



Linux driver - Binary rpm/source rpm user guide and known limitation

TABLE OF CONTENTS

1. OVERVIEW.....	2
2. DRIVER COMPILATION STEPS	2
3. RELEASE CONTENTS:	2
4. ERRATA/NOTES AND KNOWN LIMITATIONS.....	3
5. INSTALL/REMOVE/UPGRADE DRIVER PACKAGE.....	4
6. STEPS TO GENERATE BINARY RPM FROM SOURCE RPM	5

1. OVERVIEW

This README covers Avago's MegaRAID Linux driver specific limitation and information. For any OS Distro specific limitation and information please check with OS Vendor support.

2. DRIVER COMPILATION STEPS

Driver source code is placed inside released driver package. Driver source tarball should be with name - megaraid_sas-<driver_version>-src.tar.gz

- 1) Untar driver source tarball-
#tar -zxvf megaraid_sas-<driver_version>-src.tar.gz
- 2) Go to driver source directory-
#cd megaraid_sas-<driver_version>/
- 3) To compile the driver for distro use the helper script "compile.sh" bundled inside source code-
#./compile.sh

3. RELEASE CONTENTS:

For any queries on supported OS matrix, please refer "OS_Support_list.txt" text file bundled with each release. OS Support list file list out test coverage executed by Avago. Megaraid_sas driver is GPLv2 open source driver and source code level support is possible for many linux kernel versions. If you do not find binary level support for your distribution in release contents, please use source rpm method. For any distribution or supported kernel version, there can be three possible packages -

- Driver update disk (Available under folder disks-xx)
- Precompiled binary (kmod/kmp/rpms).(Available under folder rpms-xx)
- Source rpm. (Available under folder rpms-xx)

4. ERRATA/NOTES AND KNOWN LIMITATIONS

a. Oracle Linux Installation errata:

Installing Driver during Installing for UEK from CD:

For UEK kernels, only the KMODs RPMs are provided, not the DUDs. The reasoning behind this is the OEL installation is using the native Red Hat kernels, not UEK. UEK kernel RPMs can be installed after the basic installation is completed. Please check with Oracle support team w.r.t UEK kernel installation process, limitation and other technical queries which is more of generic and not related to MR Drivers.

b. RPM install dependency issues:

If driver RPM installation fails with kABI checks dependency failure message, installing RPM package the user will need to use the "--nodeps" switch when installing the binary."

Example: `rpm -ivh --nodeps kmod-megaraid_sas-vxxxxxxx_UEK.xxx.rpm`

"If "rpm -ivh throw any dependency warning/error"

RPM uses KMOD packaging dependency data to ensure the dependencies are met before installing the binary RPM.

Red Hat maintains a whitelist of kernel symbols which RPM uses to validate against the KMOD binaries. Some symbols may be in the kernel but not on the whitelist which results in a failed binary RPM install. User can use the "--nodeps" switch when installing the binary to skip those whitelist symbol checks or any other dependency."

c. Xen Server Driver update note:

Known Issue:

In case of Citrix driver update disk use, where system has multiple MegaRAID controller(at least one MR controller which is supported by Inbox Citrix<megaraid_sas> driver),user will not see storage behind controller which is not supported by Inbox Driver (normally this was not expected since user expected driver update disk will work well and it will detect all controllers).

Reason for the issue:

Since Xen Server load megaraid_sas driver from inbox before it asks for Driver update disk, user can see "No Storage Disk" on some servers as explained above.

Workaround:

If you are booting from CD, when presented with the 'boot:' prompt you want to type 'shell', where you can then rmmmod your driver before continuing with the installation.

5. INSTALL/REMOVE/UPGRADE DRIVER PACKAGE

There are two packaging formats for binary rpms -

- 1) RPM(RHEL/SLES/Fedora/OEL uses RPM package)
- 2) DEB(Ubuntu/Debian uses DEB package) in which driver binary support is provided. More info -

https://en.wikipedia.org/wiki/Deb_file_format

Please note that after install/remove driver package, system needs to be rebooted to get intended driver loaded or manually remove module and insert (read man page "rmmod" and "modprobe" for more info)

Whenever driver package is installed/uninstalled/upgraded, check output of command `#modinfo megaraid_sas`.

Output should always have updated driver version.

Steps for Driver install/remove/upgrade for .rpm package

1. To install driver RPM, type below command-

```
# rpm -ivh <DRIVER_PACKAGE>.rpm
```

2. To uninstall driver RPM, type below command to check installed driver package name-

```
# rpm -qa | grep megaraid
```

Output will give installed megaraid_sas RPM packages.

Now type `# rpm -e <package to be uninstalled>`

3. To upgrade driver RPM, type below command-

```
#rpm -Uvh <DRIVER_PACKAGE>.rpm
```

Steps for Driver install/remove/upgrade for .deb package

1. To install .deb package, execute following command,

```
#dpkg -i <DRIVER_PACKAGE>.deb
```

2. To verify the status of installed packages then type following command

```
#dpkg -s megaraid-sas
```

3. After installing megaraid_sas driver, type below

```
#modinfo megaraid_sas
```

It should show the currently installed version of megaraid_sas

4. To use installed DEB driver loaded, restart the machine and type following command to get currently loaded driver version-

```
#cat /sys/modules/megaraid_sas/version
```

This version should be same as driver version of installed driver DEB package.

5. To uninstall megaraid_sas package, type below command-
`#dpkg -r megaraid-sas`
6. Verify "modinfo megaraid_sas" megaraid_sas version should be in-box version.

6. STEPS TO GENERATE BINARY RPM FROM SOURCE RPM

There are three variant of source rpms available in this package -

- a. Source rpm which use kmodtool interface (RHEL based)
<http://rpmfusion.org/Packaging/KernelModules/Kmods2>
- b. Source rpm which use kmp build interface (SLES based)
[https://en.opensuse.org/Kernel Module Packages](https://en.opensuse.org/Kernel_Module_Packages)
- c. Source rpm which use generic build interface (Create initramfs internally and does not depend upon any external tool)

If user doesn't know which source rpm is better for their environment, we recommend trying #c (for any other distro other than Redhat/Novell)

Quick search of "rpm -qa |grep kmod" can provide hint, if kmod tool support is available or not.

To generate binary rpm from source rpm, user should have compilation/build environment to create kernel module, utilities to build RPM(e.g. rpmbuild..). E.a system where user has yum repo configured, use below.

```
# yum groupinstall "Development Tools"
```

E.a on Ubuntu user can try installing below missing components.

```
apt-get install rpm
apt-get install make
apt-get install gcc
apt-get install alien
```

Exact commands to create build environment would be different across distros so this document is not right place to cover those details. Please refer OS vendor document if needed how to create build/compilation environment for specific OS distro.

Below are steps to generate binary RPM from source RPM-

1. Install source RPM like below-

```
#rpm -ivvh megaraid_sas-<driver_version>.src.rpm
```

Installing above RPM will copy driver SPEC file to specific location (configured as part of rpm package. This path can be different for each OS distribution.)

To locate SPEC file, check output logs of above source RPM installation (see blue marked gives SPEC file location).

e.g.

```
[root@localhost tmp]# rpm -ivvh megaraid_sas-06.810.00.02-98.src.rpm
D: ===== megaraid_sas-06.810.00.02-98.src.rpm
..
Updating / installing...
 1:megaraid_sas-06.810.00.02-
98      ##### [100%]
D: ===== Directories not explicitly included in package:
D:          0 /root/rpmbuild/SOURCES/
D:          1 /root/rpmbuild/SPECS/
D: =====
D: unknown    100755  1 (  0,  0)  25
/root/rpmbuild/SOURCES/Module.supported;55a756c8
D: unknown    100644  1 (  0,  0)120552
/root/rpmbuild/SOURCES/megaraid_sas-
06.810.00.02.tar.gz;55a756c8
D: unknown    100644  1 (  0,  0) 6783
/root/rpmbuild/SPECS/megaraid_sas.spec;55a756c8
GZDIO:      17 reads,   127888 total bytes in 0.000585 secs
D: closed    db index      /var/lib/rpm/Name
D: closed    db index      /var/lib/rpm/Packages
D: closed    db environment /var/lib/rpm
```

2. Go to directory where driver SPEC file is copied as part of #1. There must be SPEC file e.g. megaraid_sas.spec/lsi-megaraid_sas or megaraid_sas.spec(driver SPEC file name could be different for different distros, so check SPEC file with megaraid string in its name).

Check spec file name in above command. It provides the location and spec filename.

3. Build binary RPM from source RPM. Below is command to do same-
#rpmbuild -ba <DRIVER_SPEC_FILE>

4. Binary RPMs will be available if #3 exits without any error. Go to directory where new binary RPM is generated.

E.a Snippet of working case -

--

```
Wrote: /root/rpmbuild/SRPMS/megaraid_sas-06.810.00.02-98.src.rpm
Wrote: /root/rpmbuild/RPMS/x86_64/megaraid_sas-06.810.00.02-
98.x86_64.rpm
Wrote: /root/rpmbuild/RPMS/x86_64/megaraid_sas-debuginfo-06.810.00.02-
98.x86_64.rpm
Executing(%clean): /bin/sh -e /var/tmp/rpm-tmp.ZbHbmH
+ umask 022
+ cd /root/rpmbuild/BUILD
+ cd megaraid_sas-06.810.00.02
+ /usr/bin/rm -rf /root/rpmbuild/BUILDROOT/megaraid_sas-06.810.00.02-
98.x86_64
+ exit 0
```

--

Step #5 is applicable only for deb package based OS distribution (Ubuntu/Debian).

5. Create driver .deb package from binary RPM, Execute below command to achieve the same-
alien -k --to-deb --scripts <GENERATED_DRIVER_RPM>

Now, Install generated binary rpms using steps mentioned at
"Install/Remove/Upgrade driver binary package"

It is always recommended to verify initramfs image to confirm updated megaraid_sas driver is packaged correctly before system reboot. Do not assume that latest driver is loaded after installing rpm generated from source rpm. See OS distribution specific documentation on how to verify initramfs image. E.a FC21 user guide link for reference.

https://docs.fedoraproject.org/en-US/Fedora/21/html/System_Administrators_Guide/sec-Verifying_the_Initial_RAM_Disk_Image.html#bh-Verifying_the_Initial_RAM_Disk_Image_and_Kernel_on_IBM_eServer_System_i

Additional Note on compiling source rpm with retpoline support for Spectre V2 mitigation:

1. Driver RPMs with retpoline support can be built using source RPM method described above. Please refer the documentation from respective OS vendors on additional information on Spectre V2 mitigation with retpoline.

2. This would require updated GCC package with support for retpoline options. Refer the latest GCC version released by OSV that supports retpoline options: “-mindirect-branch” and “-mindirect-branch-register”.
3. The RPM SPEC file provided in the source RPM contains conditional build section that can be enabled by passing “--with retpoline” option to the rpmbuild command mentioned in step #3 above.
Eg: `#rpmbuild -ba --with retpoline <DRIVER_SPEC_FILE>`
4. Passing this flag through rpmbuild command will enable the below additional GCC flags required to compile the driver source with retpoline support:
KCFLAGS='-mindirect-branch=thunk-inline -mindirect-branch-register'
5. Driver module compiled this way will expose “retpoline: Y” in modinfo output.