.

3. **Deletion of Vports/PCI Hot Unplug**

   On occasion the kernel might report SCSI errors when deleting Vports via the sysfs interface or performing a PCI Hot Unplug of an Emulex adapter:

   ```
   kernel: Synchronizing SCSI cache for disk
   kernel: FAILED
   ```

   -or-

   ```
   SCSI error: return code = 0x00010000
   ```

   Workaround

   None. These messages do not indicate a functional failure and can be ignored.

4. **Deleting Vports while devices are in use.**

   Emulex provides management utilities that allow you to delete Vports. However, there is no mechanism for the LPFC driver to detect whether devices accessed through that Vport are in use. This means you can delete a Vport when devices accessible through the Vport are mounted or when I/O is outstanding to the device. When file systems are mounted on Vports and Vports are deleted, the file systems still appear to be mounted; however, they be inaccessible.

   Workaround

   Before deleting Vports you must prepare the system affected by the Vport deletion accordingly, by unmounting all the devices accessible through the Vports, and ensuring there is no outstanding I/O.

5. **Support of 4Gb/s adapters in Direct I/O virtualized environments.**

   Default driver configuration fails to initialize 4 Gb/s adapters in virtualized environments that use Direct I/O or SFPT. This may result in a system hang or an uninitialized LPFC adapter in Intel VT-d and AMD-V IOMMU systems.

   Workaround

   In these virtualized environments that use Direct I/O or SFPT and 4 Gb/s adapters, you must load the LPFC driver with the following driver parameters set:

   - lpfc_hostmem_hgp=1
   - lpfc_sli_mode=2

   For example:

   ```
   # modprobe lpfc lpfc_hostmem=1 lpfc_sli_mode=2
   ```

   Note: A consequence of this resolution is that virtual ports are no longer supported by the LPFC driver when the lpfc_sli_mode parameter is set to 2.

6. **Order of LPFC module in the initrd module list.**

   On SLES 11 systems, if another SCSI driver, such as aic79xx, is loaded right behind the LPFC driver through the initrd image, the loading of the SCSI driver right after the LPFC driver might interrupt the SCSI mid-layer discovery process on the LUNs connected to the LPFC driver. This causes the SCSI discovery reference count put on the LPFC driver to not release and the LPFC driver cannot unload.

   Workaround

   The Emulex driver kit installation script always puts the LPFC module at the end of the INITRD_MODULES list before building the initrd image. However, if this problem is observed make sure no SCSI drivers are added right after the LPFC module in the INITRD_MODULES list.

7. **The physical interface can improperly name eth0.123 when the /etc/sysconfig/network-scripts/ifcfg-eth0.123 file contains a HWADDR declaration.**

   When using VLAN on RHEL 6.x, the main interface is not created, and the VLAN interface does not actually work correctly as a VLAN.

   This is an issue with the udev script in RHEL 6.x. In RHEL 6.x, the udev scripts interpret the HWADDR field in an ifcfg-ethX.Y configuration file to mean that the configuration belongs to a real interface. It does not check for the existence of the VLAN field to exclude the field as a real interface. Therefore, you may see unpredictable behavior when including the HWADDR field in the ifcfg-ethX.Y file for a VLAN. The setting may, or may not, work. When this setting does not work, it is because the scripts have created or renamed the main interface as ethX.Y instead of just ethX. Later, when adding the VLAN interface, it fails because ethX does not exist.

   Workaround

   Remove the HWADDR line in ifcfg-ethX.Y files that refer to VLAN configurations. You must also clean up the /etc/udev/rules.d/70-persistent-net.rules file to remove all the incorrect references to ethX.Y.

   For more information, see https://bugzilla.redhat.com/show_bug.cgi?id=723936.

8. **When using bonding on top of VLANS on a RHEL-based system, networking appears to hang during system boot or when restarting the network.**

   On a RHEL-based system, having a bond on top of a VLAN while using ONBOOT=yes, can cause the system to hang during boot or when restarting the network. The system will appear to hang while trying to start one of the slave interfaces.

   For example, if you have a configuration similar to the following:

```
DEVICE=bond0
BROADCAST=10.255.255.255
GATEWAY=10.0.0.254
IPADDR=10.0.0.14
NETMASK=255.0.0.0
ONBOOT=yes

BONDING_OPTS="mode=0 miimon=100"
DEVICE=eth2.2
BOOTPROTO=none
ONBOOT=yes
MASTER=bond0
SLAVE=yes
VLAN=yes

DEVICE=eth3.2
BOOTPROTO=none
ONBOOT=yes
MASTER=bond0
SLAVE=yes
VLAN=yes
```

   This system hangs because the RHEL networking scripts try to bring up the eth2.2 and eth3.2 interfaces twice. It will bring them up once because the bonding interface specifies them as slave interfaces, and then it tries to bring them up a second time because it sees the ONBOOT=yes parameter on a VLAN device.

   The issue occurs because after the devices are first brought up, the MAC addresses of the interfaces changes. Usually the first interface's MAC address is set on the rest of the interfaces. When the device is brought up a second time, the MAC address of the interface no longer matches the HWADDR field. The network scripts will wait for an interface with a matching MAC address to appear, which does not occur.

   Workaround

   Set the ONBOOT parameter to “no” on the slave interfaces. This prevents the system network scripts from trying to bring up the interface a second time. This will not negatively affect the operation of the bonded interface because the ifcfg-bond0 file still has ONBOOT set to yes. The networking scripts will attempt to bring up the bond0

interface on boot, which will cause the slave interfaces to be brought up and configured correctly.

If you are only using bonding without VLAN, the network scripts do not bring up slave interfaces, so the issue is avoided. The issue occurs only when bonding is on top of VLAN interfaces.

9. **Devloss timeout after swapping ports.**

   The driver may not finish discovery when two initiator ports are swapped. This causes all devices accessible through one or both of these initiator ports to time out and all I/O to fail.

   Workaround

   Do one of the following:

   - When swapping cables, replace each cable, one at a time, and allow discovery to finish before replacing the next cable. To determine if discovery is finished, read the "state" sysfs parameter.
   - When swapping cables, allow devloss timeout to fire before replacing the cables. (This fails all outstanding I/O.)

10. **LILO Boot Loader is not supported on i386 and x86_64 architectures.**

    The LILO Boot Loader on i386 and x86_64 architectures is not supported for this driver. If the LILO boot loader is used, after the LPFC driver package is installed and upon reboot an incorrect initial ramdisk is used, and the system might not boot correctly.

    Workaround

    The boot loader supported with this driver is GRUB, which is the default boot loader for most of the Linux distributions. LILO is an older boot loader used on i386 and x86_64 architectures only. GRUB works correctly with the driver package's installation script.

11. **Suspend to disk command results in a kernel Oops.**

    If you attempt to suspend to disk using the command:

    ```
    #echo disk > /sys/power/state
    ```

    the LPFC driver encounters a kernel Oops.

    The sysfs parameter "/sys/power/state" is used to suspend and resume the system. The LPFC driver does not support the suspend to disk and resume command. Do not attempt to use this sysfs parameter when the LPFC driver is loaded.

    Workaround

    None.

12. **Potential error messages during the driver kit removal process.**

    As part of the driver kit removal process initiated via the "lpfc-install -u" command the previous in-box LPFC driver version (driver version part of the Linux distribution), which was saved as part of the current driver kit install process, is restored and becomes the active driver. However, the driver kit un-installation process by design does not remove any entries in the Linux distribution configuration file (modprobe.conf). As such, parameters that would have been valid for the just-removed driver versions and entered in the modprobe.conf file, are also used to load the just-restored in-box driver version. This potentially create problems when:

- The just-removed driver version might include module parameters that did not exist in the older just-restored driver version, and
- One or more of these module parameters are included in the configuration file (modprobe.conf).

If the above criteria are met, you see an error message during the uninstallation process of the driver kit, such as:

```
Loading LPFC Driver .FATAL: Error inserting lpfc
(/lib/modules/<kernel_revision>/kernel/drivers/scsi/lpfc/lpfc.ko):
Unknown symbol in module, or unknown parameter (see dmesg)
```

For example, this issue can be observed is when uninstalling an 8.2.0.x driver kit, which had DH-CHAP functionality enabled, on a Linux distribution with an older 8.1.10.x in-box driver version.

Workaround

If such an error is seen during the kit removal process, you need to edit the Linux configuration file (modprobe.conf) and remove all entries that list LPFC driver module parameters, that is entries that start with:

```
options lpfc ...
```

Then attempt to uninstall the driver kit again.

Note: To find the module parameters supported by an LPFC driver module, type:

```
# modinfo <driver_dir>/lpfc.ko
```

13. Potential connection loss due to an FCF Failover issue with a Cisco FCoE switch.

An issue was discovered with Cisco Nexus 5000-series FCoE switch firmware 4.1(3)N2(1) or earlier, in NPV mode, that may cause the loss or interruption of SCSI connections when used with the Emulex OneConnect UCNAs. The switch incorrectly sends out a Discovery Advertisement to All-ENode-MACs from the FCF MAC with which the FC uplink was down. The end result is that sometimes the UCNA hangs on to an offline FCF or experiences back-to-back FCF failover and it may potentially lead to a Linux SCSI mid-layer devloss timeout.

Workaround

It has been verified that Cisco's 4.2(1)N1(1) release has corrected the issue. Emulex highly recommends that you upgrade your Cisco Nexus 5000-series FCoE switch firmware to 4.2(1)N1(1) or later to avoid this issue. If you decide to use 4.1(3)N2(1) or earlier firmware with your Cisco Nexus 5000-series FCoE switch and this issue is encountered, increase the FC transport dev_loss_tmo parameter to 60 seconds. This can be accomplished in one of two ways:

- Update the FC transport dev_loss_tmo parameter. For example:

```
# echo 60 > /sys/class/fc_remote_ports/rport-3:0-1/dev_loss_tmo
```

-or-

- Update the LPFC driver's lpfc_nodev_tmo parameter. For example:

```
# echo 60 > /sys/class/scsi_host/host3/lpfc_nodev_tmo
```

14. **Inband management connection loss during FCF failover with Brocade FC switch.**

    During FCF failover from one FC uplink to another in configurations with a Cisco FCoE switch in NPV mode and a Brocade DCX Director FC switch with firmware 6.1.1a, there can be cases where the OneConnect UCNA symbolic node names of the FCoE UCNA interfaces involved in the FCF failover disappear. This is observed from the OneCommand Management application, where the inband communication from one of the UCNAs to the other involved in the FCF failover is lost.

    During the FCF failover from one FC uplink to another, although all the OneConnect UCNAs failed over successfully for fabric logging and target rediscovery, it was observed that occasionally the Brocade DCX Director FC switch failed to establish the initiator-to-initiator forwarding with the UCNAs involved in the failover. This is based on the requirement that the FC switch should forward the initiator-to-initiator I/Os received from the FCoE uplink back into the same FCoE uplink. This issue has been observed with the Brocade DCX switch running firmware version 6.1.1a.

    Workaround

    None.

NIC Version 10.0.803.12 Known Issues

1. **ifconfig displays "RUNNING" flag when the cable is not connected.**

   This is a known issue in the Linux operating system stack.

   Workaround

   Use "ip link show" to properly display the "Oper status" value.

2. **On RHEL 6.3 (or earlier) and SLES 11 SP2 (or earlier) systems, if the NIC driver is unloaded in a hypervisor when VFs are assigned to a guest operating system, unexpected behavior may result.**

   Workaround

   To move to a different NIC driver or load the driver without enabling VFs (num_vfs=0), reboot the host.

3. **Kdump is not supported when SR-IOV is enabled.**

   Workaround

   None.

4. **PING is not working when attempting to bridge the 1G or 10G ports to the virtual machines when SR-IOV is enabled for 10G ports in the BIOS.**

   This issue occurs due to limitations of the virtual Ethernet bridge. All transmitted broadcast packets are looped back by the controller. This affects the functionality of the Linux bridge, as it appears as if the same ARP broadcast packets are received on two different interfaces.

   Workaround

   1) Set the aging of the bridge to 0 using the following command:

      "brctl setageing <bridge> 0"

      This causes the bridge to behave like a hub and flood the packet to all the ports (except the one on which the packet arrived) every time. This may impact

performance. If you have only two interfaces on this bridge (one NIC interface and virbr0-nic), there is no performance impact.

2) Another option may be to use the MacVTap interface to the guest instead of the bridge interface. The MacVTap interface is only available in the RHEL 6.3 or newer kernels.

5. **Unable to ping the RHEL 6.4 KVM guest (VM) from a remote host when using the MacVTap driver and IPv6.**

   When an IPV6 unicast address is configured, the Linux kernel also configures an Ethernet multicast MAC address in the format 33-33-xx-xx-xx-xx. In the context of a guest operating system (VM), the mcast address that needs to be configured comes into existence after the VM is booted up and the unicast IPv6 address is configured in the guest operating system.

   When using a MacVTap interface to bridge the NIC interface to the VM's interface, the previously-mentioned Ethernet MAC address configuration in the VM does not reach the MacVTap interface automatically. Since the MacVTap interface does not put the NIC interface into promiscuous mode, this results in an IPv6 ping failure from the remote host to the VM's interface.

   Workaround

   Select one of the following workaround options:

   - Use the following command:

     ```
     ifconfig eth<x> allmulti
     ```

     where "eth<x>" is the interface in the hypervisor to which the MacVTap interface is bridged. However, this causes all multicast traffic to be received.

   - Alternatively, you can configure the unicast IPv6 address (assigned to the VM interface) on the MacVTap interface in the hypervisor. This results in the Ethernet mcast MAC address being configured in the NIC interface.

     Use the following command:

     ```
     ifconfig macvtap<x> inet6 add <ipv6 address>
     ```

     where "macvtap<x>" is the MacVTap interface in the hypervisor and "<ipv6 address>" is the unicast IPv6 address assigned to the VM interface.

6. **When SR-IOV is enabled, NIC priority group (PG) and priority flow control (PFC) are not supported.**

   Workaround

   None.

7. **For OCe11100-series adapters, when the driver is loaded with num_vfs=32, the initialization of two of the VFs fails and only 30 VF interfaces are created.**

   Workaround

   None.

8. **In certain configurations, timeout errors may occur during maximum performance socket testing.**

Workaround

The default Linux driver settings may not be appropriate for optimal performance in all scenarios. If performance appears to be lower than expected, there are several driver and system settings that can be modified to improve performance. See the server documentation to determine the correct system settings and the optimal memory and processor configuration.

Along with proper hardware configuration, some driver settings can be modified to improve performance. The following are some recommended settings to examine while tuning for better performance. See the *Emulex Drivers for Linux User Manual* for an explanation of the available settings.

All Linux:

- Run the perf_tune_benet.sh script (included with Linux NIC driver)

Linux Xen:

- Only configure interfaces created with the "netfront" source model and ignore the second set of interfaces created with "8139cp"

Linux KVM:

- Use the "virtio" device model instead of "Hypervisor Default"

9. **The description for OCe14102-U3-D shows as 'unknown device 0720' in the Xen Center for Xen 6.1.**

   Workaround

   Run the update-pciid command to display the correct adapter name.

10. **Firmware dump using ethtool-W is not supported for OCe14000-series adapters.**

    Workaround

    None.

11. **For OCe14000-series adapters, if the port is not linked up to the switch at the time of driver loading with a non-zero num_vfs parameter, ethtool / ifconfig will report that the physical link is detected, even though there is no physical link.**

    Workaround

    None.

12. **When a CNA is configured in a NIC + iSCSI profile, the NIC and iSCSI traffic could be configured to share the total bandwidth and NIC traffic will be assured a minimum bandwidth.**

    In such a configuration, a VF interface will inherit the minimum bandwidth of the PF.

    Workaround

    None.

iSCSI Versions 10.0.719.1006 and 10.0.803.7 Known Issues

1. **System locks up on the next kmalloc call from any kernel entity when the iSCSI (be2iscsi) driver, released as part of the RHEL 6.1 distribution kernel (inbox version), is unloaded.**

   This issue is caused by a defect in the RHEL 6.1 kernel's iSCSI boot code. There is no known workaround if you plan to use the inbox iSCSI driver. A patch to the kernel iSCSI boot code should correct it. However, if you do not plan to unload the inbox iSCSI driver while the system is running, the issue will not be seen and everything should operate normally.

   The following workaround applies if, and only if, you plan to use the out-of-box iSCSI driver for iSCSI boot installation, which can be found in the DUP ISO that is provided. When you install RHEL 6.1, the inbox iSCSI driver is automatically loaded first, which requires Anaconda to unload the inbox iSCSI driver and then load the out-of-box iSCSI driver from the DUP ISO. This process causes the system to lock up due to the previously-mentioned issue with unloading the inbox iSCSI driver. The workaround prevents the inbox iSCSI driver from loading, and then loads the out-of-box iSCSI driver. The only way to prevent the inbox iSCSI driver from automatically loading without affecting other modules is by blacklisting the be2iscsi module name at the installation boot prompt.

   Workaround

   During the RHEL 6.1 boot media installation, you must first prevent the inbox iSCSI driver from loading by blacklisting it with the following command:

   ```
   dd blacklist=be2iscsi
   ```

   This command blacklists any iSCSI module with the “be2iscsi” name, including the out-of-box iSCSI driver. To bypass this limitation, the device driver that is included in the DUP ISO has been renamed to “beiscsi”, to differentiate it from the standard “be2iscsi” which is part of the RHEL6.1 distribution. Therefore, the driver module installed as part of the DUP ISO installation will be “beiscsi”.

   To address the driver module name difference, you must subsequently install a later version standard iSCSI (be2isci) driver that installs the driver binary RPMs. As part of the driver binary RPM installation process, the previously installed “beiscsi” module will be removed, and the previously blacklisted “be2iscsi” module name will be removed from the OS blacklist. As a result, the standard “be2iscsi” driver module will be installed and used.

   If you must install a previous version (downgrade) from the “beiscsi” driver version immediately after installing it via the DUP ISO process, it is highly recommended that before downgrading the “beiscsi” driver, you first upgrade or overwrite (with the same driver version as the “beiscsi” one included in the DUP ISO) using the driver binary RPMs. This way the RPM installation process will properly restore the iSCSI (be2iscsi) driver naming scheme as described above.

2. **In Linux, the device driver no longer loads in the operating system after a kernel update.**

   During the installation of a Linux kernel update, the following message (or a similar message) may be seen:

```
Cannot determine dependencies of module be2iscsi. Is modules.dep up
to date?
```

Workaround

Issue the following commands from the Command Line Interface (CLI) to resolve any driver conflicts. The [new-kernel] and [old-kernel] statements below must be replaced as appropriate.

a) cd /lib/modules/[new-kernel]/weak-updates/

b) ln -s /lib/modules/[old-kernel]/updates/be2iscsi/be2iscsi.ko

c) ln -s /lib/modules/[old-kernel]/updates/lpfc.ko

d) ln -s /lib/modules/[old-kernel]/updates/be2net/be2net.ko

Note: Steps b through d may not all be necessary depending on the personality of the installed UCNA.

e) depmod 2.6.18.60-0.93.1-smp

f) mkinitrd

g) shutdown -r now

After this workaround is applied, the server can boot to the new kernel.

3. **System hang may occur while booting with the iSCSI personality mode and inbox drivers on RHEL 5.9 systems.**

   Workaround

   Install the out-of-box iSCSI driver (be2iscsi) while the adapter is in the NIC personality mode, or use the out-of-box driver DUD while installing the operating system.

4. **There is a known Dracut issue that occurs on some systems with specific network settings, where iSCSI BIOS can fail to boot from the iSCSI LUN and kernel panic can occur.**

   Workaround

   a) Boot the system with the "rdshell" option in the kernel parameter. When the system cannot find or mount the boot partition, it will exit to the basic command shell.

   b) Manually run "iscsistart -b". This will add the boot target to the system.

   c) Type "exit", and the system will continue to boot normally.

5. **When Oracle UEK 6.5 x64 and RHEL 6.x and 5.x systems are used with the OCe14102-UM adapter with iSCSI/NIC enabled, iSCSI BIOS can fail to boot from the iSCSI LUN and kernel panic can occur.**

   Workaround

   a) Boot the system with the "rdshell" option in the kernel parameter. When the system cannot find or mount the boot partition, it will exit to the basic command shell.

   b) Manually run "iscsistart -b". This will add the boot target to the system.

6. **The Open-iSCSI administration utility binds the MAC address of each iSCSI port as an identity to create a configuration database. If a MAC address for an iSCSI port**

changes, the configuration data will be invalid. MAC addresses can be changed via firmware updates, or by changing the profile or personality of the adapter.

Non-boot persistent sessions will no longer work. Therefore, all previously mounted partitions will not be found. When iscsid attempts to open sessions through the iscsi port with a changed MAC address, expect to see the following message:

```
beiscsi_ep_connect shost is NULL
```

Workaround

Clean up the saved configuration and then recreate a new interface, discovery, and login.

7. **The open-iSCSI driver be2iscsi depends on some OS utilities and packages. If these packages are not present, the session establishment may not be possible after successful installation of the driver rpm.**

   Workaround

   Normally, a system previously installed with iSCSI hardware will have these packages installed automatically. In some corner cases, such as a new iSCSI adapter that was installed after the OS installation, these rpm packages will need to be installed manually. These packages can be found on the installation CD.

   For RHEL6.x and RHEL7, the dependencies will be:

   - dracut-network
   - iscsi-initiator-utils

   For SLES11spx, the dependency will be:

   - open-iscsi

8. **In systems with an open-iSCSI be2iscsi driver installed but not used as bootable, any iSCSI LUN configured in iSCSISelect will not get detected automatically. When the open-iSCSI be2iscsi driver is installed as bootable, any non-bootable iSCSI LUNs configured in iSCSISelect will not get detected automatically.**

   Workaround

   Configure the iSCSI LUNs using iscsiadm or OCM.

Technical Tips

1. **The standard 'make' compile command does not work as expected for the iSCSI driver.**

   Workaround

   The following ‘make’ command must be run from within the driver src directory:

   ```
   make -C /usr/src/<kernel dir> M=`pwd` CONFIG_BE2ISCSI=m
   ```

   Where the <kernel dir> is:

   For RHEL 6.5 SS#:

   ```
   /usr/src/kernels/2.6.32-431.el6/
   ```

   For SLES 11 SP# default variant:

   ```
   /usr/src/linux-obj/x86_64/default
   ```

2. **Additional physical NICs added to the XenServer do not appear in XenCenter.**

   Workaround

   See the following link for instructions on adding an additional physical NIC to the XenServer.

   http://support.citrix.com/article/CTX121615/

3. **The Open-iSCSI driver persistent targets are maintained in the Host Nodes database. The driver recognizes only the boot-target that is persistent, and is not aware of any non-boot persistent targets on the adapter.**

   Workaround

   Do not use iSCSISelect to configure persistent non-boot targets for all variants of RHEL 6, RHEL 7, SLES 11, and SLES 12 releases.

Note: References to OCe11100 series products also apply to OCe11100R series products.