Galleon Tech Note

XSR Write Protection

Revision history:

Rev.	Date	Changes	Sign
1.0	29-Nov-16	Document created	SM
1.1	29-Jan-18	Updated section 1.6.1 with new script: sys-suspend-read-only	SM
1.2	1-Mar-18	Added section 1.6.2 to describe hardware write-protection	SM
1.3	26-Apr-19	Added section regarding RWTAB	NL
1.4	10-Feb-22	New logo	TG

Abstract

This document describes how to enable and manage write protection for a standard XSR configuration.

1 Write Protection

1.1 Introduction

Galleon XSRs have several non-volatile (NV) devices that are not accessible from standard operating software. However, periodic firmware upgrades on some NV devices require write access for special equipment or software. To ensure these devices are not inadvertently written to, hardware write protection is available.

The internal system drive SSD containing the operating system may also be write protected to satisfy security requirements.

1.2 Write-protected Devices

The following are examples of internal NV devices that can be write protected through hardware:

- Ethernet configuration PROMs
- PCI-Express configuration PROMs
- XMC devices
    - MCU FLASH
    - Configuration EEPROM
- System Drive (if drive is equipped with write protection)

1.3 Non Write-protected Devices

The following are examples of internal NV devices that are not write protected through hardware. In these cases, utilities exist to erase or program the NV devices to a known state.

- USB configuration PROMs
- SATA controller FLASH
- XMC devices
    - FPGA FLASH
- BIOS
- Removable Data Module (RDM)

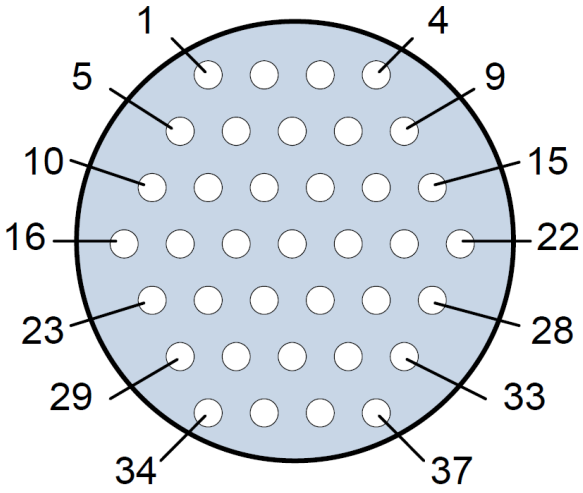
1.4 J1 Control and Management

The Galleon XSR has two digital I/O connectors of type Glenair Mighty Mouse series 801, size 13-37 in the rear panel. J1 provides the control and management interface. The connector provides the following control interface signals:

- 1x VGA Output
- 1x 1000 base-T Ethernet Interface
- 2x USB 2.0 Interface
- Miscellaneous

For normal head-less operation, only the Ethernet interface is required. The USB interfaces may be used for keyboard/mouse, data offload, etc. The miscellaneous connector contains a write protection signal amongst other signals.

XSR Connector type:	Glenair p/n: 801-009-07ZNU13-37SA
Mating Cable connector (example):	Glenair p/n: 801-007-16ZNU13-37PA
Mating Galleon Cable:	XSR-CBL-BRK37A-006



Mating Face View of XSR Connector
(Cable connector numbers are reversed)

Figure 1 J1 Control

1.5 J1 Write Protect Signal

The write protect signal is located on pin 12 of J1. The signal is marked as reserved, NVWE# or NVWP# in the delivered Interface Configuration Description (ICD) depending on the system configuration.

The signal is pulled up through an internal resister in the XSR such that when left disconnected, it is in an inactive state. When pulled (tied) to ground through pin 12 of J1, the signal is in an active state. The active-low state is factory configured as write enable (NVWE#) or write protect (NVWP#) depending on customer requirement. The default configuration is active-low write enable (NVWE#). In the default configuration, the XSR NV devices are write protected when pin 12 of J1 is left disconnected.

Galleon provides a special lab cable that provides access to the write protect signal on J1. The write protect signal is connected to pin 2 on the MISC DB15 connector on Galleon cable XSR-CBL-BRK37A-006.

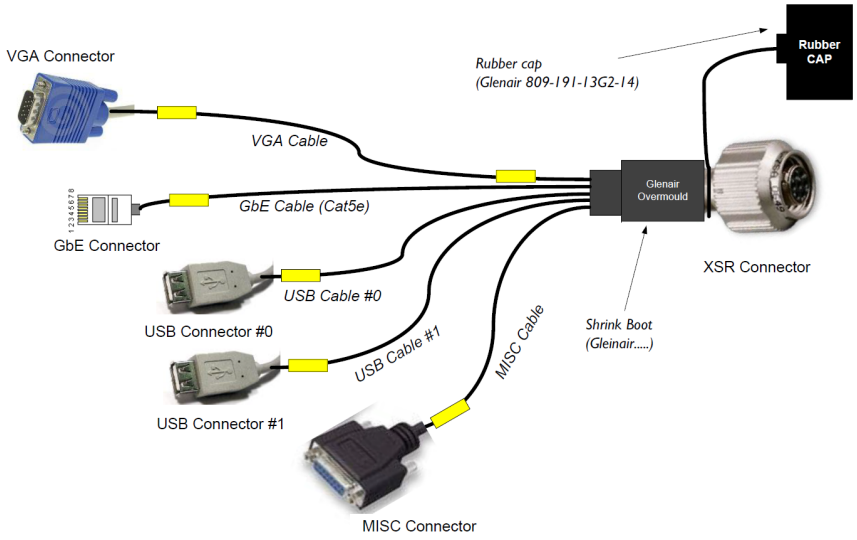


Figure 2 XSR-CBL-BRK37A-006

Using the lab cable, the NVWE# or NVWP# pin can be tied (pulled) to ground using a simple jumper between pins 2 and 9 (GND) on the MISC connector.

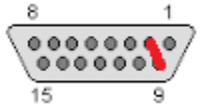


Figure 3 MISC DB15 Connector

1.5.1 Write-Protected System Drive

System drive SSDs are available with or without write protection built in. When using an SSD that is not equipped for write protection, the J1 write protect signal has no effect on the system drive. In this manner, the J1 write protect signal can be configured for NVWE# to protect other NV devices yet still enable writing of the SSD. This is the default configuration.

When using an SSD equipped with write protection, the NVWE# signal must be pulled low in order enable writing of the SSD. When NVWE# is pulled low, all other NV devices are also enabled for writing. Alternatively, when the write protect signal is configured for NVWP#, it is normally write enabled and must be pulled low to enable write protection on the SSD and other NV devices.

1.6 Read-Only Operating System Configuration

When the system drive SSD is configured as write protected, the operating system must be told to mount in read-only mode. This is initially done during the partitioning of the SSD during Linux Installation.

Depending on the O/S, a screen similar to the one shown below will allow for the disk to be partition in read-only mode. Both “noatime” and “ro” should be selected during initial portioning.

Once installation is complete, the operating system is placed in read-only mode, but also creates a RAM disk where all temporary files are stored (i.e. */tmp*, */var*, etc.) The contents of the RAM disk are lost when power to the system is removed.

When the system comes back up, the root and any other system partitions will be mounted read-only. All the files and directories listed in /etc/rwtab will be mounted read-write on the tmpfs filesystem in RAM. Additional files and directories can be added to rwtab to make them writable after reboot.

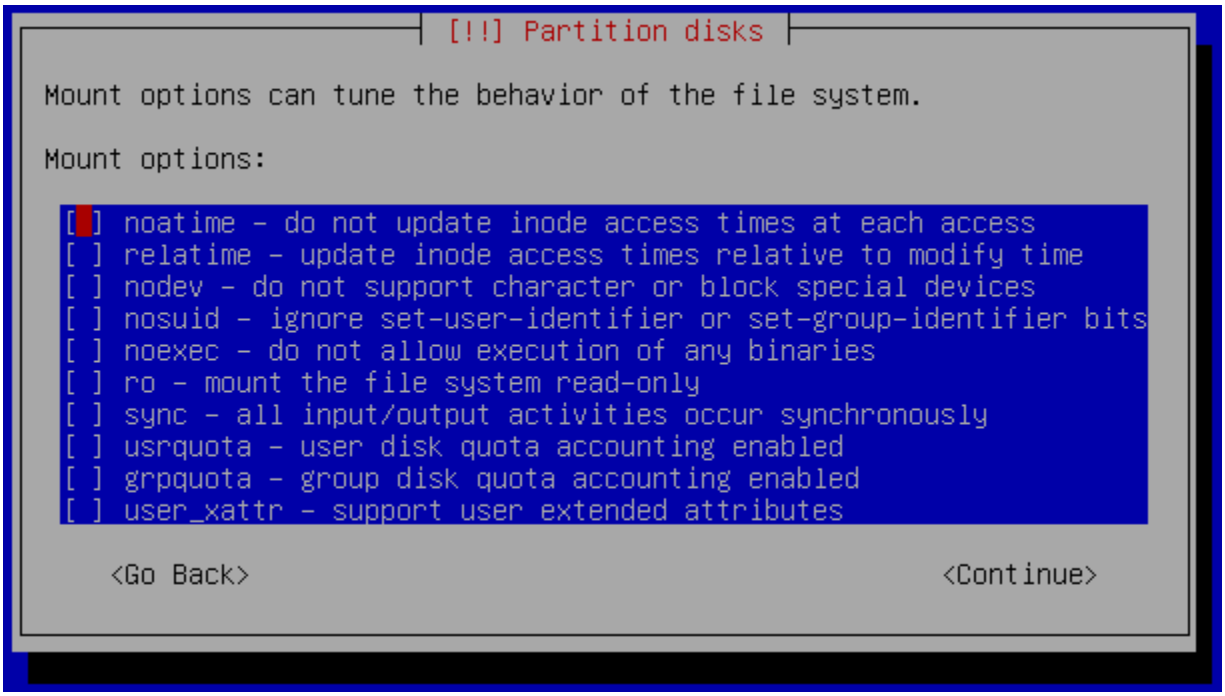


Figure 4 SSD Partitioning

1.6.1 Placing read-only system in write mode

To change any system parameters (i.e. IP addresses, user credentials), to add or modify software, etc, a utility is supplied to enable temporary write access to the system drive. The utility will change the mode of the file system to read/write.

Note that the write protection applies to the system SSD only. User data is normally stored on the removable data cartridges which are not write protected. The default mount point for user data is */data*.

The following steps are required to perform changes to the read only file system:

1. Make sure the hardware signal is configured to disable write protection (#NVWE pulled low on normally protected systems or #NVWP floating on normally enabled systems).

2. Boot the operating system

3. Run the following command to enable file system r/w operation:

 # sys-suspend-read-only

This script is used to disable the read-only configuration of the system drive temporarily, allowing the operator to perform maintenance that requires write access to the system drive.

The script starts by checking the state of hardware write-protection on the system drive. If hardware write-protection is activated, the script will just display an error message and terminate, leaving the system drive mounted as read-only.

If write-protection is deactivated, the script will remount the system drive as writable, and unmount all files and directories that are mirrored on RAM disk (tmpfs). The system drive will remain in this writable state until the system is rebooted.

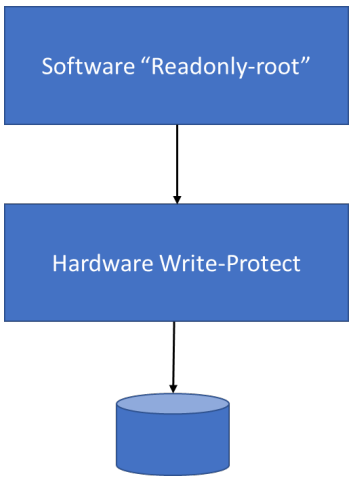
4. Perform the changes required to the system

5. Reboot to place system drive back into read-only state.

1.6.2 System Drive Dependencies

Galleon currently provides hardware write-protected system drives from two different manufacturers. Attempting to write to a write-protected drive will result in no data being written to the drive in either case. However, the response from the drive will differ between the two manufacturers.

When considering the response of the hardware write-protect, it is important to understand that the operating system will first determine validity of the write prior to writing data to the drive. An error will be returned from the operating system if it determines the operation is invalid for a read-only drive and the write will not be performed. In this manner, the “readonly-root” configuration takes precedence over the hardware write-protect.



When in “readonly-root” mode, the operating system will not attempt to write to any part of the root partition “/” of the drive.

As an example, if one were to attempt to create a directory in the root file system (e.g. #mkdir /newdir), the operating system would respond with an error that the directory cannot be created: “unable to create directory <…>, read-only file system”

This is not true for files and directories placed in the temporary filesystem located in RAM disk. They may be freely written because they are placed in RAM. Temporary files (/tmp directory) and log files (/var directory) fall into this category. Similarly, other files and directories may be placed in RAM by listing them in /etc/rwtab. These will be mounted read-write on the tmpfs filesystem upon boot up.

See https://access.redhat.com/solutions/40906 for additional information.

If, on the other hand, an operation is performed outside of the root file system, yet still on a partition of the write-protected drive, this is where the differences between drives will appear.

1.6.2.1 TCS Drive

If another partition of the write-protected drive is mounted under the mount point “/mnt”, the operating system does not know whether this is a local drive or remote mount of another drive. A write to “/mnt” will proceed from the operating system to the drive, but the drive will not allow it to happen because it is hardware write-protected. The data will not be written to the drive. On the TCS drives, a series of error messages will appear.

For example, when executing:

```
echo “test” > /mnt/test.txt
```

where ‘/mnt/’ is the mount point of a file system on a write-protected TCS Drive, you get 7 repetitions of the following error messages:

```
ata4.00: exception Emask 0x0 SAct 0x0 SErr 0x0 action 0x0
ata4.00: irq_stat 0x40000001
ata4.00: failed command: WRITE DMA
ata4.00: cmd ca/00:08:3f:20:04/00:00:00:00:00/e0 tag 0 dma 4096 out
        res 51/40:08:3f:20:04/00:00:00:00:00/e0 Emask 0x9 (media error)
ata4.00: status: {DRDY ERR}
ata4.00: error: {UNC}
ata4.00: configured for UDMA/133
ata4: EH complete
```

followed by:

```
sd 3:0:0:0: [sdc] Unhandled sense code
sd 3:0:0:0: [sdc] Result: hostbyte=DID_OK driverbyte=DRIVER_SENSE
sd 3:0:0:0: [sdc] Sense Key: Medium Error [current] [descriptor]
Descriptor sense data with sense descriptors (in hex):
```

```
    72 03 11 04 00 00 00 0c 00 0a 80 00 00 00 00 00
    00 04 20 3f
sd 3:0:0:0: [sdc] Add. Sense: Unrecovered read error - auto reallocate failed
sd 3:0:0:0: [sdc] CDB: Write (10): 2a 00 00 04 20 3f 00 00 08 00
Buffer I/O error on device sdc1, logical block 33792
lost page write due to I/O error on sdc1
```

1.6.2.2 Virtium StorFly Drives

In the same scenario described in Section 1.6.2.1, the Virtium drive will also not allow the data to be written to the drive, but no errors will appear. The drive will simply discard the data and not report any errors.

This is actually the preferred method, as the controller detects it is an invalid operation earlier in the process and does not present error messages as artifacts.

1.7 Conclusion

By properly configuring the write protect pin on the J1 connector, most NV devices on the XSR can be write protected. Special consideration to a write protected system disk should be given during installation and when performing updates for proper read-only system operation.

2 Appendix

2.1 Example: Adding files to RAM disk

There are times when files or directories need to be added to the RAM disk. For this discussion assume that myfavoriteprogram runs automatically when the system boots. Now myfavoriteprogram expects to read the file testlibdirfile.txt located in /var/lib/TESTLIBDIR/testlibdirfile.txt when it runs. In this example testlibdirfile.txt is a text file that is required for myfavoriteprogram to run.

Adding test directory and test file to /etc/rwtab that will remain persistent after reboot.

#Login to system

```
[root@XSR galleon]# ssh root@192.168.100.100
root@192.168.100.100's password:
[root@localhost ~]# cd /var/lib
```

#Show initial contents of /var/lib

```
[root@localhost lib]# ls
alternatives  dhclient  initramfs  misc          nfs       polkit-1  rpcbind   rsyslog  stateless  tuned
authconfig    games     logrotate  net-snmp      os-prober  postfix  rpm       samba    systemd    up2date
dbus          gssproxy  machines   NetworkManager  plymouth  rhsm     rpm-state  selinux  tftpboot   yum
```

#Run Suspend Read Only Script

```
[root@localhost lib]# sys-suspend-read-only
Remounting root file system as read-write ... OK
Checking that root file system is writable ... OK
Unmounting '/var/lib/stateless/writable' ... OK
Unmounting '/var/cache/man' ... OK
Unmounting '/var/log' ... ERROR
Unmounting '/var/lib/dbus' ... OK
Unmounting '/tmp' ... ERROR
Unmounting '/var/lib/dhclient' ... OK
Unmounting '/var/tmp' ... OK
Unmounting '/etc/adjtime' ... OK
Unmounting '/etc/resolv.conf' ... OK
Unmounting '/var/lib/NetworkManager' ... OK
Unmounting '/var/lib/systemd/random-seed' ... OK
Unmounting '/var/spool' ... OK
Unmounting '/var/lib/samba' ... ERROR
Unmounting '/var/lib/nfs' ... ERROR
Unmounting '/var/lib/net-snmp' ... OK
Unmounting '/var/lib/logrotate' ... OK
Unmounting '/etc/sysconfig/network-scripts' ... OK
```

```
Unmounting '/var/lib/dhclient' ... OK

WARNING: Failed to unmount 4 mirror directories
      The files in these directories are still stored in volatile memory,
      and any changes to them will be lost after a reboot.
```

The reason that four directories failed to unmount is due to the fact that those directories have open files in them. For example 'var/lib/samba' and 'var/lib/nfs' have files open because the samba and nfs services are running. If a file needs to be added to a directory that is not unmounted, then the service utilizing that file will need to be stopped and the 'sys-suspend-read-only script should be run again to unmount the directory.

Create test library TESTLIBDIR

```
[root@localhost lib]# mkdir TESTLIBDIR
[root@localhost lib]# cd TESTLIBDIR/
```

Add file test file testlibdirfile into TESTLIBDIR directory

```
[root@localhost TESTLIBDIR]# cat >> testlibdirfile.txt
This text was wrote to the /var/lib/TESTLIBDIR/testlibdirfile.txt with the write enable signal asserted.
```

Edit rwtab to include "files /var/lib/TESTLIBDIR"

```
[root@localhost TESTLIBDIR]# vi /etc/rwtab
```

The entry '/var/lib/TESTLIBDIR' has been added to the /etc/rwtab file.

```
[root@localhost TESTLIBDIR]# cat /etc/rwtab
dirs    /var/cache/man
dirs    /var/gdm
dirs    /var/lib/xkb
dirs    /var/log
dirs    /var/lib/puppet
dirs    /var/lib/dbus

empty   /tmp
empty   /var/cache/foomatic
empty   /var/cache/logwatch
empty   /var/cache/httpd/ssl
empty   /var/cache/httpd/proxy
empty   /var/cache/php-pear
empty   /var/cache/systemtap
empty   /var/db/nscd
empty   /var/lib/dav
empty   /var/lib/dhcpd
empty   /var/lib/dhclient
empty   /var/lib/php
empty   /var/lib/pulse
```

```
empty   /var/lib/systemd/timers
empty   /var/lib/ups
empty   /var/tmp

files       /var/lib/TESTLIBDIR
files       /etc/adjtime
files       /etc/ntp.conf
files       /etc/resolv.conf
files       /etc/lvm/cache
files       /etc/lvm/archive
files       /etc/lvm/backup
files       /var/account
files       /var/lib/arpwatch
files       /var/lib/NetworkManager
files       /var/cache/alchemist
files       /var/lib/gdm
files       /var/lib/iscsi
files       /var/lib/logrotate.status
files       /var/lib/ntp
files       /var/lib/xen
files       /var/empty/sshd/etc/localtime
files       /var/lib/systemd/random-seed
files       /var/spool
files       /var/lib/samba
files   /var/log/audit/audit.log
files       /var/lib/nfs
dirs /var/lib/net-snmp
files /var/lib/net-snmp/.snmp-exec-cache
files /var/log/gec-bit.log
files /var/lib/nfs/etab
```

Reboot the system and remove the jumper on pins 2 & 9.

```
[root@localhost TESTLIBDIR]# reboot
Connection to 192.168.100.100 closed by remote host.
Connection to 192.168.100.100 closed.
```

Log back into the system.

```
[root@XSR galleon]# ssh root@192.168.100.100
root@192.168.100.100's password:
[root@localhost ~]# cd /var/lib/
```

The contents of /var/lib now contains TESTLIBDIR.

```
[root@localhost lib]# ls
alternatives dhclient initramfs misc        nfs      polkit-1 rpcbind  rsyslog stateless tftpboot yum
authconfig   games    logrotate net-snmp    os-prober postfix rpm      samba   systemd   tuned
dbus         gssproxy machines  NetworkManager plymouth  rhsm    rpm-state selinux  TESTLIBDIR
up2date
```

```
[root@localhost lib]# cd TESTLIBDIR/
```

The file testlibdirfile.txt previously created previously is persistent upon reboot.

```
[root@localhost TESTLIBDIR]# ls
testlibdirfile.txt
```

The contents of testlibdirfile.txt previously created persistent upon reboot.

```
[root@localhost TESTLIBDIR]# cat testlibdirfile.txt
This text was wrote to the /var/lib/TESTLIBDIR/testlibdirfile.txt with the write enable signal asserted.
```

Add additional information to existing testlibdirfile.txt

```
[root@localhost TESTLIBDIR]# cat >> testlibdirfile.txt
This text was added to testlibdirfile.txt with the write enable signal NOT asserted.
The added text is written to the testlibdirfile.txt file but is not persistent upon reboot.
```

The contents of testlibdirfile.txt have been edited and remain until reboot.

```
[root@localhost TESTLIBDIR]# cat testlibdirfile.txt
This text was wrote to the /var/lib/TESTLIBDIR/testlibdirfile.txt with the write enable signal asserted.
This text was added to testlibdirfile.txt with the write enable signal NOT asserted.
The added text is written to the testlibdirfile.txt file but is not persistent upon reboot.
```

Add 2ndNONpersistenttestfile.txt to show that files can be added to the directory while the write enable signal is NOT asserted.

```
[root@localhost TESTLIBDIR]# cat >> 2ndNONpersistenttestfile.txt
This file was added to the /var/lib/TESTLIBDIR directory with write enable signal NOT asserted.
This file can be read and added to but is not persistent up reboot.
```

Show files contained in TESTLIBDIR prior to rebooting system.

```
[root@localhost TESTLIBDIR]# ls
2ndNONpersistenttestfile.txt  testlibdirfile.txt
[root@localhost TESTLIBDIR]# cat testlibdirfile.txt
This text was wrote to the /var/lib/TESTLIBDIR/testlibdirfile.txt with the write enable signal asserted.
This text was added to testlibdirfile.txt with the write enable signal NOT asserted.
The added text is written to the testlibdirfile.txt file but is not persistent upon reboot.
[root@localhost TESTLIBDIR]# cat 2ndNONpersistenttestfile.txt
This file was added to the /var/lib/TESTLIBDIR directory with write enable signal NOT asserted.
This file can be read and added to but is not persistent up reboot.
```

Reboot XSR

```
[root@localhost TESTLIBDIR]# reboot
Connection to 192.168.100.100 closed by remote host.
Connection to 192.168.100.100 closed.
```

Log Back in to XSR

```
[root@XSR galleon]# ssh root@192.168.100.100
root@192.168.100.100's password:
[root@localhost ~]# cd /var/lib/TESTLIBDIR/
```

The contents of /var/TESTLIBDIR have returned to only files that are persistent.

```
[root@localhost TESTLIBDIR]# ls
testlibdirfile.txt
[root@localhost TESTLIBDIR]# cat testlibdirfile.txt
This text was wrote to the /var/lib/TESTLIBDIR/testlibdirfile.txt with the write enable signal asserted.
```