# Xyratex PMC 8001 SPC Driver User Guide

Version 3.00

**Authors: Tim Pearce**
**Wing-Hon Lam**
**Lei Feng**

**Owner: Tim Pearce**
CPD Firmware Development Manager
Xyratex Technology Ltd
46831 Lakeside Blvd
Fremont
CA 94538, USA

**+408 421 4506**
+510 687 5263

tpearce@xyratex.com

**Date last saved:** 12/9/2010 3:04:00 PM

PN 0966341-01 EC 1014790

**Xyratex Confidential**

## Document change history

| Date | Docn Revision | Change | Initials |
|---|---|---|---|
| 12 July 10 | 0.01 | Initial layout and content | TP+WHL |
| 15 July 2010 | 0.02 | Some fixes | TP+WHL |
| 19 July 10 | 0.03 | Tidied | TP |
| 19 July 10 | 0.04 | Added rescan info | WHL |
| 19 July 10 | 0.05 | Added Package Contents | TP+WHL |
| 02 Aug 10 | 1.00 | Added WWN details | TP+WHL |
| 02 Aug 10 | 1.01 | Directory name changes | TP |
| 08 Aug 10 | 1.02 | Updated version reporting | TP |
| 09 Sep 10 | 1.03 | Parameter tweaks | TP |
| 16 Sep 10 | 1.04 | Changed firmware download and tweaked install method | TP |
| 20 Sep 10 | 1.05 | Better install instructions and modified all places to reference them. | TP |
| 20 Sep 10 | 1.06 | Review comments folded in and some wording tweaks for clarity and to stick to the rules of English language | TP+LF |
| 22 Sep 10 | 1.07 | Added detail on driver unload | TP |
| 04 Oct 10 | 1.08 | Added brief make instructions | TP |
| 14 Oct 10 | 1.09 | Modified download technique to include reference to inbuilt firmware | TP |
| 19 Oct 10 | 1.10 | Added driver parameter summary | TP |
| 26 Oct 10 | 1.11 | Typos and cleanup | TP |
| 18 Oct 10 | 1.12 | Reviewed | TP |
| 19 Nov 10 | 2.00 | Added minor review comments for final phase 2 release | TP |
| 09 Dec 10 | 2.01 | Correction in hotplugging | TP |
| 17 mar 11 | 3.00 | P/N | TP |

## Table of contents

## Table of tables

## Introduction

This document is intended for use by engineers installing the driver, supporting it and trouble shooting it

You must ensure you have the latest document at all times as it is updated frequently.

All Linux commands are case sensitive and should be entered in a Linux terminal (also called a command line or command prompt).

Please also refer to "Errata" on page 18.

## Package contents

There are four separate packages delivered with the driver. The table below outlines each package. Note that RPM is RedHat Package Manager. The script only version of the installer is provided for people who do not have RPM in their Linux.

| | RPM based | Script-only zip |
|---|---|---|
| **Binary Only** | pm8001-0.1.36-e5.src.rpm | pm8001-0.1.36-e5_bin.tar.bz2 |
| **Source and binaries** | pm8001-0.1.36-e5.x86.rpm | pm8001-0.1.36-e5_src.tar.bz2 |

**Table 1 - Installer file functional defintion matrix**

Note that the `e5` above is highlighted because this number will need substituting with your actual driver version number which starts at 1 and counts upwards with successive builds where it is prefixed with 'e' for engineering builds and 'f' for formally released builds.

You only need to use one of the installers in the matrix above.

## Overview

The Xyratex PMC 8001 Driver is a kernel object that allows Linux systems to talk to the PMC-8001 SPC (SAS Protocol Controller) chip on the Stingray motherboard.

The SPC is effectively an embedded HBA so this driver should be thought of as an HBA driver. The SPC is a SAS (Serial attached SCSI) controller so the bulk of data passed through the driver conforms to the SAS and SCSI standards.

The driver is designed to fit in Linux under the libsas layer. This means it will present devices and allow access in the same way as other standard Linux SCSI device drivers.

Before installing or using the driver it is important that you read this document in order to avoid accidental damage to the operating system. A thorough backup is recommended before installing.

## Pre-requisites

As indicated above, libsas must be installed before installing the Xyratex PMC 8001 driver. If not already installed libsas can be installed using the following command

```
modprobe libsas
```

**Note:** All Linux commands shown are case sensitive.

In order to ensure that firmware can be downloaded you must ensure that the 4 firmware files supplied by the driver have been copied to /lib/firmware. If you use the Xyratex supplied installation method this will be done for you. If the files are not in this location then the driver will use its default inbuilt firmware for download with factory settings.

## Licensing

This driver contains open-source program code conformant with the GPL v2 license schemes. If included in any distributable software this license must be quoted and maintained and source must be available on request by anybody. All modifications are subject to the same license. Xyratex makes no warranty and provides no protection against open source claims on any driver or tool code whether binary or source file.

## Supported Operating Systems, hardware and configurations

The following hardware is supported

| | |
|---|---|
| Motherboards: | Stingray (generic) |
| | Stingray (non-generic) |
| PMC SPC HW version: | Rev C chip |
| PMC SPC FW version: | 01.10.05.00 (built-in default for driver) |

The following operating systems are supported by the driver. Note that all are Intel CPU based and are all 64 bit.

A) CentOS 5.4 kernel 2.6.18
B) CentOS 5.4 kernel 2.6.18-164
C) CentOS 5.5 kernel 2.6.18-194
D) RedHat Enterprise Linux (RHEL) 5.4 kernel 2.6.18-164
E) RedHat Enterprise Linux (RHEL) 6.0 kernel 2.6.32 (build from source)
F) RedHat Enterprise Linux (RHEL) 6.0 kernel 2.6.33 (build from source)

The following features are not supported in the driver

1) SMP (SAS Management protocol) sent by client/user applications
2) STP (SATA Tunneling Protocol) – this means no pure SATA support

The following features are supported in the driver

1) Firmware download at start of day (no firmware detected on SPC)
2) Firmware download after start of day (SOD) (via reload of driver)
3) All SCSI data and SES Commands
4) Basic debugging
5) Basic error handling
6) Hotplug of drives with automatic rescan of drives
7) WWN modification

## Installing the driver

For this section you need to be logged on as an administrator which is usually the user Id called 'root'.

As stated above libsas must be installed first. This is usually part of standard Linux distributions so from a terminal use the following command to install libsas:

```
modprobe libsas
```

**Note:** all commands shown are case sensitive.

Ensure the appropriate Xyratex PMC 8001 SPC Driver installation file (see "Table 1 - Installer file functional defintion matrix" on page 3) is located in a directory that you know e.g. `/usr/tmp` and either unzip this package into the same directory or use RPM to unpack it into your selected directory.

All installers contain a script called `pm8001install`. This should be executable after unzipping however if not you can change its permissions so it is executable as follows:

```
chmod 777 pm8001install
```

The install program can take several parameters. Generally, they can be combined in any order. Items in hex must be preceded by 0x and be the full length permitted e.g. 0xabcdef0123456789 or 0x25

| Parameter | Meaning | Example |
|---|---|---|
| *No parameters* | Install with no debug enabled and the default WWN | `pm8001install` |
| uninstall | Uninstall the driver.<br>Before uninstalling or unloading the driver please read "Errata" on page 18. | `pm8001install uninstall` |
| wwn4 xxx | Set up the driver to use 4x ports using the 64 bit WWN (16 hex digits) instead of xxx (this must be the full length WWN in hex and starting 0x)<br>See "WWN support and modification" on page 12 for more details.<br>Semantically this is mutually exclusive with the wwn8 parameter. | `pm8001install wwn4 0xabcdef0123456789` |
| wwn8 xxx | Set up the driver to use 8x ports using the 64 bit WWN (16 hex digits) instead of xxx (this must be the full length WWN in hex and starting 0x)<br>See "WWN support and modification" on page 12 for more details.<br>Semantically this is mutually exclusive with the wwn4 parameter. | `pm8001install wwn8 0xabcdef0123456789` |
| db xxx | Install the driver with a new debug level set to xxx | `pm8001install db 0x25` |

| | | |
|---|---|---|
| | which is a 2 digit hex number. See "Debugging" on page 9 for details on debug levels. | |

**Table 2 - installer parameters and functions**

**Note:** The driver takes care of halting and de-installing any pre-existing driver. It then installs and loads the driver and ensures it is set up for auto-load at each Linux boot time. Before uninstalling or unloading the driver please read "Errata" on page 18.

**Note:** The driver installation contains an image of the SPC firmware held in four separate binary files. If the driver starts up and discovers the SPC has no firmware loaded it will download its image onto the SPC automatically.

**Note:** If there are no drives connected at the time of installation, or when loading of the driver, the driver will still load but no devices will be found.

**Note:** Do not combine the `uninstall` parameter with any other parameter.

## *Checking it installed – version reporting*

Once installed you may wish to check which version is installed to ensure it has all worked correctly

There are standard ways in Linux to get this information

The most common is modinfo e.g.

```
    modinfo pm8001.ko
```

In addition the kernel log contains information about which driver was started such as the entry below (see "Debugging" on page 9 below for details on logs and logging).

```
Jul 24 11:53:10 CentOS-Stingray kernel: pm8001 0000:04:00.0: pm8001:
driver version 0.1.36
```

## Debugging

The Xyratex PMC 8001 SPC Driver has 8 categories of debug messages as follows in order to allow you to decide the quantity of output you require. Note that enabling debug may affect performance and possibly stability especially under high load.

| Category | Bit Field | Hex Format |
|---|---|---|
| Error Message Logging | 0 | 0x01 |
| Driver Initialization Logging | 1 | 0x02 |
| Discovery Layer Logging | 2 | 0x04 |
| I/O Path Logging | 3 | 0x08 |
| libsas EH  Function Logging | 4 | 0x10 |
| IOCTL Message Logging | 5 | 0x20 |
| Misc. Message Logging Group 1 | 6 | 0x40 |
| Misc. Message Logging Group 2 | 7 | 0x80 |

**Table 3 - Debugging Categories**

The values specified in the table above get used when the driver is installed.

To calculate this parameter start by OR-ing the fields in the table above that match what you wish to view. For instance:

```
      Bit 0 + Bit 2 + Bit 5
=     0x01  + 0x04  + 0x20
=     0x25
```

Then you will need to reinstall the driver in debugging mode. See "Installing the driver" on page 6 for details on how to do this. If you plan on sending debug data to Xyratex please use a debug level 0xFF. Make sure that the debugging parameter argument is set to the value calculated above e.g.

```
    pm8001install db 0x25
```

```
(Replace 0x25 with whatever your calculated category combination was)
```

This will cause a driver to be unloaded first if already in operation. Before uninstalling or unloading the driver please read "Errata" on page 18.

Now insure that klogd is running as follows on the terminal

```
    ps -ef
```

If klogd is not listed in the output from the above then

```
    klogd&
```

All output will be sent to the standard Linux kernel log in /var/log/messages.

## Hot plugging drives or devices

When drives are hot-removed or hot-inserted a drive rescan is performed automatically.

Hot plugged drives may take a few seconds to appear or disappear from the device list in /dev. It is important to allow time for the drive to completely spin up and be ready for IO - this can sometimes be more than 30 seconds. When removing a drive always allow at least 10 seconds before reinserting it.

Should you need to force Linux to rescan the available devices you can do this by using the following technique:

1. Find what's the host number for the HBA:
   ```
   ls /sys/class/sas_host/
   ```
   (You will see items listed such as host1 or host2. These are generically referred to as host# below)

2. Ask the HBA to rescan the SCSI devices on that HBA replacing host# with the value gleaned from step 1 as follows :
   ```
   echo - - - > /sys/class/scsi_host/host#/scan
   ```
   (The wildcards "- - -" tell Linux to look at every channel, every target, every LUN. There is a space between each dash.)

When hotplugging drives you must allow time for the whole Linux system stack to settle (i.e. to have recognized and finished handling the drive state change) between individual removals and insertions. This may be several seconds in some systems.

## WWN support and modification

A root WWN is stored by the SPC chip itself. This is a 64 bit WWN and is used to generate all other WWNs needed by the SPC. The WWN is persisted by the SPC in VPD (Vital Product Data). Once the WWN is correctly set it is always persisted.

There are two modes for the WWN. These define whether you want the system configured as a singled 8x wide-port coming into the SPC or as dual 4x ports.

You need to use different parameters depending upon which mode you need

To set the VPD you can either

a) It can be hard coded into SPC firmware (highly non-recommended)
b) Load a whole VPD image when programming the VPD in manufacturing. This is likely supportable by manufacturing only.
c) It can be specified as a parameter when loading the driver. Once set it remains persisted. This is supportable as a method of field change although it is desirable NOT to have all customers changing WWNs so the parameter should not be in any user guides – just available for FAEs and OEMs

To program the root WWN you must install or re-install the driver as shown in "Installing the driver" on page 6.

The full 64 bit hex value (16 hex digits) should be used with no spaces, a mandatory 0x as the leading 2 characters and also using hex digits (0-9a-f) only in lower case. Note that if you omit the 0x at the beginning the value will be interpreted as decimal and a translation to hex will be done which means you may not get the values set that you desired.

These parameters cause 2 WWNs to be persisted in both cases for each half of the wide-port – however the two WWNs are the same for a 8x so it appears as a single port. The VPD storage for each format is as follows:

<u>4x (wwn4)</u>
```
0x009911ff33e21a00
0x009911ff33e21a01
```

<u>8x (wwn8)</u>
```
0x009911ff33e21a00
0x009911ff33e21a00
```

Note that these are root WWNs. All other WWNs used in the SPC are derived from these so an appropriate gap between WWNs assigned during manufacturing to avoid clashes is strongly recommended. An incremental gap of 16 is recommended which implies leaving the last digit as zero in the hex field.

**Note:** The WWN setting parameters (and other methods) should only be used when forcing the WWN to change and not on every driver load in order to cut down on VPD write counts. The programmed value is held in EEPROM and persisted across reboots.

The WWNs are stored in EEPROM at offset 0x00 of the SPC EEPROM user space section. They are 64 bits and stored in little-Endian format.

## Firmware download

As previously mentioned the driver supports SPCs with and without flash. Having no flash implies doing a firmware download to the SPC when the device driver is first started up. The Xyratex installers place the downloadable firmware files in the correct location at install time (there are 4 binary files) and ensures they get downloaded at driver load.

The four files to be download which will normally be supplied by the chip vendor.

These 4 files will need to be placed into /lib/firmware in order to be located and downloaded correctly by the driver. If you do not do this the driver will download its 4 default firmware files that are kept internally to the driver with factory default settings. This may downgrade your enclosure if you were already at a higher level but is useful when handling kick-start installs and other activities that remove /lib/firmware from the Linux distribution.

If you wish to change or upgrade the firmware on the SPC after Linux has booted simply replace the firmware files in /lib/firmware with your new files and then re-install the driver as shown in “Installing the driver” on page 6.

Currently the SPC 1.10 firmware is included in the package. The four binary download files are named as follows:

```
aap1img.bin
ioping.bin
ilaimg.bin
istrimg.bin
```

## SCSI Error Handler

The Linux environments possess a number of error handlers for SCSI errors. One of these handles drives that have been stopped – for instance from a SCSI Start/Stop Unit command.

When the handler receives an IO failure due to the drive being stopped it automatically restarts the device. This is not always desirable in dual controller environments where one controller might wish to be able to halt a device and have it stay halted.

To assist with this the driver has a parameter to disable the SCSI error handlers. The default is that the handler is enabled however to disable the handler you need to start the driver with the parameter

```
pm8001_scsi_ehandler=0
```

**Note**: Setting the value to 1 ensures it is enabled however is effectively a no-op.

The parameter is intended as a test-only parameter.

## Driver parameter summary

The driver supports the following parameters.

**Note:** If you wish to persist most of these parameters you will need to modify your modprobe.conf file to add the parameters into it when the driver is started.

| Parameter | Meaning | Example | Default |
|---|---|---|---|
| pm8001_logging_level | Set the debug output level | pm8001_logging_level=0x25 | 0x00. See "Debugging" on page 9. |
| pm8001_wwn_by4 | Change the stored WWNs to the value specified in 2x4 mode.<br><br>This is mutually exclusive with pm8001_wwn_by8.<br><br>It is recommended this parameter is not added to config files eg modprobe.conf. | pm8001_wwn_by4=0x123456789abcdef0 | If not specified the WWNs are left alone as the ones last stored. See "WWN support and modification" on page 12. |
| pm8001_wwn_by8 | Change the stored WWNs to the value specified in 1x8 mode.<br><br>This is mutually exclusive with pm8001_wwn_by4.<br><br>It is recommended this parameter is not added to config files eg modprobe.conf. | pm8001_wwn_by8=0x123456789abcdef0 | If not specified the WWNs are left alone as the ones last stored. See "WWN support and modification" on page 12. |
| pm8001_scsi_ehandler | Set to 1 to enable the default SCSI error handler. Set to 0 to disable it<br>Intended for test use only. | pm8001_scsi_ehandler=0 | Default is 1. See "SCSI Error Handler" on page 15. |

**Table 4 - Driver Parameters Summary**

## Building the driver

If you are using the RPM (RedHat package manager) version of the installation programs that contain source then the build will be done automatically for you as part of the RPM set up process.

If you are using the zip based installation then the process is still pretty simple. There is a dependency upon having GCC installed in your build environment however, assuming you have that, change your working directory to be the root folder where you unzipped the driver source to an run

```
make
```

If you are using one of the supported platforms the driver will now build.

# Errata

It is possible that attempts to unload the driver may fail. The root cause of this is that the reference counter for the driver may not get to 0. This only occurs in certain circumstances:

1) It generally requires an expander reboot before you enter this state – a rare activity – and even then may see the problem, and…
2) It requires you to load/unload your driver after boot, and…
3) It requires heavy driving of the *sg* portion of Linux – this is not the main data path

**Note:** The memory leakage is small – and requires use of libsas - and is about 8k per discovery.

The basic workaround is to reboot your system

The basic issue is caused by the driver reference count not being decremented some times under heavy sg load during a gem reboot. This means Linux vetoes any future attempt to unload the driver. The bug is 'caused' by the way sg operates and is outside the scope of this driver. The lack of cleanup in sg in these circumstances explains the memory leakage. Linux sg operates differently to sd and sd does not seem to show the same problem.