



VMware ESX Server

Using Raw Device Mapping

This technical note is intended to help ESX Server administrators to understand how raw device mapping works and decide when it's appropriate to use it.

This document contains the following topics:

[What Is Raw Device Mapping? on page 2](#)

[Should I Use Raw Device Mapping? on page 4](#)

[Benefits of Raw Device Mapping on page 4](#)

[Limitations of Raw Device Mapping on page 6](#)

[More About Raw Device Mapping on page 7](#)

[Virtual Compatibility Mode Versus Physical Compatibility Mode on page 8](#)

[Dynamic Name Resolution on page 9](#)

[Raw Device Mapping with Virtual Machine Clusters on page 11](#)

[Comparing Raw Device Mapping to Other Means of SCSI Device Access on page 12](#)

[Managing Raw Device Mappings on page 13](#)

[Utilities on page 13](#)

[Sample Operations on page 14](#)



What Is Raw Device Mapping?

Introduced with ESX Server 2.5, raw device mapping allows a special file in a VMFS volume to act as a proxy for a raw device. The mapping file contains metadata used to manage and redirect disk accesses to the physical device. The mapping file gives you some of the advantages of a virtual disk in the VMFS file system, while keeping some advantages of direct access to physical device characteristics. In effect, it merges VMFS manageability with raw device access.

In prior releases, ESX Server has supported using raw disks as storage for virtual machines in addition to virtual disk files stored in a VMFS volume. While VMFS is recommended for most virtual disk storage, there is sometimes a need for raw disks. The most common use is as data drives for Microsoft Cluster Services (MSCS) configurations using clusters between physical machines, or between physical and virtual machines.

With ESX Server 2.5, VMware is encouraging the use of raw device mapping in the following situations:

- When SAN snapshot or other layered applications are run in the virtual machine. Raw device mapping better enables scalable backup offloading systems using the features inherent to the SAN.
- In any MSCS clustering scenario — virtual to virtual clusters as well as physical to virtual clusters. VMware now recommends that cluster data and quorum disks be configured as raw device mappings rather than as files on shared VMFS.

Until now, raw disks were difficult to manage for systems administrators. ESX Server 2.5 changes this with raw device mappings. A raw device mapping is effectively a symbolic link from a VMFS to a raw LUN. This makes LUNs appear as files in a VMFS volume. The mapping file — not the raw LUN — is referenced in the virtual machine configuration. The mapping file contains a reference to the raw LUN. Some important benefits of raw device mappings are:

- VMotion can now be used to migrate virtual machines using raw LUNs
- The process of adding raw LUNs to virtual machines is now simplified, both in the VMware Management Interface and in the VMware VirtualCenter client.
- Distributed file system features such as file locking, permissions and naming become available.

There are two compatibility modes for raw device mappings, virtual and physical. Virtual compatibility mode allows a mapping to act exactly like a virtual disk file, including the use of redo logs. Physical compatibility mode allows direct access to the SCSI device, for those applications needing lower level control.

With raw device mappings, VMware now has the most flexible storage capability for virtual machines in the industry.

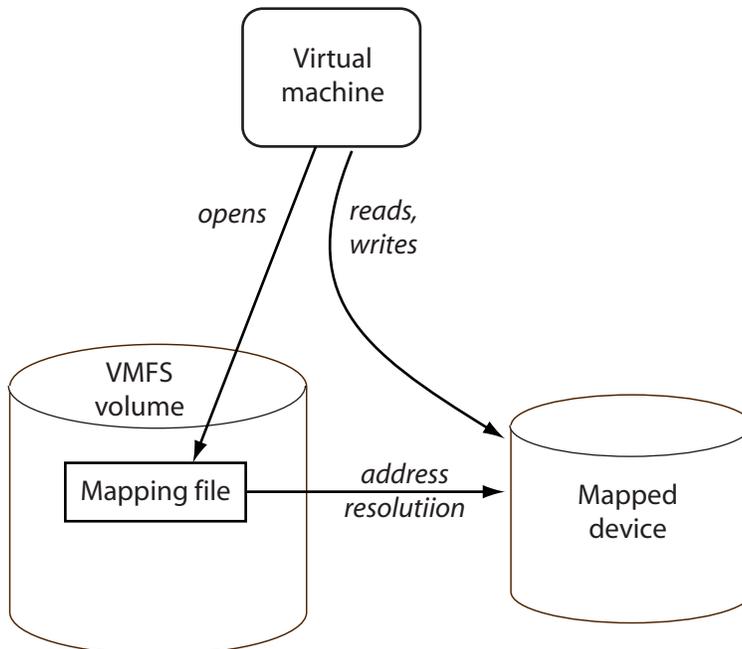


Terminology

In other contexts, such as the VirtualCenter client console, raw device mapping may be described in terms such as “Mapping a VMFS volume into a datastore,” “Mapping a system LUN” or “mapping a disk file to a physical disk volume.” These terms all refer to raw device mapping.

The following terms are used in this document or related documentation:

- **Raw Disk** — A disk volume accessed by a virtual machine as an alternative to a virtual disk file; it may or may not be accessed via a mapping file.
- **Raw Device** — Any SCSI device accessed via a mapping file. For ESX Server 2.5, only disk devices are supported.
- **Raw LUN** — A logical disk volume located in a SAN.
- **LUN** — Acronym for a logical unit number.
- **Mapping File** — A VMFS file containing metadata used to map and manage a raw device.
- **Mapping** — An abbreviated term for a raw device mapping.
- **Mapped Device** — A raw device managed by a mapping file.
- **Metadata File** — A mapping file.
- **Compatibility Mode** — The virtualization type used for SCSI device access (physical or virtual).
- **SAN** — Acronym for a storage area network.
- **VMFS** — A high-performance file system used by VMware ESX Server.



Raw Device Mapping Redirects Data Transfers



Should I Use Raw Device Mapping?

Raw device mapping provides a number of benefits, but it can't be used in every situation. In general, virtual disk files are preferable for manageability. When you need raw devices, the use of a mapping file is preferred for most situations. This section explains both the benefits and limitations of raw device mapping.

Benefits of Raw Device Mapping

User-Friendly Persistent Names

Raw device mapping provides a user-friendly name for a mapped device — the name of its mapping file. When you use a mapping, you don't need to refer to the device by its device name, as required with previous versions of ESX Server. You refer to it by the name of the mapping file — for example, use:

```
/vmfs/myvmfsvolume/myrawdisk.vmdk.
```

Dynamic Name Resolution

Raw device mapping stores unique identification information for each mapped device. The VMFS file system resolves each mapping to its current SCSI device, regardless of changes in the physical configuration of the server due to adapter hardware changes, path changes, device relocation, and so forth.

Distributed File Locking

Raw device mapping makes it possible to use VMFS distributed locking for raw SCSI devices. Distributed locking on a raw device mapping makes it safe to use a shared raw LUN without losing data when two virtual machines on different servers access the same LUN.

File Permissions

Raw device mapping makes file permissions possible. The permissions of the mapping file are applied at file open time to protect the mapped volume. Previously, permissions for raw devices could not be enforced by the file system.

File System Operations

Raw device mapping makes it possible to use file system utilities to work with a mapped volume, using the mapping file as a proxy. Most operations that are valid for an ordinary file can be applied to the mapping file, and are redirected to operate on the mapped device.

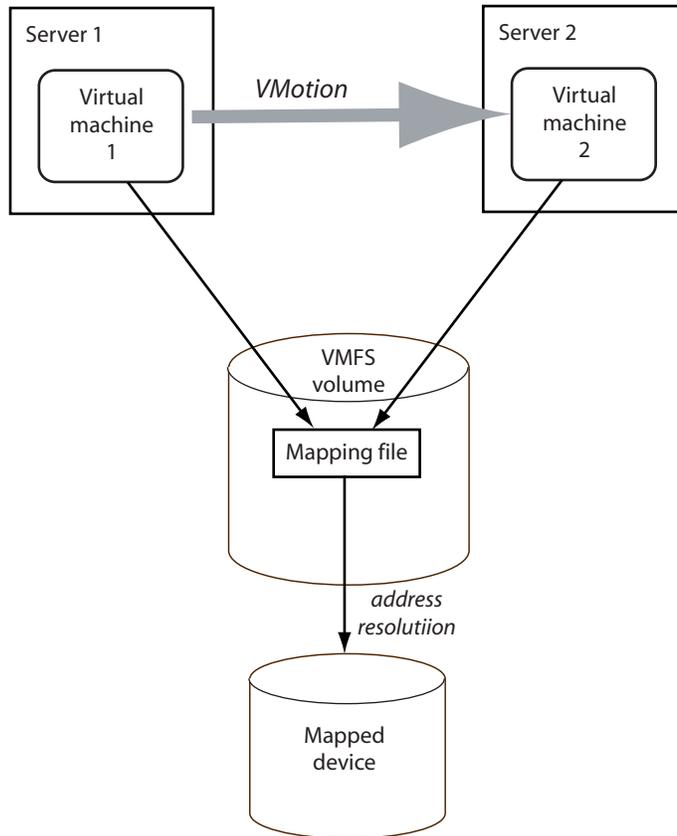
Redo Logs

Raw device mapping makes it possible to keep a redo log for a mapped volume. The redo log has the name of the mapping file, with `.REDO` appended. Note that redo logs are not possible when raw device mapping is used in physical compatibility mode.



VMotion

Raw device mapping allows you to migrate a virtual machine with VMotion. Previously, this was only possible for virtual machines that used virtual disk files. When you use raw device mapping, the mapping file acts as a proxy to allow VirtualCenter to migrate the virtual machine using the same mechanism that exists for virtual disk files.



VMotion of a Virtual Machine Using a Raw Device Mapping

Note: You cannot migrate virtual machines with raw, clustered, or non-persistent mode disks using VMotion. If you have clustered disks, you can store them on separate VMFS volumes from the virtual machines prior to migrating them using VMotion.



SAN Management Agents

Raw device mapping makes it possible to run some SAN management agents inside a virtual machine. Similarly, any software that needs to access a device using raw SCSI commands can be run inside a virtual machine. This kind of software may be referred to as "SCSI target based software".

This use requires configuring physical compatibility mode for the mapping file.

Note: VMware works with vendors of storage management software to ensure that their software functions correctly in environments that include ESX Server. Some applications of this kind are:

- SAN management software
- storage resource management (SRM) software
- snapshot software
- replication software

Such software uses physical compatibility mode for raw device mappings, so that the software can access SCSI devices directly.

Various management products are best run centrally (not on the ESX Server machine), while others run well in the service console or in the virtual machines. VMware does not certify these applications or provide a compatibility matrix. To find out if a SAN management application is supported in an ESX Server environment, please contact the SAN management software provider.

Limitations of Raw Device Mapping

Not Available for Block Devices or RAID Devices

Raw device mapping (in the current implementation) uses a SCSI serial number to identify the mapped volume. Since block devices and some direct-attach RAID devices do not export serial numbers, they can't be used in raw device mappings.

Not Available for Devices Attached to a Shared Adapter

If your SCSI adapter is configured as a shared adapter, you can't use raw device mapping for any of its devices. Only adapters dedicated to the VMkernel support raw device mapping.

Available with VMFS-2 Volumes Only

Raw device mapping requires the VMFS-2 format. If you choose not to convert your VMFS volume from VMFS-1 format to VMFS-2 format, you cannot use raw device mapping.

No Redo Log in Physical Compatibility Mode

If you are using raw device mapping in physical compatibility mode, you can't use a redo log with the disk. Physical compatibility mode allows the virtual machine to manage its own snapshot or mirroring operations. This conflicts with the SCSI virtualization objectives of physical compatibility mode.

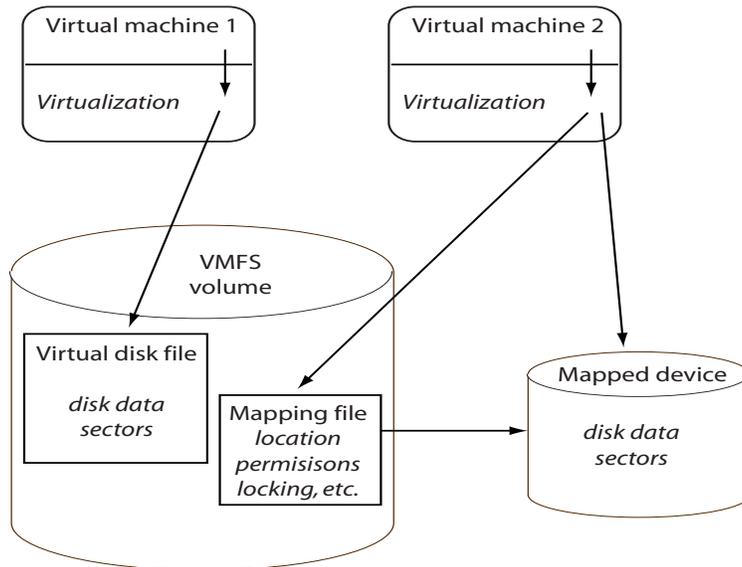
Redo logs are available, however, in virtual mode. The compatibility mode is explained in more detail later in this document.



More About Raw Device Mapping

A raw device mapping file is a special file in a VMFS volume that manages metadata for its mapped device. There is a one-to-one mapping between mapping files and mapped devices. The mapping file is presented to the VMware Service Console as an ordinary disk file, available for the usual file system operations. To the virtual machine, the storage virtualization layer presents the mapped device as a virtual SCSI device.

Key contents of the metadata in the mapping file include the location of the mapped device (name resolution) and the locking state of the mapped device.



Mapping File Metadata

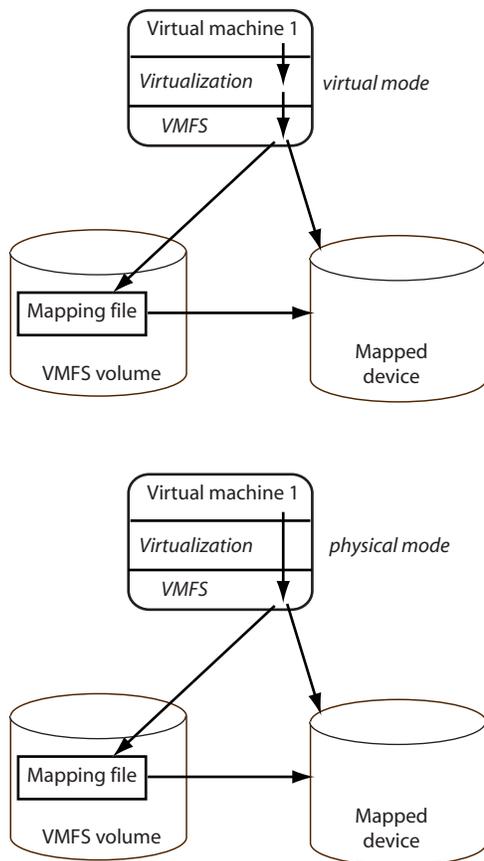
Note that raw device mapping requires the mapped device to be a whole LUN; mapping to a partition only is not supported.



Virtual Compatibility Mode Versus Physical Compatibility Mode

Virtual mode for a mapping specifies full virtualization of the mapped device. It appears to the guest operating system exactly the same as a virtual disk file in a VMFS volume. The real hardware characteristics are hidden. Virtual mode allows customers using raw disks to realize the benefits of VMFS such as advanced file locking for data protection and redo logs for streamlining development processes. Virtual mode is also more portable across storage hardware, presenting the same behavior as a virtual disk file.

Physical mode for a raw device mapping specifies minimal SCSI virtualization of the mapped device, allowing the greatest flexibility for SAN management software. In physical mode, the VMkernel passes all SCSI commands to the device, with one exception: The REPORT LUNs command is virtualized, so that the VMkernel can isolate the LUN for the owning virtual machine. Otherwise, all physical characteristics of the underlying hardware are exposed. Physical mode is useful to run SAN management agents or other SCSI target based software in the virtual machine. Physical mode also allows virtual to physical clustering for cost-effective high availability



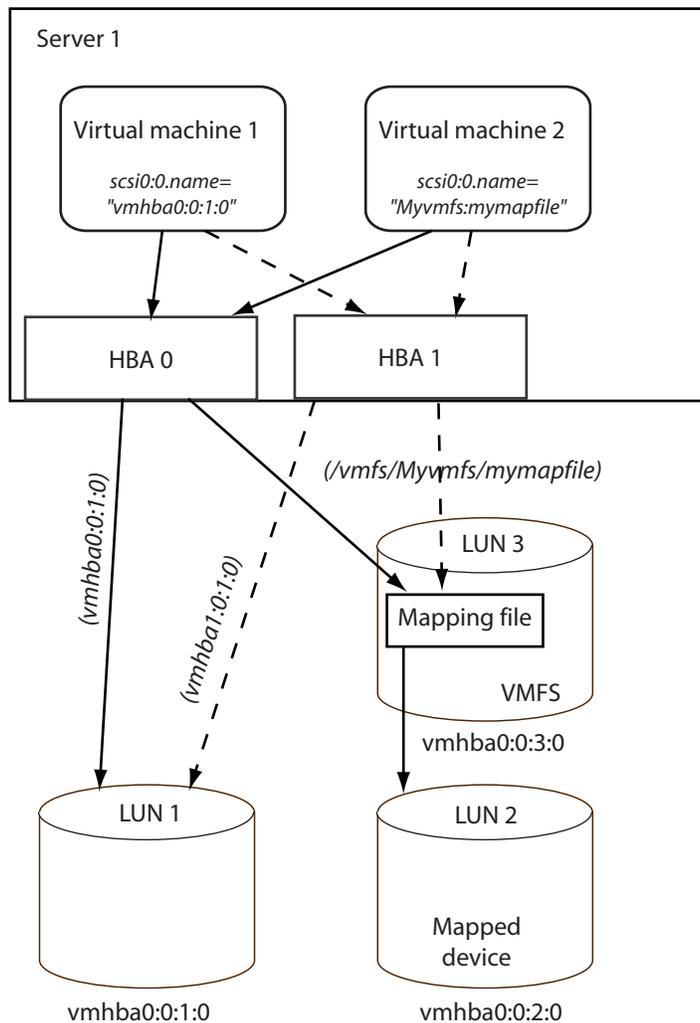
Virtual And Physical Compatibility Modes



Dynamic Name Resolution

Previously, raw access to disks or partitions required you to use the device name for the disk. Whenever the path to a SAN LUN changed, or new adapters were added to the server, the device name could change, requiring you to change the virtual machine configuration to compensate. Raw device mapping allows you to give a permanent name to a device by referring to the name of the mapping file in the `/vmfs` directory.

The example in the following illustration shows three LUNs. LUN 1 is accessed by its device name, which is relative to the first visible LUN. LUN 2 is a mapped device, managed by a mapping file on LUN 3. The mapping file is accessed by its path name in the service console, which is fixed.



Example of Name Resolution

Suppose HBA 0 is taken offline. The HBA numbering is relative, so when the server reboots, HBA 1 now becomes HBA 0. The device name for LUN 1 now becomes `vmhba0:0:1:0` instead of `vmhba1:0:1:0`. Virtual machine 1 is unable to reference LUN 1 until the virtual machine



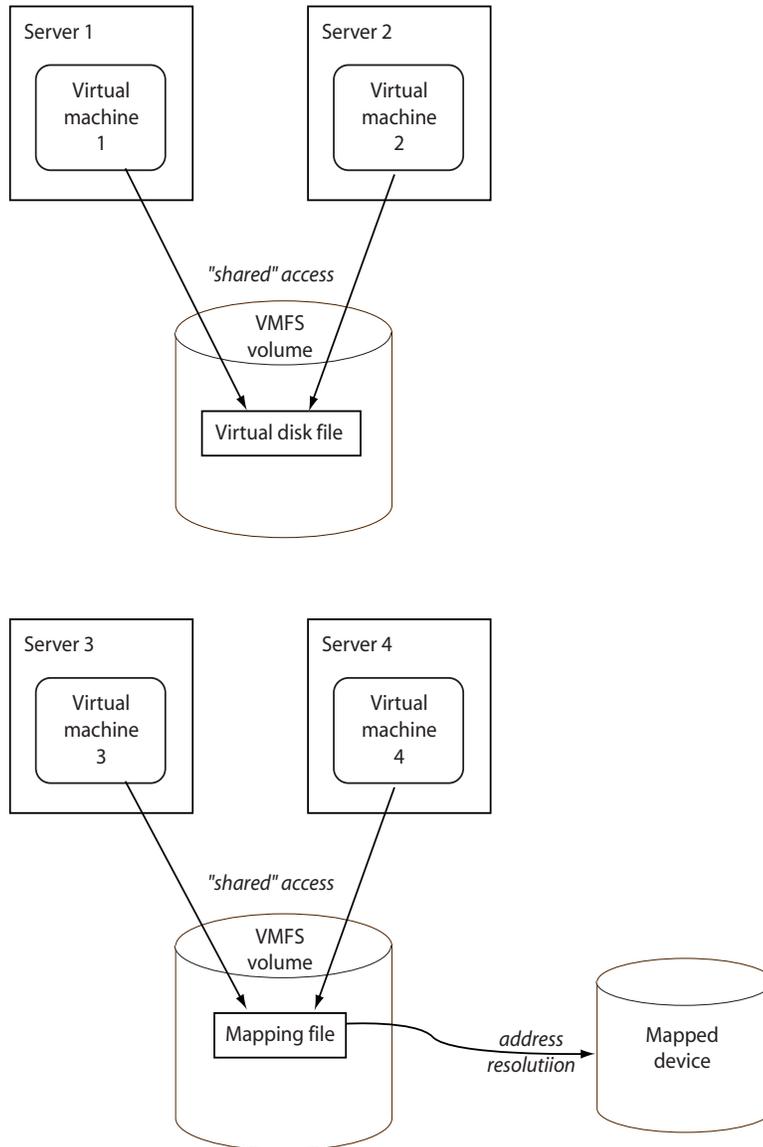
configuration is manually re-targeted to the new device name. Virtual machine 2, however, is able to access LUN 2 normally, because the VMkernel re-targets the mapping automatically.

All mapped LUNs are uniquely identified by the VMkernel and the identification is stored in its internal data structures. Any change in the SCSI path, such as a Fibre Channel switch failure or the addition of a new host bus adapter, has the potential to change the vmhba device name, because the name includes the path designation (Initiator, Target, LUN). Dynamic name resolution compensates for all such changes by adjusting the data structures to re-target LUNs to their new device names.



Raw Device Mapping with Virtual Machine Clusters

VMware recommends the use of raw device mapping with virtual machine clusters that need to access the same raw LUN for failover scenarios. The setup is similar to a virtual machine cluster that accesses the same virtual disk file, but a raw device mapping file replaces the virtual disk file. The VMFS must be configured in shared access mode, to allow more than one virtual machine to open the mapping file simultaneously.



Access from Clustered Virtual Machines



Comparing Raw Device Mapping to Other Means of SCSI Device Access

To help you choose between several available access modes for SCSI devices, the following table provides a quick comparison of features available with the different modes.

	Virtual Disk File	Virtual Mapping	Physical Mapping	Raw Disk
SCSI Commands Passed Through	NO	NO	YES (note 1)	YES (note 1)
VirtualCenter Support	YES	YES	YES	NO
Redo Logs	YES	YES	NO	NO
Distributed Locking	YES	YES	YES	NO
Clustering	All except N+1	All except N+1	All except CIB (note 2)	All except CIB (note 2)
SCSI Target-Based Software	NO	NO	YES	YES

Note 1. REPORT LUNS is not passed through

Note 2. CIB = Cluster-In-a-Box



Managing Raw Device Mappings

Utilities

The tools available to manage raw device mappings include the VMware Management Interface, the `vmkfstools` utility and ordinary file system utilities used in the VMware Service Console. VMware VirtualCenter can also be used to manage raw device mappings. Refer to www.vmware.com/support/vc12/doc/c10maplunex.html for VirtualCenter documentation.

VMware Management Interface

The management interface contains pages to configure hardware and add disks, where you can create a raw device mapping. The management interface also has a file manager that can be used to import and export virtual disks, including mapped raw devices. You can also perform common file system operations, such as renaming and setting permissions, in the file manager.

Some of the procedures are described in [Sample Operations on page 14](#).

vmkfstools

The `vmkfstools` utility can be used in the service console to do many of the same operations available in the management interface. Typical operations applicable to raw device mappings are the commands to create a mapping, to query mapping information such as the name and identification of the mapped device, and to import or export a virtual disk. The form of these commands is shown here:

```
vmkfstools -r <mapped device> <mapping file>
vmkfstools -P <mapping file>
vmkfstools -i <template disk> <mapping file>
vmkfstools -e <template disk> <mapping file>
```

Some sample commands are shown in [Sample Operations on page 14](#).

File System Operations

Most common file system operations done in the service console can be applied to raw device mappings. The results for raw device mappings may be different from what you expect, in some cases.

```
ls -l
```

The `ls` command with the `-l` option shows the file name and permissions of the mapping file, while showing the length of the mapped device.

```
du
```

Similarly, the `du` command shows the space used by the mapped device, rather than the mapping file.

```
mv
```

The `mv` command renames the mapping file, but doesn't affect the mapped device.

```
cp
```

You can use the `cp` command to copy the contents of a mapped device, but it does not work in reverse: You can't use it to copy a virtual disk file to a mapped device. Instead, you can use the `dd` command.

```
dd
```

You can use the `dd` command to copy data into or out of a mapped device. It does not copy the contents of the mapping file.



Sample Operations

This section demonstrates several operations involving raw device mappings. You can perform these operations either in the VMware Service Console or in the VMware Management Interface. Both methods are described.

The following steps walk through this sequence of operations:

1. Create a raw device mapping
2. Import a virtual disk template file to a mapped device
3. Clone the mapping file and the mapped device
4. Query and change permissions of a mapping
5. Export a mapped device to a virtual disk template

Create a Raw Device Mapping

Suppose you want to create a disk as a raw device mapping named `Barney.vmdk` in a VMFS volume named `vmfs1`. You want to map to LUN 12 on target 2 from HBA 0. You want to attach this mapping to a virtual machine named `/vm/Rubble.vmx`, and you want physical compatibility mode for the guest. In the service console, type:

```
vmkfstools -r vmhba0:2:12 /vmfs/Vmfs1/Barney.vmdk
```

Edit the configuration file for the virtual machine and add lines similar to these:

```
scsi0:0.name = "Vmfs1:Barney.vmdk"
scsi0:0.deviceType = "scsi-passthru-rdm"
```

The value `scsi-passthru-rdm` specifies physical compatibility mode.

Note: This example assumes you have no other virtual disk attached to the virtual machine.

If you use the management interface, on the other hand, follow these steps:

1. Log on as the root user or as the owner of the virtual machine to which the mapped disk will belong.
2. On the Status Monitor tab, click the arrow to the right of the virtual machine icon in the list of virtual machines.
3. From the drop-down menu, choose **Configure Hardware**.
4. In the Hardware Configuration window, click **Add Device**.
5. In the Add Device window, click **Hard Disk**.
6. Click **System LUN/Disk**.
7. Choose LUN `vmhba0:2:12:` from the Storage Controller LUN list.
8. Select **Use Metadata**, then choose the Metadata File Location.
9. Enter the name of the mapping file: `/vmfs/Vmfs1/Barney.vmdk`
10. Select a virtual SCSI identifier from the Virtual SCSI Node list.
11. Choose **Physical** for the Compatibility mode.
12. Click **OK** to add the disk.

Import a Virtual Disk Template File to a Mapped Device

Suppose you want to copy the contents of the new mapped disk from a GSX Server virtual disk file at `/templates/Rubble.vmdk`. In the service console, use this command:

```
vmkfstools -i /templates/Rubble.vmdk /vmfs/Vmfs1/Barney.vmdk
```



Alternatively, in the management interface, perform these steps:

1. Log on as the root user or as the owner of the virtual machine to which the mapped disk belongs.
2. Click the **Manage Files** link near the upper right corner.
3. In the file tree pane at the left side of the File Manager window, click `/templates`.
4. In the Current Directory pane at the right of the window, click the selection checkbox to the left of `Rubble.vmdk`.
5. At the bottom of the Current Directory pane, click the **Copy** button.
6. In the file tree pane, click `vmfs`, then `Vmfs1`.
7. In the Current Directory pane, click the **Paste** button.

Clone the Mapping and the Mapped Device

Suppose you want to clone the mapping to a new mapping named `/vmfs/Vmfs1/Fred.vmdk`, using a mapped device at LUN 14 on target 2 from HBA 0. The easiest way to do this is to export `Barney.vmdk` to a template virtual disk file, then import the template to `Fred.vmdk`. In the service console, you can use commands similar to these:

```
vmkfstools -e /tmp/Slate.vmdk Vmfs1:Barney.vmdk
vmkfstools -r vmhba0:2:14:0 /vmfs/Vmfs1/Fred.vmdk
vmkfstools -i /tmp/Slate.vmdk Vmfs1:Fred.vmdk
```

Note the intermediate command that creates a raw device mapping for a destination. If you omit that step, the result will be a virtual disk file instead of a raw device mapping.

Query and Change Permissions of a Mapping

Next you want to change the permissions on `Fred.vmdk` so that all users have write access. Using the service console, query the permissions and then set them for the mapping just like an ordinary Linux file:

```
ls -l /vmfs/Vmfs1/Fred.vmdk
chmod 0777 /vmfs/Vmfs1/Fred.vmdk
```

Alternatively, using the management interface, perform these steps:

1. Log on as the root user or as the owner of the virtual machine to which the mapped disk belongs.
2. Click the **Manage Files** link near the upper right corner.
3. In the file tree pane at the left side of the File Manager window, click `vmfs`, then `Vmfs1`.
4. In the Current Directory pane at the right of the window, click the selection checkbox to the left of `Fred.vmdk`.
5. At the bottom of the Current Directory pane, click the **Edit Properties** button.
6. In the Edit window, select all nine access checkboxes, then click the **OK** button.

Export a Mapped Device to a Virtual Disk Template

Finally, suppose you want to export the mapped disk `Fred.vmdk` to a GSX Server virtual disk file named `Flintstone.vmdk`.

In the service console, type this command:

```
vmkfstools -e /templates/Flintstone.vmdk Vmfs1:Fred.vmdk
```

Alternatively, in the management interface, perform these steps:



1. Log on as the root user or as the owner of the virtual machine to which the mapped disk belongs.
2. Click the **Manage Files** link near the upper right corner.
3. In the file tree pane at the left side of the File Manager window, click `vmfs`, then `vmfs1`.
4. In the Current Directory pane at the right of the window, click the selection checkbox to the left of `Fred.vmdk`.
5. At the bottom of the Current Directory pane, click the **Copy** button.
6. In the file tree pane, click `/templates`.
7. In the Current Directory pane, click the **Paste** button.