

## SECTION 9 iSCSI

The Internet Small Computer Systems Interface (iSCSI) is an implementation of the regular SCSI commands over IP. SUSE Linux Enterprise Server 10 allows to connect to an iSCSI target as an iSCSI initiator, or to offer storage space to other iSCSI initiators as an iSCSI target.

### Objectives

1. [iSCSI Background](#)
2. [iSCSI Configuration](#)

## Objective 1 iSCSI Background

SCSI commands are used to enable systems to communicate with I/O devices, especially storage devices. The Internet Small Computer Systems Interface implements the regular SCSI commands over IP.

With iSCSI, the operating system (iSCSI initiator) does not send the SCSI command directly to a device, but to another system connected via the network (iSCSI target); this system will then redirect the SCSI commands to the actual storage device—for instance a partition, a logical volume or a file.

This is basically a client/server connection between iSCSI initiator (client) and iSCSI target (server). However, unlike usual client/server connections, usually only one client may connect to the server. The iSCSI protocol is standardized in RFC3720.

With iSCSI it is possible to connect to remote storage, which can be a disk, a logical volume or a file, and it will appear to the system as a local disk. This is also known as SAN. The advantage of iSCSI in comparison to other SAN solutions is primarily in the area of cost. Common SAN solutions use rather expensive FC (fibre channel) hardware, whereas iSCSI can use the already existing ethernet network. Furthermore, many SAN solutions also provide an iSCSI interface.

Considering the amount of data transferred over the network it is important that enough bandwidth is available, for instance Gigabit Ethernet. It often makes sense to set up a dedicated network for the iSCSI connections.

The remote devices are available on Linux as a regular SCSI (/dev/sdX) device and should be mounted with the option `_netdev`. You can configure the iSCSI targets to require a username and password upon connection from an iSCSI initiator.

There is more than one iSCSI solution available on Linux, but all of them use by default port 3260 for the iSCSI connection.

## Objective 2    iSCSI Configuration

The configuration and maintenance of iSCSI on RHEL4 and SUSE Linux Enterprise Server 10 differ in some aspects.

- [Red Hat Enterprise Linux 4](#)
- [SUSE Linux Enterprise Server 10](#)

### ***Red Hat Enterprise Linux 4***

With RHEL, iSCSI is available since Service Pack 2. It is implemented by two modules, `iscsi_sfnet` and `scsi_transport_iscsi`, and the package `iscsi-initiator-utils`.

To see if any iSCSI devices are currently used, use the command **`iscsi-ls -l`**.

If there are any devices listed, then you should analyze and save the configuration files used for the iSCSI connection: **`/etc/ietd.conf`**, **`/etc/iscsi.conf`** and **`/etc/initiatorname.iscsi`**. Their role will be described in detail later in this section.

Before you migrate to SUSE Linux Enterprise Server 10, you should stop the iSCSI daemon with the command **`service iscsi stop`**.

You should also check `/etc/fstab` for any configured iSCSI devices.

## SUSE Linux Enterprise Server 10

SUSE Linux Enterprise Server 10 can be configured as iSCSI target and as iSCSI initiator. You can do this with the configuration tool YaST or manually.

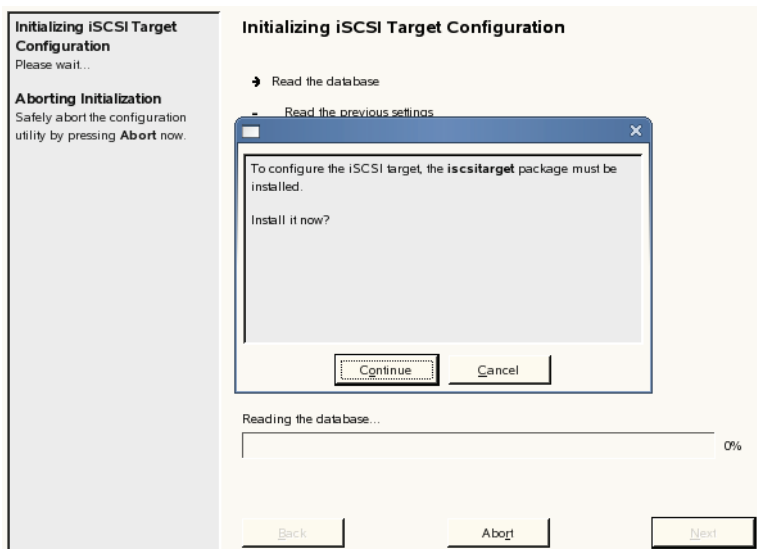
- [Set up an iSCSI Target](#)
- [Set up an iSCSI Initiator](#)
- [Mount iSCSI Targets Automatically at Boot Time](#)

### Set up an iSCSI Target

You can either start YaST and select **Network Services > iSCSI Target**, or start the iSCSI Target module directly by entering as root in a console window **yast2 iscsi-server**.

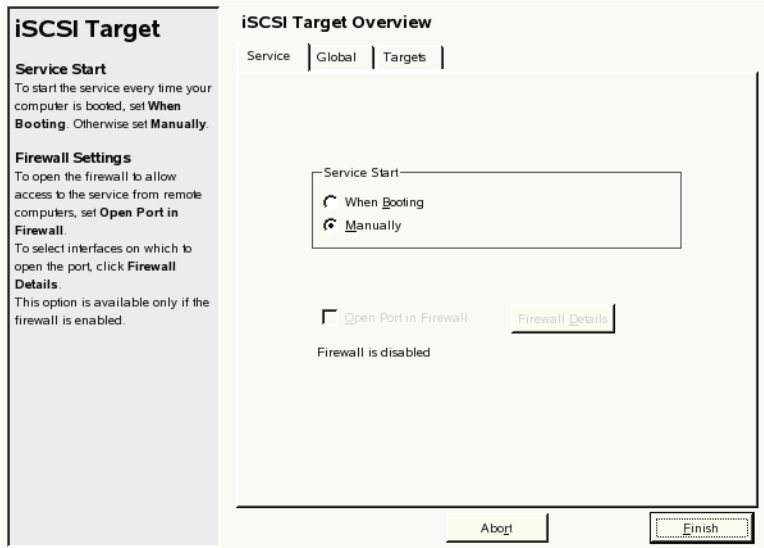
If the iscsitarget package is not installed, YaST will invite you to do that:

Figure 9-1



In the main menu of the iSCSI Target configuration there are 3 tabs available. Service is the first one; here you can configure how iSCSI will be started. Most likely **When Booting** is the right choice for you.

Figure 9-2



If your SuSE Firewall is active, you should select **Open Port in Firewall** to be able to access the port needed for iSCSI. If the Firewall is not activated, the option cannot be selected, as visible in the above screenshot.

On the **Global** register you can configure the authentication between iSCSI target and initiator during the discovery process.

Figure 9-3

**iSCSI Target**

Select the type of authentication. Use **No Authentication** or one of **Incoming** and **Outgoing** (can be both together). Then insert **User** and **Password**. For incoming authentication, it is possible to **Add** more pairs and **Edit** and **Delete** them.

**iSCSI Target Overview**

Service | Global | Targets

No Authentication

Incoming Authentication

Username Password

Add Edit Delete

Outgoing Authentication

Username Password

Abort Finish

By default **No Authentication** is selected. That means that every iSCSI client can discover this target.

The difference between **Incoming** and **Outgoing Authentication** is as follows:

Incoming Authentication signifies that the iSCSI initiator is responsible for the authentication. If you use **Outgoing Authentication**, then the iSCSI target is responsible.

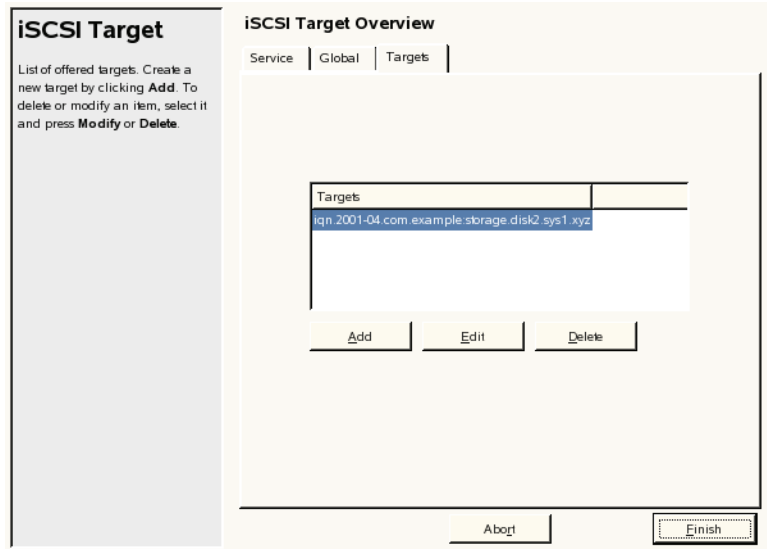
Challenge-Handshake Authentication Protocol (CHAP) is used for authentication.

The passwords defined here will be stored in the configuration files as clear text.

You can also set up the iSCSI environment without authentication (default).

On the third register you can configure your iSCSI targets:

**Figure 9-4**



By default there is already an iSCSI target defined, but because there is no backend configured, it is not directly usable. You could edit this target, but you can also delete it and create a new one.

Before you add an iSCSI target you should decide on the backend device you want to use. As backend device you can use a regular disk, a logical volume or a file. To demonstrate the configuration we will use a file with a size of 1 GB, which can be created with the following commands:

```
mkdir /srv/iscsi
```

```
dd if=/dev/zero of=/srv/iscsi/first-device bs=1M count=1024
```

Selecting the **Add** button opens the following dialog:

**Figure 9-5**

**iSCSI Target**

Create a new target. Replace template values with the correct values. For **Target**, use the format iqn.yyyy-mm.. For **Path**, use block devices, regular files, LVM, or RAID.

**Add iSCSI Target**

Target	Identifier
iqn.2006-10.com.example	9d60efb1-7a9e-481a-95b5-68da4ef1b6e9
LUN	Path
0	/tmp/file

Abort Next

In this dialog you can define the name of the iSCSI device and the backend to use. To use more LUNs (Logical Unit Numbers) on one iSCSI target you have to edit the file `/etc/ietd.conf` manually.

The target name should be unique world wide. By default, the following syntax is used:

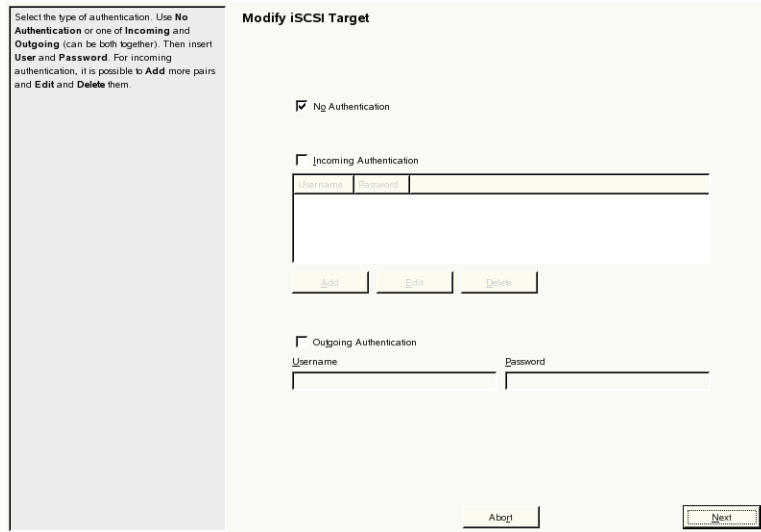
***iqn.YYYY-MM.reverse-domain-name: identifier***

**iqn** (iSCSI qualified name) is normally followed by the date of device creation and the domain name of the company in reverse order. The identifier is optional and can be used for a name based identification, like for example “iscsi-database-target”.

In the **path** box you define the backend device; in our case this would be `/srv/iscsi/first-device`.

Selecting **Next** opens the next dialog where you can configure the authentication for this target.

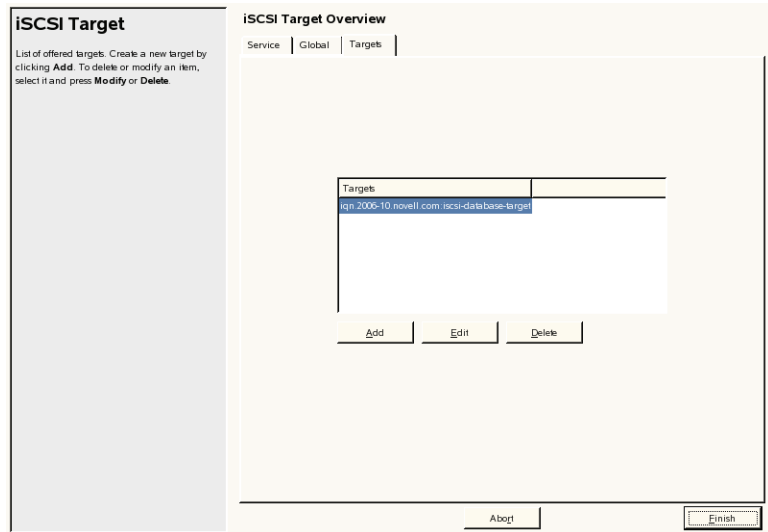
**Figure 9-6**



By default there is no authentication used. If you add a user and password for **Incoming Authentication**, you have to remember user and password for the iSCSI initiator setup.

After selecting **Next** you can view your new iSCSI target:

**Figure 9-7**



After selecting **Finish** the settings are written to the system.

The command **rciscsitarget status** (or **/etc/init.d/iscsitarget status**) displays whether the iSCSI target daemon is running or not. If it is, **cat /proc/net/iet/volume** will display your configured iSCSI target.

The iSCSI Target configuration is stored in **/etc/ietd.conf** and is managed by the service **iscsitarget-daemon** (**ietd**: iSCSI Enterprise Target Daemon).

The following listing shows an `/etc/ietd.conf` with two targets; one uses the device `/dev/sda6` as physical storage, the other one a file. The comments explain further parameters you might want to define for each target:

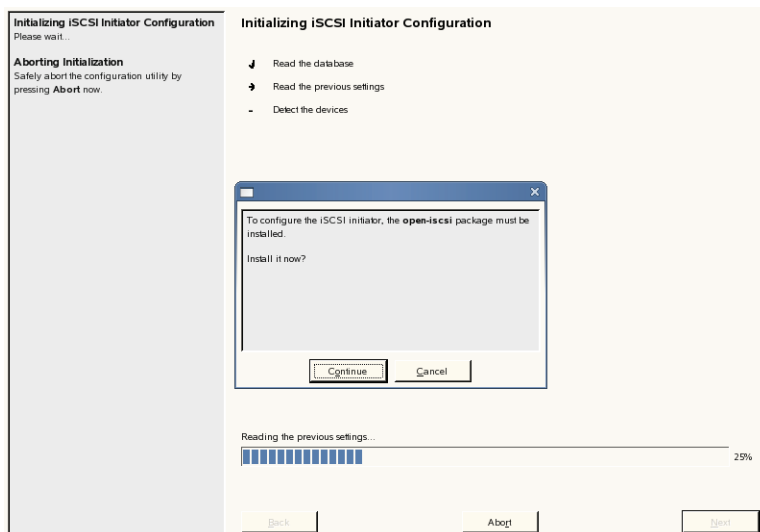
```
Target iqn.2006-10.com.digitalairlines:878137d1-d58f ...
Lun 0 Path=/dev/sda6,Type=fileio
Target iqn.2006-10.com.digitalairlines:14736b2d-f00e ...
Lun 1 Path=/srv/iscsi/second-device,Type=fileio
    # Users, who can access this target. The same rules
    # as for discovery users apply here. Leave them
    # alone if you don't want to use authentication.
    #IncomingUser joe secret
    #OutgoingUser jim 12charpasswd
    # Logical Unit definition
    # You must define one logical unit at least.
    # Block devices, regular files, LVM, and RAID can
    # be offered to the initiators as a block device.
    #Lun 0 Path=/dev/sdc,Type=fileio
    # Alias name for this target
    # Alias Test
    # various iSCSI parameters
    # (not all are used right now, see also iSCSI spec
    # for details)
    #MaxConnections          1
    #InitialR2T              Yes
    #ImmediateData          No
    #MaxRecvDataSegmentLength 8192
    #MaxXmitDataSegmentLength 8192
    #MaxBurstLength         262144
    #FirstBurstLength       65536
    #DefaultTime2Wait       2
    #DefaultTime2Retain    20
    #MaxOutstandingR2T     8
    #DataPDUInOrder        Yes
    #DataSequenceInOrder   Yes
    #ErrorRecoveryLevel    0
    #HeaderDigest           CRC32C,None
    #DataDigest             CRC32C,None
    # various target parameters
    #Wthreads                8
```

## Set up an iSCSI Initiator

You can either start YaST and select **Network Services > iSCSI Initiator**, or start the iSCSI Initiator module directly by entering as root in a console window **yast2 iscsi-client**.

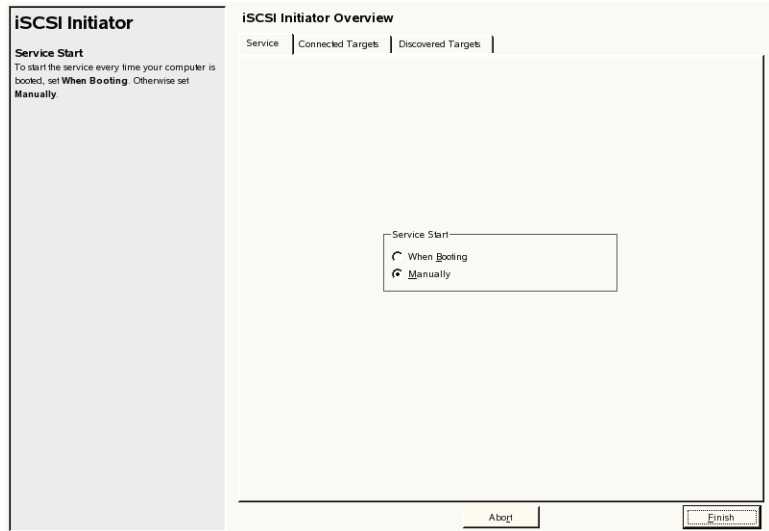
If the open-iscsi package is not installed, YaST will invite you to do that.

Figure 9-8



After the installation, the following dialog opens up:

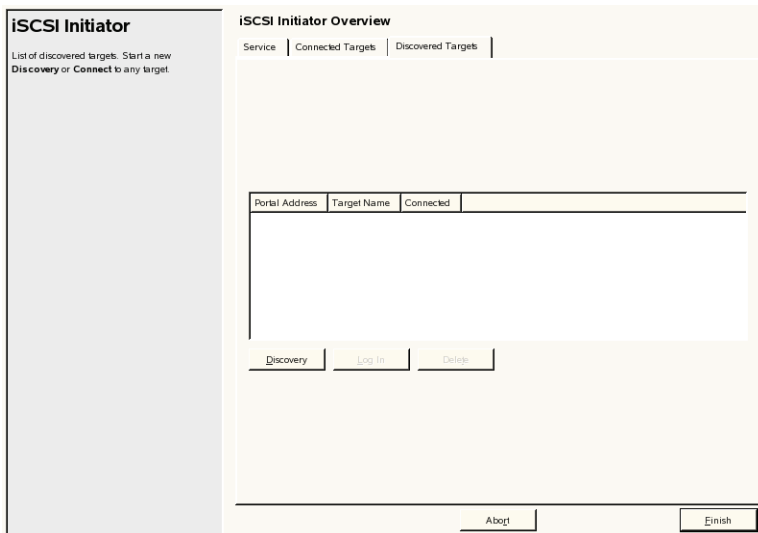
**Figure 9-9**



In the **Service** tab you can configure whether the iSCSI client should start automatically at boot time or manually.

The **Connected Targets** tab shows the active connections. But first you have to discover and configure an iSCSI target device. To do this, select the **Discovered Targets** tab:

**Figure 9-10**



Currently there are no target devices discovered. With the **Discovery** button you can start a search on a specific IP address.

**Figure 9-11**

**iSCSI Initiator**

Enter the **IP Address** of the discovered server. Only change **Port** if needed. For authentication, use **Username** and **Password**. If you do not need authentication, select **No Authentication**.

**Warning**

When accessing an iSCSI device **READ/WRITE**, make sure that this access is exclusive. Otherwise there is a potential risk of data corruption.

**iSCSI Initiator Discovery**

IP Address: \_\_\_\_\_ Port: 3200

Ng Authentication

Incoming Authentication

Username: \_\_\_\_\_ Password: \_\_\_\_\_

Outgoing Authentication

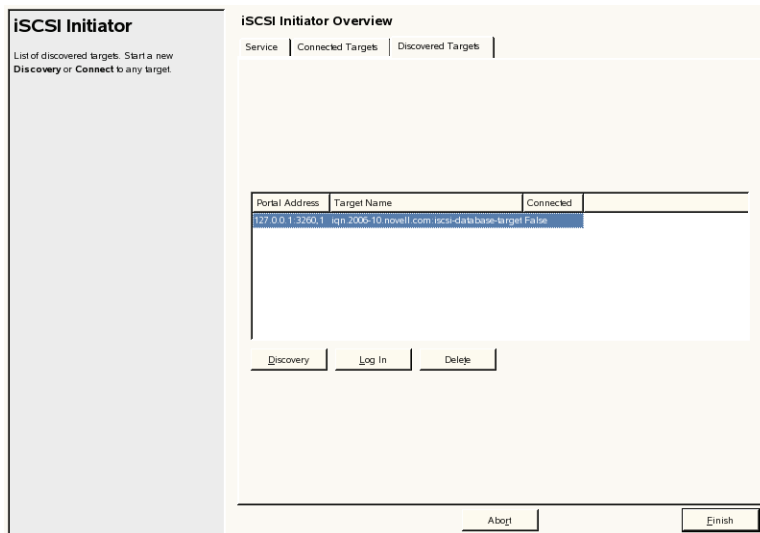
Username: \_\_\_\_\_ Password: \_\_\_\_\_

About Next

Enter the IP address you want to scan for iSCSI target devices. If username and password are needed for the discovery, enter them here. By default you don't have to use any authentication.

You can scan your own system by entering the IP address 127.0.0.1. If an iSCSI target device is available on the tested system, a dialog similar to the following opens up:

**Figure 9-12**

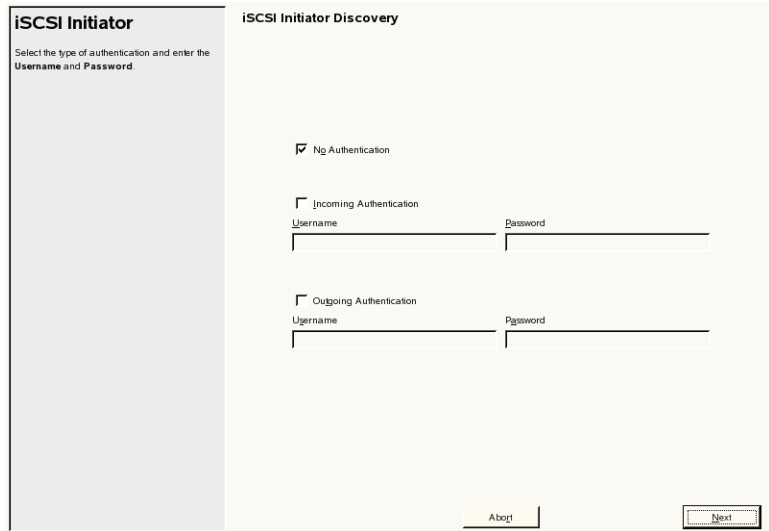


If there are several iSCSI target devices available, they will all get listed. The iqn (iSCSI Qualified Name) is used to identify the target.

After selecting the iSCSI target from the list you have to log in using the **Log In** button.

The following dialog appears:

**Figure 9-13**



By default there is no authentication necessary. However, if you have, for example, configured your iSCSI target device with incoming authentication, then you have to select **Outgoing Authentication** in this dialog and enter the same username and password which you used for the iSCSI target configuration.

After selecting **Next**, the **Connected Targets** tab displays your configured iSCSI devices.

**Figure 9-14**



With the **Toggle Start-Up** button you can switch between **automatic** and **manual**, which defines how the device is activated.

Select **Finish** to write the settings to the system.

The client configuration is stored in the file `/etc/iscsi.conf` and within the directory `/var/lib/open-iscsi/` in the database files `discovery.db` and `node.db`. The tool to use for manual configuration of these files is **iscsiadm**.

The name of the iSCSI initiator is contained in the file  
`/etc/initiatorname.iscsi`:

```
da3:~ # cat /etc/initiatorname.iscsi
## DO NOT EDIT OR REMOVE THIS FILE!
## If you remove this file, the iSCSI daemon will not start.
## If you change the InitiatorName, existing access control lists
## may reject this initiator. The InitiatorName must be unique
## for each iSCSI initiator. Do NOT duplicate iSCSI InitiatorNames.
InitiatorName=iqn.1996-04.de.suse:01.427f8cb385f5
```

A description of how `iscsid`, `iscsiadm` and the configuration files  
play together is contained in  
`/usr/share/doc/packages/open-iscsi/README`.

In the file `/var/log/messages` you should see entries similar to the  
following:

```
Oct  4 11:55:56 da3 kernel: scsi3 : iSCSI Initiator over TCP/IP, v1.0-595
Oct  4 11:55:56 da3 kernel: sess_param(173) 1 1 1 8192 8192 262144 65536 2
0 1 1 1 0 1 1
Oct  4 11:55:56 da3 kernel:   Vendor:   IET           Model: VIRTUAL-DISK
Rev: 0
Oct  4 11:55:56 da3 kernel:   Type:     Direct-Access
ANSI SCSI revision: 04
Oct  4 11:55:56 da3 kernel: SCSI device sdb: 2097152 512-byte hdwr sectors
(1074 MB)
Oct  4 11:55:56 da3 kernel: sdb: Write Protect is off
Oct  4 11:55:56 da3 kernel: sdb: Mode Sense: 77 00 00 08
Oct  4 11:55:56 da3 kernel: SCSI device sdb: drive cache: write through
Oct  4 11:55:56 da3 kernel: SCSI device sdb: 2097152 512-byte hdwr sectors
(1074 MB)
Oct  4 11:55:56 da3 kernel: sdb: Write Protect is off
Oct  4 11:55:56 da3 kernel: sdb: Mode Sense: 77 00 00 08
Oct  4 11:55:56 da3 kernel: SCSI device sdb: drive cache: write through
Oct  4 11:55:56 da3 kernel:   sdb: unknown partition table
Oct  4 11:55:56 da3 kernel: sd 3:0:0:0: Attached scsi disk sdb
Oct  4 11:55:56 da3 kernel: sd 3:0:0:0: Attached scsi generic sg2 type 0
Oct  4 11:55:56 da3 iscsid: version 1.0-604
Oct  4 11:55:56 da3 iscsid: iSCSI daemon with pid=6172 started!
Oct  4 11:55:56 da3 iscsid: iSCSI sync pid=6173 started
Oct  4 11:55:56 da3 iscsid: connection0:0 is operational now
```

You can see that there is a new SCSI device available (`/dev/sdb`). You can create partitions using `fdisk`, create file systems on the partitions and mount them.

With the command **lsscsi** you can see all available SCSI devices.

In case you want to see whether any active iSCSI connection exists, have a look into the file `/proc/net/iet/session` on the server hosting the target using **cat**. You can see all iSCSI connections to a target on this host, as in the following example:

```
da3:~ # cat /proc/net/iet/session
tid:1
name:iqn.2006-10.com.digitalairlines:d3d43303-8b2c-43df-83ea-6721942677e2
sid:281474980708864 initiator:iqn.1996-04.de.suse:01.427f8cb385f5
cid:0 ip:127.0.0.1 state:active hd:none dd:none
```

You can see that a connection from the IP address 127.0.0.1 is established to the Target with the id 1 (tid).

While it is possible to have more than one connection you should keep in mind that it is not recommended to mount an iSCSI target more than once without any precautions (like a cluster file system on top of the iSCSI target).

## Mount iSCSI Targets Automatically at Boot Time

On the server (iSCSI target) side you have to make sure that the `iscsi target` service will be started automatically after a reboot. Use the command **chkconfig iscsitarget on** (or **insserv iscsitarget**) to do this.

On the client (iSCSI initiator) side the `open-iscsi` daemon has to start automatically at boot time. This is achieved using the command **chkconfig open-iscsi on**.

If you checked the button **When booting** in the respective YaST dialog, it is not necessary to issue the above `chkconfig` commands.

Generally you enter all the devices which should be mounted automatically at boot time into the file `/etc/fstab`. One problem you face with iSCSI devices is that no network is available when the file `/etc/fstab` is read during the boot process. Another problem is that you do not know for sure which local SCSI device will be used for the iSCSI target.

There is a solution for both problems. To be sure that the iSCSI device will be mounted to the same directory every time you have to use the UUID (Unique Universal ID) from the partition. To establish the UUID of a device use the following command:

```
da3:~ # udevinfo -q symlink -n /dev/sdb1
disk/by-id/scsi-149455400000000000000000001000005d9a01000
2008000-part1
disk/by-path/ip-127.0.0.1:3260-iscsi-ign.2006-10.com.digit
alairlines:d3d43303-8b2c-43df-83ea-6721942677e2-part1
disk/by-uuid/8b4dcb80-372d-44ba-a35b-96b5619b8237
```

In this example, the last line contains the UUID of the device `/dev/sdb1`. Now you can create an entry like the following in `/etc/fstab` to mount this iSCSI device to the `/iscsi` directory:

```
/dev/disk/by-uuid/8b4dcb80-372d-44ba-a35b-96b5619b8237  
/iscsi auto hotplug,defaults 0 0
```

With the mount option **hotplug** the first problem is also solved. This option signifies that the hotplug daemon, which is started after the network, will take care of this mount.

**Exercise 9-1    *Set up an iSCSI Target and an iSCSI initiator***

In this exercise, you learn how to set up an iSCSI target and how to access that target using an iSCSI initiator.

You will find this exercise in the workbook.

***(End of Exercise)***

## Summary

Objective	Summary
1. <a href="#">iSCSI Background</a>	<p>The Internet Small Computer Systems Interface implements the regular SCSI commands over IP.</p> <p>The server offering storage is called iSCSI target, the client using that storage is called iSCSI initiator.</p>
2. <a href="#">iSCSI Configuration</a>	<p>iSCSI target and initiator can easily be configured using the respective YaST modules.</p> <p>The iSCSI Target configuration is stored in <code>/etc/ietd.conf</code>.</p> <p>The iSCSI initiator configuration is stored in <code>/etc/iscsi.conf</code>, <code>/etc/initiatorname.iscsi</code>, and in the files <code>discovery.db</code> and <code>node.db</code> in <code>/var/lib/open-iscsi</code>.</p>