

Open-iSCSI.org Software iSCSI Initiator Configuration & Multipathing Guide

Abstract

This document covers installing and configuring the open-iSCSI.org Software iSCSI initiator under debian or Ubuntu Linux and enabling the MPIO capability

Copyright © 2007 EqualLogic, Inc.

December 2007

EqualLogic is a registered trademark of EqualLogic, Inc.

All trademarks and registered trademarks mentioned herein are the property of their respective owners.


Possession, use, or copying of the documentation or the software described in this publication is authorized only under the license agreement specified herein.

EqualLogic, Inc. will not be held liable for technical or editorial errors or omissions contained herein. The information in this document is subject to change. Your performance can vary.

EqualLogic, Inc.
100 Spit Brook Road
Building ZK02
Nashua, NH 03063

Tel: 603.579.9762
Fax: 603.579.6910

Support: 877.887.7337



REVISION INFORMATION

The following table describes the release history of this document.

Technical Report Revision	Date	Change
1.0	12/2007	Initial Release

Table of Contents

Revision Information	3
Installation Notes:	5
Obtaining and unpacking the source code:	5
Compiling the source code	6
Discovering Targets:	9
Logging into Targets	10
Checking Session Status:.....	10
Logging off a target:.....	11
Persistent Device naming:	12
Create Filesystem on iSCSI volume:.....	15
Mounting iSCSI Filesystems at Boot:	15
Configuring Multitpath Connections:	16

INSTALLATION NOTES:

Notes:

- Don't use the debian open-iSCSI package that comes with debian etch. At the time of this document, that debian package was over a year old, and was v1.0 build 754. The build as of this writing is v2.0 build 865.15.
- This guide can be used with Ubuntu v7.x and debian etch.
- Minimum suggested kernel revision 2.6.18.
- You need 'flex', 'bison', 'gcc' and the kernel-headers for your kernel in order to compile the open-iSCSI initiator.
- **DISCLAIMER:** Currently EqualLogic hasn't certified the open-iSCSI initiator. We cannot guarantee its suitability. We suggest that you [subscribe](#) to the open-iscsi.org mailing list for technical support. Please contact your EQL sales rep or customer support for the latest information.

OBTAINING AND UNPACKING THE SOURCE CODE:

First download the tarball from the open-iSCSI website; open-iscsi-2.0-865.15.tar.gz.

```
# mkdir /source
# cd /source
# wget http://www.open-iscsi.org/bits/open-iscsi-2.0-865.15.tar.gz
--11:24:24--  http://www.open-iscsi.org/bits/open-iscsi-2.0-865.15.tar.gz
           => `open-iscsi-2.0-865.15.tar.gz'
Resolving www.open-iscsi.org... 209.210.238.66
Connecting to www.open-iscsi.org|209.210.238.66|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 230,678 (225K) [application/x-gzip]

100%[=====>] 230,678      109.50K/s

11:24:27 (109.35 KB/s) - `open-iscsi-2.0-865.15.tar.gz' saved [230678/230678]
```

Now unpack the archive.

```
#tar xvzf open-iscsi-2.0-865.15.tar.gz
open-iscsi-2.0-865.15/
open-iscsi-2.0-865.15/kernel/
open-iscsi-2.0-865.15/kernel/libiscsi.c
open-iscsi-2.0-865.15/kernel/scsi_transport_iscsi.c
open-iscsi-2.0-865.15/kernel/iscsi_tcp.h
open-iscsi-2.0-865.15/kernel/2.6.16-18_compat.patch
open-iscsi-2.0-865.15/kernel/2.6.20-21_compat.patch
open-iscsi-2.0-865.15/kernel/iscsi_tcp.c
....
```

COMPILING THE SOURCE CODE

Change to that directory

```
#cd <source dir>/open-iscsi-2.0-865.15
```

There are two edits that we suggest you do before building the initiator.

1. Enable auto connect at startup time. This means that on startup discovered volumes will be connected automatically. (Default is manual)
2. Add a 'killall' line to the startup script to ensure the iSCSI daemon is stopped.

Enable automatic connect in the source directory edit <source dir>/etc/iscsid.conf

```
#vi etc/iscsi.conf

#*****
# Startup settings
#*****

# To request that the iscsi initd scripts startup a session set to "automatic".
# node.startup = automatic
#
# To manually startup the session set to "manual". The default is manual.
node.startup = manual
```

- Uncomment 'node.startup = automatic'
- Comment out 'node.startup = manual'

So it will look like this when you're done.

```
#####  
# Startup settings  
#####  
  
# To request that the iscsi initd scripts startup a session set to "automatic".  
node.startup = automatic  
#  
# To manually startup the session set to "manual". The default is manual.  
#node.startup = manual
```

Save the file and exit.

Add the *'killall'* command to the startup script.

#vi etc/initd/initd.debian (this script is used for both Ubuntu and debian)

Find the section labeled 'stop()'

```
stop() {  
    stoptargets  
    log_daemon_msg "Stopping iSCSI initiator service"  
    start-stop-daemon --stop --quiet --pidfile $PIDFILE --exec $DAEMON  
    rm -f $PIDFILE  
    modprobe -r ib_iser 2>/dev/null  
    modprobe -r iscsi_tcp 2>/dev/null  
    log_end_msg 0  
}
```

Add a *'killall -1 iscsid'* after the *'rm -f \$PIDFILE'* line.

```
rm -f $PIDFILE  
killall -1 iscsid
```

Save the file and exit.

Now you can build the code.

1. Run `'make clean'`

```
# make clean
make -C utils clean
make[1]: Entering directory `/source/open-iscsi-2.0-865.15/utils'
rm -f *.o iscsi-iname
make[1]: Leaving directory `/source/open-iscsi-2.0-865.15/utils'
make -C usr clean
make[1]: Entering directory `/source/open-iscsi-2.0-865.15/usr'
rm -f *.o iscsid iscsiadm iscsistart
make[1]: Leaving directory `/source/open-iscsi-2.0-865.15/usr'
make -C kernel clean
make[1]: Entering directory `/source/open-iscsi-2.0-865.15/kernel'
make -C /lib/modules/2.6.22-14-generic/build M=`pwd` KBUILD_OUTPUT= V=0
clean
make[2]: Entering directory `/usr/src/linux-headers-2.6.22-14-generic'
make[2]: Leaving directory `/usr/src/linux-headers-2.6.22-14-generic'
rm -f Module.symvers
make[1]: Leaving directory `/source/open-iscsi-2.0-865.15/kernel'
make -C utils/fwparam_ibft clean
make[1]: Entering directory `/source/open-iscsi-2.0-865.15/utils/fwparam_ibft'
rm -f *.o fwparam_ibft
make[1]: Leaving directory `/source/open-iscsi-2.0-865.15/utils/fwparam_ibft'
```

2. Run `'make'` when complete you'll see the following.

```
Compilation complete           Output file
-----
Built iSCSI Open Interface module: kernel/scsi_transport_iscsi.ko
Built iSCSI library module:      kernel/libiscsi.ko
Built iSCSI over TCP kernel module: kernel/iscsi_tcp.ko
Built iSCSI daemon:              usr/iscsid
Built management application:     usr/iscsiadm
Built utility:                    utils/fwparam_ibft/fwparam_ibft

Read README file for detailed information.
```

3. Run `'make install'`

```
# /etc/init.d/open-iscsi start
```


Enable the iSCSI service to be started at boot time; the *update-rc.d* command can be used as follows:

```
# update-rc.d -f open-iscsi defaults
Adding system startup for /etc/init.d/open-iscsi ...
/etc/rc0.d/K20open-iscsi -> ../init.d/open-iscsi
/etc/rc1.d/K20open-iscsi -> ../init.d/open-iscsi
/etc/rc6.d/K20open-iscsi -> ../init.d/open-iscsi
/etc/rc2.d/S20open-iscsi -> ../init.d/open-iscsi
/etc/rc3.d/S20open-iscsi -> ../init.d/open-iscsi
/etc/rc4.d/S20open-iscsi -> ../init.d/open-iscsi
/etc/rc5.d/S20open-iscsi -> ../init.d/open-iscsi
```

You also need to do the same for the Multipath daemon.

```
# update-rc.d -f multipath-tools defaults
(If it's already set you'll see this message)
System startup links for /etc/init.d/multipath-tools already exist.
```

DISCOVERING TARGETS:

Once you have the iSCSI service running you will use the *'iscsiadm'* utility to discover, login and logout of targets.

To get a list of available targets type:

```
#iscsiadm -m discovery -t st -p <Group IP address>:3260
```

Example:

```
#iscsiadm -m discovery -t st -p 172.23.10.240:3260
```

```
172.23.10.240:3260,1 iqn.2001-05.com.equallogic:0-8a0906-83bcb3401-16e0002fd0a46f3d-open-iscsi-test
```

The example shows that the *'open-iscsi-test'* volume has been found. However, it's not yet logged in.

LOGGING INTO TARGETS

Here are two ways to connect to iSCSI targets:

(man iscsiadm for other methods)

- Log into all targets.

```
#iscsiadm -m node -l
```

- Log into an individual target.

```
#iscsiadm -m node -l -T <Complete Target Name> -l -p <Group IP address>:3260
```

Example:

```
#iscsiadm -m node -l -T iqn.2001-05.com.equallogic: 83bcb3401-16e0002fd0a46f3d-open-iscsi-test -p <Group IP address>:3260
```

CHECKING SESSION STATUS:

To see the connection status run:

```
#iscsiadm -m session -P 0
tcp: [1] 172.23.10.240:3260,1 iqn.2001-05.com.equallogic:6-8a0900-ffef60402-36c23acd6e745be0-dw-open-iscsi-vol0
tcp: [2] 172.23.10.240:3260,1 iqn.2001-05.com.equallogic:6-8a0900-ffef60402-36c23acd6e745be0-dw-open-iscsi-vol0
tcp: [3] 172.23.10.240:3260,1 iqn.2001-05.com.equallogic:6-8a0900-8d1cb3401-c2f0002decc46730-dw-open-iscsi-backup
tcp: [4] 172.23.10.240:3260,1 iqn.2001-05.com.equallogic:6-8a0900-8d1cb3401-c2f0002decc46730-dw-open-iscsi-backup
```

For a more verbose output change to '-P 1'

```
#iscsiadm -m session -P 1
```

```
Target: iqn.2001-05.com.equallogic:6-8a0900-ffef60402-36c23acd6e745be0-dw-  
open-iscsi-vol0
```

```
Current Portal: 172.23.10.246:3260,1
```

```
Persistent Portal: 172.23.10.240:3260,1
```

```
*****
```

```
Interface:
```

```
*****
```

```
Iface Name: iface.eth0
```

```
Iface Transport: tcp
```

```
Iface Initiatorname: iqn.2005-03.org.open-iscsi:2009610cb1b
```

```
Iface IPaddress: 172.23.49.170
```

```
Iface HWaddress: 00:04:23:C7:E1:5A
```

```
Iface Netdev: default
```

```
SID: 1
```

```
iSCSI Connection State: LOGGED IN
```

```
iSCSI Session State: LOGGED_IN
```

```
Internal iscsid Session State: NO CHANGE
```

```
Current Portal: 172.23.10.244:3260,1
```

```
Persistent Portal: 172.23.10.240:3260,1
```

```
*****
```

```
Interface:
```

```
*****
```

```
Iface Name: iface.eth1
```

```
Iface Transport: tcp
```

```
Iface Initiatorname: iqn.2005-03.org.open-iscsi:2009610cb1b
```

```
Iface IPaddress: 172.23.49.171
```

```
Iface HWaddress: 00:04:23:C7:E1:5B
```

```
Iface Netdev: default
```

```
SID: 2
```

```
iSCSI Connection State: LOGGED IN
```

```
iSCSI Session State: LOGGED_IN
```

```
Internal iscsid Session State: NO CHANGE
```

LOGGING OFF A TARGET:

It's the same as logging into a target, except, that you use `-u` instead of `-l`

```
#iscsiadm -m node -u -T iqn.2001-05.com.equallogic:0-8a0906-83bcb3401-  
16e0002fd0a46f3d-open-iscsi-test -p <Group IP address>:3260
```

PERSISTENT DEVICE NAMING:

Devices using the software initiators do not have a persistent naming scheme, and do not guarantee that a device (i.e. `/dev/sdc`) will always have the same device node. Persistent Naming describes mechanisms where the system identifies devices without relying on the `/dev` node, and provides a reference point for it that does not change at reboot.

First you have to comment out the 'Blacklist all devices' section in `/etc/multipath.conf` file

Note:

If the example file, `multipath.conf` is not in `/etc`, copy it from `/usr/share/doc/multipath-tools/multipath.conf.synthetic`

```
#cp /usr/share/doc/multipath-tools/multipath.conf.synthetic
/etc/multipath.conf
```

```
# Blacklist all devices by default. Remove this to enable multipathing
# on the default devices.
blacklist {
    devnode "*"
}
```

So it should look like this:

```
#blacklist {
#     devnode "*"
#}
```

Then restart the multipathd daemon

```
#/etc/init.d/multipathd restart
```

Now check that dev-mapper has configured the volume.

```
#multipath -v2
#multipath -ll
mpath0 (36090a01840b3bc833d6fa4d02f00e016) dm-2 EQLOGIC,100E-00
[size=8.0G][features=0][hwandler=0]
\_ round-robin 0 [prio=1][active]
\_ 2:0:0:0 sdb 8:16 [active][ready]
```

The highlighted number is the UUID of the volume. That never changes. You can use that UUID to create a persistent, friendlier name. For example you can name it the same as you called the volume on the EQL array.

Again edit the `/etc/multipath.conf` file.

Uncomment the following section and change the defaults to match your UUID and set a friendly alias name.

Edit the `/etc/multipath.conf` file and uncomment out the following:

```
#multipaths {
#     multipath {
#         wwid                3600508b4000156d700012000000b0000
#         alias                yellow
#         path_grouping_policy multibus
#         path_checker         readsector0
#         path_selector        "round-robin 0"
#         failback             manual
#         rr_weight             priorities
#         no_path_retry        5
#         rr_min_io            100
#     }
#     multipath {
#         wwid                1DEC_____321816758474
#         alias                red
#     }
# }
```

Change the number after 'wwid' to the UUID for your volume. Change the 'alias' to something more friendly or use the volume name from the array. Change the 'rr_min_io' to 10.

Here's an example showing how to create persistent names for more than one volume.

```
multipaths {
    multipath {
        wwid                36090a02830f251891f74744263735281
        alias                open-iscsi-test
        path_grouping_policy multibus
        path_checker         readsector0
        path_selector        "round-robin 0"
        failback             manual
        rr_weight             priorities
        no_path_retry        5
        rr_min_io            10
    }
    multipath {
        wwid                36090a01840b31c74e173a4873200a02f
        alias                svr-vol
    }
}
```

Save the file, then run:

```
#multipath -v2
#multipath -ll
open-iscsi-test (36090a02830f251891f74744263735281) dm-1 EQLOGIC,100E-00
[size=100G][features=1 queue_if_no_path][hw_handler=0]
\_ round-robin 0 [prio=0][active]
  \_ 9:0:0:0 sdc 8:48 [active][ready]
svr-vol (36090a01840b31c74e173a4873200a02f) dm-0 EQLOGIC,100E-00
[size=10G][features=0][hw_handler=0]
\_ round-robin 0 [prio=0][enabled]
  \_ 6:0:0:0 sdb 8:16 [active][ready]
```

```
#ls -l /dev/mapper
```

```
total 0
crw-rw---- 1 root root 10, 63 2007-11-16 17:15 control
brw-rw---- 1 root disk 254, 1 2007-11-19 15:59 open-iscsi-test
brw-rw---- 1 root disk 254, 0 2007-11-19 15:58 svr-vol
```

You now have persistent names to access those volumes.

CREATE FILESYSTEM ON ISCSI VOLUME:

Now create a filesystem on that device.

Example:

```
#mke2fs -j -v /dev/mapper/open-iscsi-test
```

This creates an EXT3 filesystem on that device.

Any filesystem that your Linux OS supports is available for iSCSI volumes.

Disk virtualization products like Logical Volume Manager (LVM), may require modification to the system start up scripts, since the iSCSI service starts after LVM. This means LVM'd iSCSI volumes won't be activated automatically.

MOUNTING ISCSI FILESYSTEMS AT BOOT:

In order to mount a filesystem that exists on an iSCSI Volume connected through the open-iSCSI Software initiator, you need to add a line to the `/etc/fstab` file. The format of this line is the same as any other device and filesystem with the exception being that you need to specify the `_netdev` mount option, and you want to have the last two numbers set to 0 (first is a dump parameter and the second is the fsck pass).

The `_netdev` option delays the mounting of the filesystem on the device listed until after the network has been started and also ensures that the filesystem is unmounted before stopping the network subsystem at shutdown.

An example of an `/etc/fstab` line for a filesystem to be mounted at boot that exists on an iSCSI Volume is as follows:

```
/dev/mapper/open-iscsi-test /mnt/open-iscsi-test ext3 _netdev,defaults 0 0
```

CONFIGURING MULTIPATH CONNECTIONS:

To create the multiple logins needed for Linux dev-mapper to work you need to create an *'interface'* file for each GbE interface you wish to use to connect to the array.

You have to move (rename) the *'iface.example'* to *'iface.ethX'* file in */etc/iscsi/ifaces*.

Example: `#mv /etc/iscsi/ifaces/iface.example /etc/iscsi/ifaces/iface.eth0`

You then edit the file and add in the MAC address of the ETH0 adapter.

You can get the MAC address with the following command:

```
#ifconfig eth0 | grep HW
```

```
eth0  Link encap:Ethernet HWaddr 00:50:56:97:70:70
```

The MAC address in this case is: 00:50:56:97:70:70

To bind by hardware address put the NIC's MAC address after *iface.hwaddress=*

Example:

```
iface.hwaddress = 00:50:56:97:70:70
```

Here's an example of what the */etc/iscsi/ifaces/iface.eth0* looks like:

```
# cat /etc/iscsi/ifaces/iface.eth0

#
# Example iSCSI interface config
#
# There must be a separate iscsi interface config file for each NIC, network
# interface or port or iscsi HBA you want to bind sessions to.
#
# For hardware/offload iscsi, this is created for you when you run iscsiadm.
# For software iscsi, you must define these files yourself.
#

# Set the iscsi transport/driver to use for the iface by setting
# iface.transport_name
# example:
# iface.transport_name = tcp

# This values is required and valid values for iface.transport_name are:
# - tcp (Software iSCSI over TCP/IP)
# - iser (Software iSCSI over infiniband)
# - qla4xxx (Qlogic QLA4XXX HBAs)

# __One__ of the following values are required for the binding.
#
# To bind by network interface name (example: eth0, eth2:2, eth1.3)
# set iface.net_ifacename
```



```
# example:
# iface.net_ifacename = eth0

# To bind by hardware address set the NIC's MAC address to iface.hwaddress
# example:
iface.hwaddress = 00:50:56:97:70:70

# Note you can only bind using one value. If you set multiple values
# the behavior is not defined.
```

Repeat for the other ports you wish to use. E.g. if you are going to use ETH1, then copy the /etc/iscsi/ifaces/iface.eth0 file to /etc/iscsi/ifaces/iface.eth1.

Then edit the MAC address to reflect the MAC address of the ETH1 adapter.

If you have already discovered your volumes, you now need to re-discover the target(s).

```
#iscsiadm -m discovery -t st -p <Group IP Addr>:3260
```

```
172.23.10.90:3260,1 iqn.2001-05.com.equallogic: 0-8a0906-83bcb3401-
16e0002fd0a46f3d-open-iscsi-test
172.23.10.90:3260,1 iqn.2001-05.com.equallogic: 0-8a0906-83bcb3401-
16e0002fd0a46f3d-open-iscsi-test
```

You should see an entry for each interface you specified.

You'll need to log into the volume.

```
#iscsiadm -m node -l -T iqn.2001-05.com.equallogic:0-8a0906-8951f2302-815273634274741f-open-iscsi-test -p <Group IP Addr>:3260
```

```
#iscsiadm -m session
```

```
tcp: [3] 172.23.10.90:3260,1 iqn.2001-05.com.equallogic: 0-8a0906-83bcb3401-16e0002fd0a46f3d-open-iscsi-test
```

```
tcp: [4] 172.23.10.90:3260,1 iqn.2001-05.com.equallogic: 0-8a0906-83bcb3401-16e0002fd0a46f3d-open-iscsi-test
```

This shows that both adapters have connected to the array.

To verify that the multipathing is correctly configured type:

```
#multipath -v2
```

```
#multipath -ll
```

```
dw-test-vol (36090a02830f251891f74744263735281) dm-1 EQLOGIC,100E-00
[size=100G][features=1 queue_if_no_path][hw_handler=0]
\_ round-robin 0 [prio=0][active]
  \_ 9:0:0:0 sdd 8:48 [active][ready]
  \_ 8:0:0:0 sde 8:64 [active][ready]
dw-svr-vol (36090a01840b31c74e173a4873200a02f) dm-0 EQLOGIC,100E-00
[size=10G][features=0][hw_handler=0]
\_ round-robin 0 [prio=0][enabled]
  \_ 6:0:0:0 sdb 8:16 [active][ready]
  \_ 7:0:0:0 sdc 8:32 [active][ready]
```

In this example you see that there are two paths to each volume.