

Greater than 2TB support on different Operation Systems

Areca Raid Controller have 2 features to support a volume with greater than 2TB capacity :

1. **4K Block (For Windows)** : It change the sector size from standard 512 to 4K. the maximum volume capacity up to 16TB.

This option works under Windows platform only, and it **CAN NOT** be converted to Dynamic Disk, because 4K is not a standard sector size.

P.S. This option available in SATA controllers only, not SAS controllers

2. **LBA 64** : This option use 16 byte CDB instead original 10byte. LBA64 is the standard method to addressing a device with over 2TB capacity. The maximum volume capacity up to 512TB.

This option works on different OS which supports 16byte CDB, such as :

Windows 2003 with SP1 or later

Linux kernel 2.6.x or latter

FreeBSD 5.2.1 or latter

P.S. Both Windows 2000 and XP 32bit don't support LBA64 mode.

Something important about the 4K Block sector size ...

Because 4K is not standard sector size, if you would like to use this mode, please make sure your software have such ability to manage a storage device with 4K sector size. If software can't manage it correctly, it may cause unexpected data corruption problem.

Method 1 : 4K Block (For Windows)

1. Press F6 or TAB to enter configuration menu.
2. Select which controller to configured, then Choose **Quick Volume/Raid Setup**.
3. Enter password. (default is **0000**)
4. Choose raid level. If volume capacity will exceed 2TB, controller will show the **Greater 2 TB volume Support** sub-menu.
5. Choose **Use 4K Block**



6. Configure other parameters, then exit configuration menu to confirm your operations.

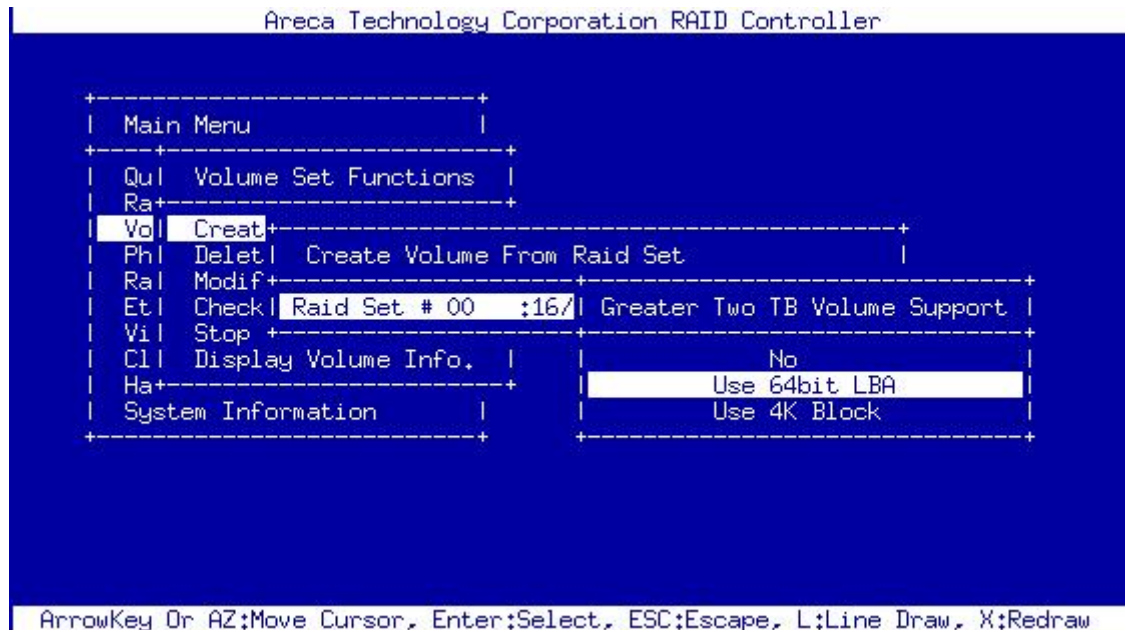
You can also refer user's manual **Chapter 3. BIOS Configuration**

4K Block mode can be configured through BIOS console only !

Method 2 : LBA 64

A. If you are using BIOS console to configure raidset :

- A1. Press F6 or TAB to enter configuration menu.
- A2. Select which controller to configured, then Choose **Quick Volume/Raid Setup**.
- A3. Enter password. (default is **0000**)
- A4. Choose raid level. If volume capacity will exceed 2TB, controller will show the **Greater 2 TB volume Support** sub-menu.
- A5. Choose **Use 64bit LBA**



- A6. Configure other parameters, then exit configuration menu.

You can also refer user's manual **Chapter 3. BIOS Configuration**

B. If you are using Web Browser-Based Configuration to configure raidset :

- B1. Start a browser, enter the IP address of raid controller, then login McRAID manager.
(The default user name is **admin**, and password is **0000**)
- B2. From Main Menu, choose **Quick Function > Quick Create**.
- B3. Choose raid level. If volume capacity will exceed 2TB, controller will enable the **Greater 2 TB volume Support** selection menu.
- B4. Choose **64bit LBA**

Enter Volume Attribute On Raid Set # 00

Volume Name	<input type="text" value="ARC-1261-VOL#02"/>		
Member Disks	<input type="text" value="16"/>		
Volume Raid Level	<input type="text" value="Raid 6"/> ▼		
Max Capacity Allowed	10300 GB		
Select Volume Capacity	<input type="text" value="10300"/>	GB	
Greater Two TB Volume Support	<input type="text" value="No"/> ▼		
Volume Initialization Mode	<div><div>No</div><div>64bit LBA</div></div>	Initialization	<input type="text" value=""/> ▼
Volume Stripe Size	<input type="text" value="64"/> ▼	KBytes	
Volume Cache Mode	<input type="text" value="Write Back"/> ▼		
Tagged Command Queuing	<input type="text" value="Enabled"/> ▼		
SCSI Channel:SCSI ID:SCSI Lun	<input type="text" value="0"/> ▼	: <input type="text" value="0"/> ▼	: <input type="text" value="1"/> ▼

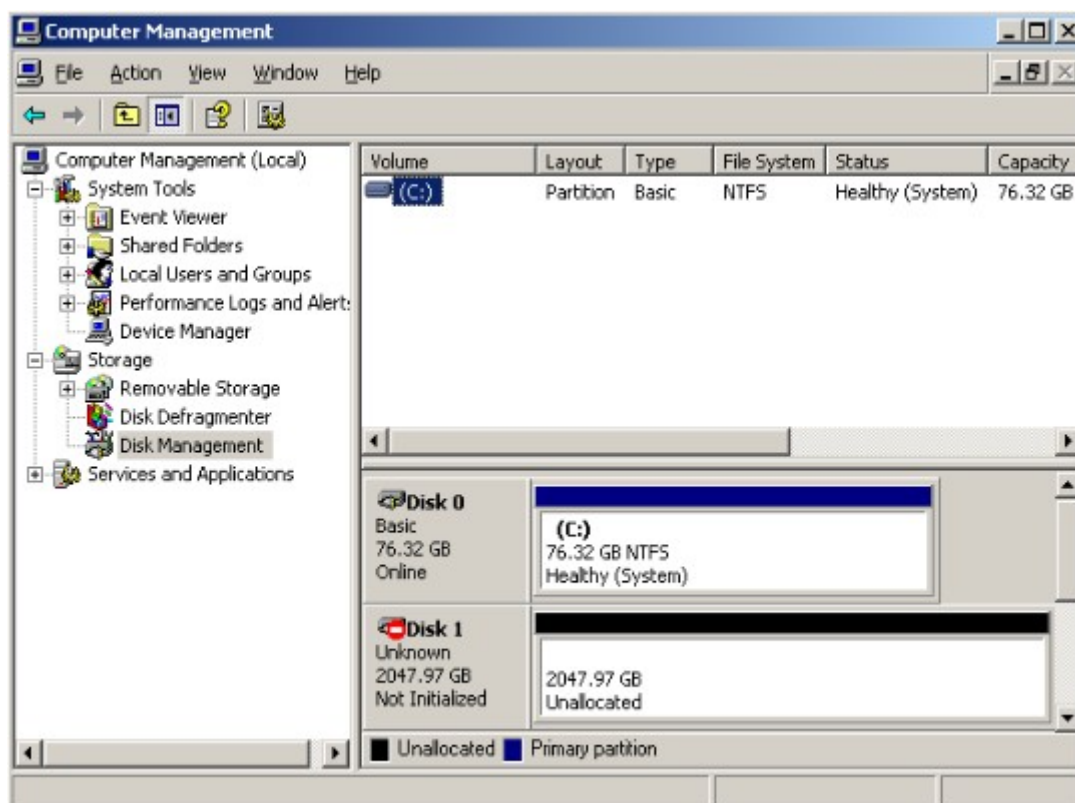
B5. Configure other parameters, then check the **Confirm The Operation**, click **Submit** icon to completed.

You can also refer user's manual **Chapter 6. Web Browser-Based Configuration**

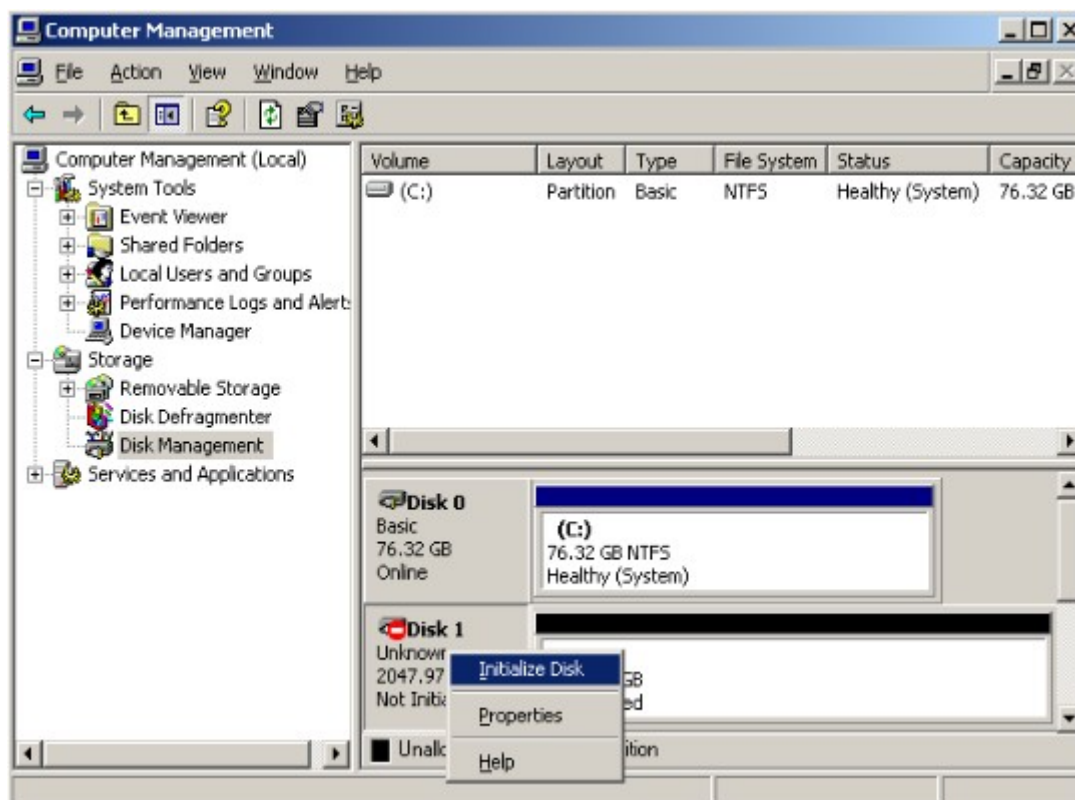
Using greater than 2TB volume in Windows by “4K Block (For Windows)” option

1. Using “4K Block (For Windows)” function to support greater 2TB volume in Windows 2003 SP1:

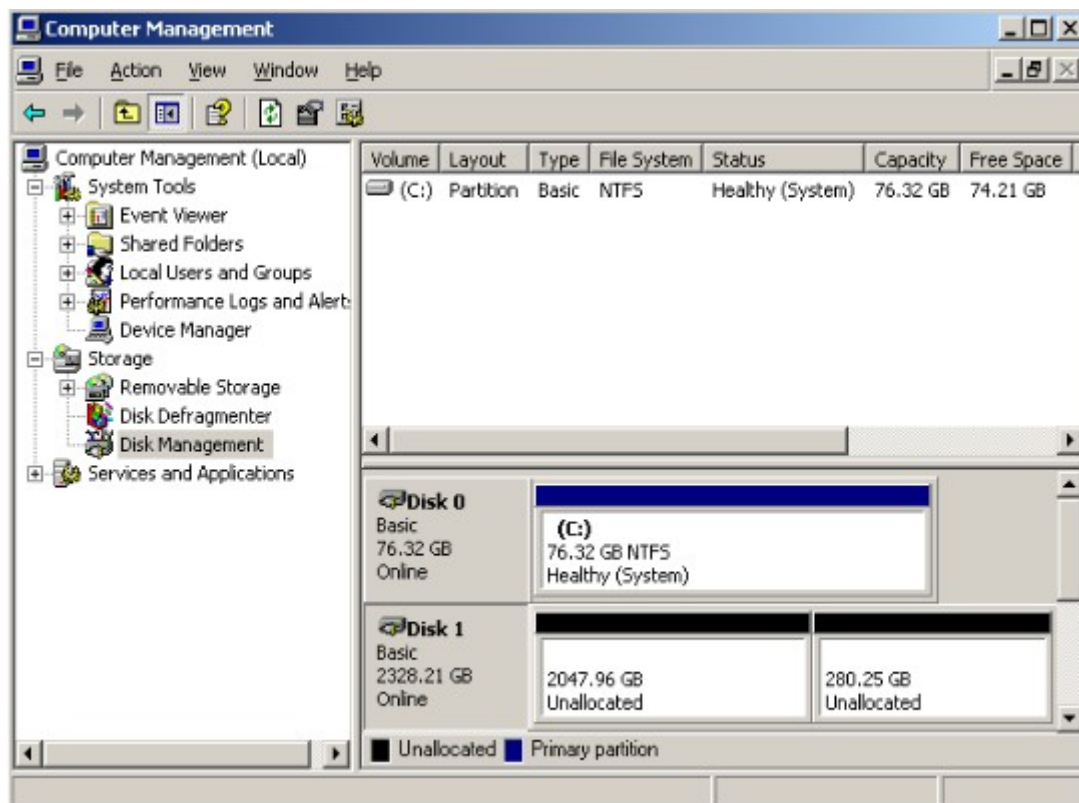
1.1 The greater than 2TB volume will listed as an unknown device in Disk Management :



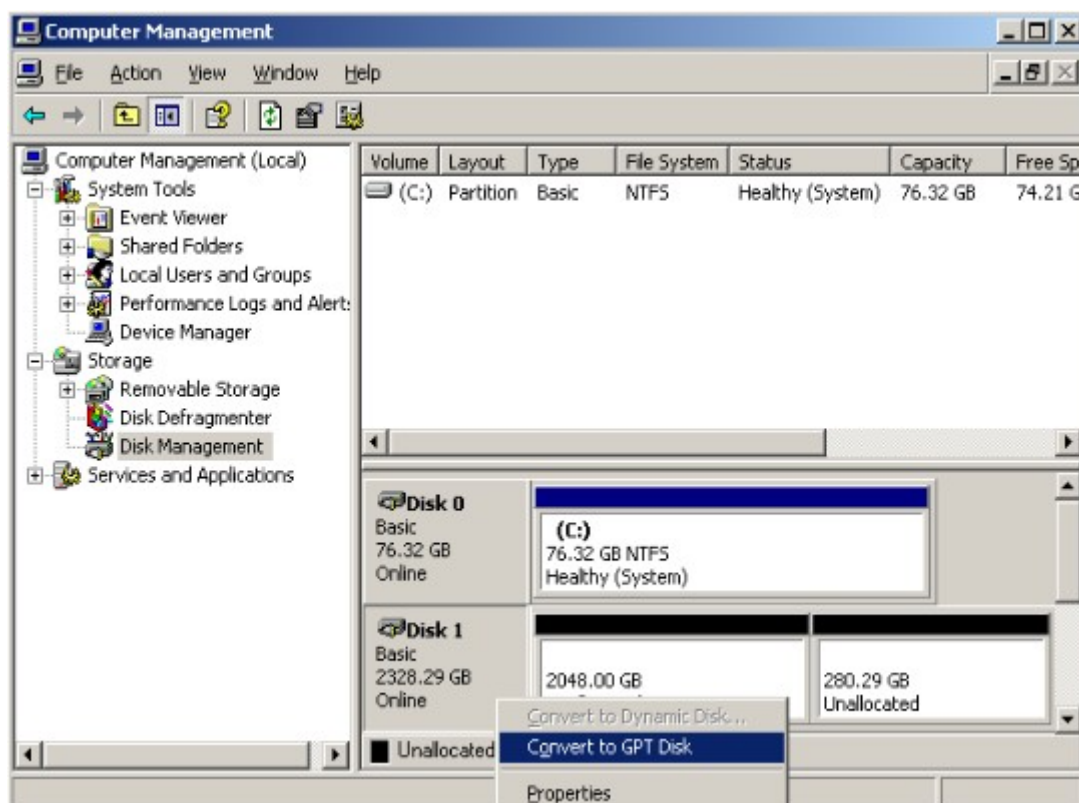
1.2 You need to execute **Initialize Disk** first.



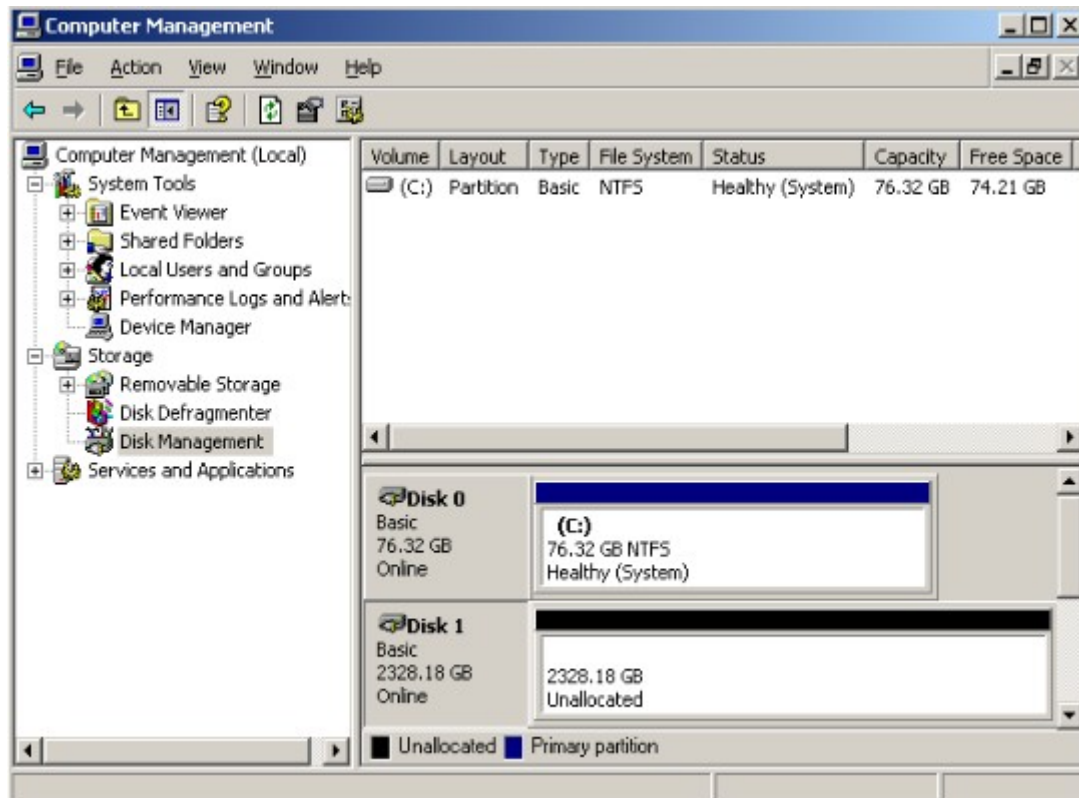
1.3 After initialize this disk, you will able to see a device divided to 2 parts : a 2TB space and the rest :



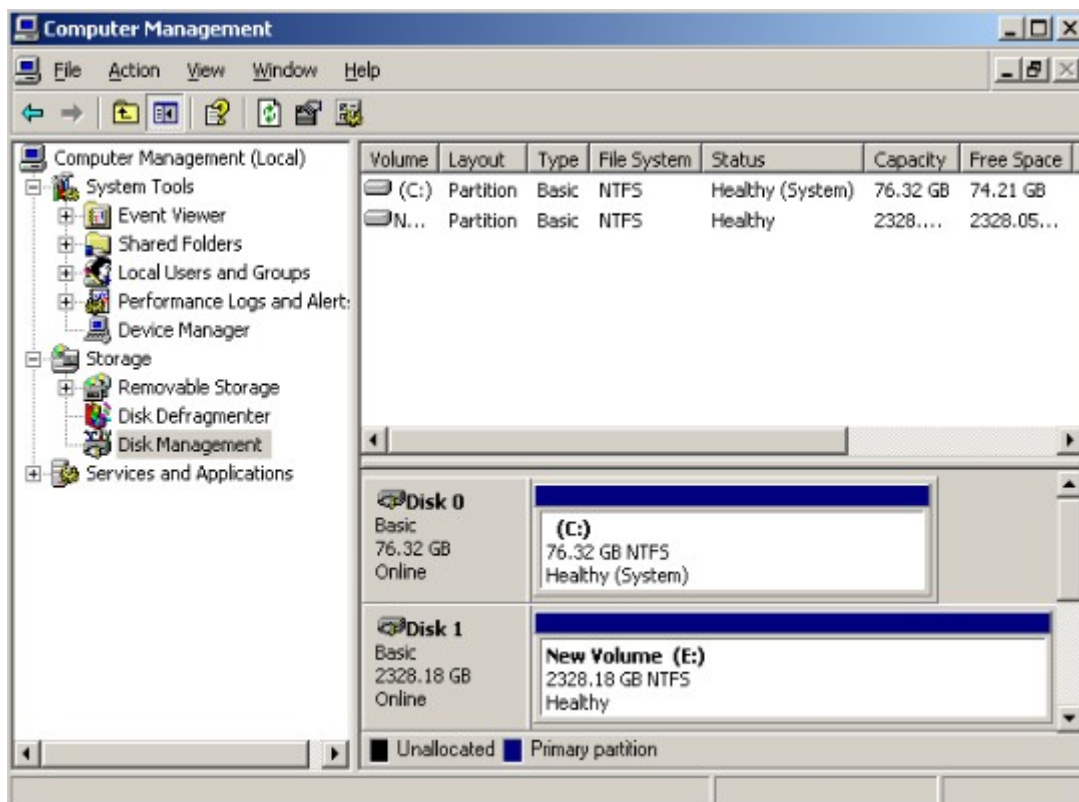
1.4 If you want to configure a partition with greater than 2TB capacity, you need convert this disk to GPT Disk. :



1.5 After you convert it to GPT Disk, the device will become a single disk with full capacity.



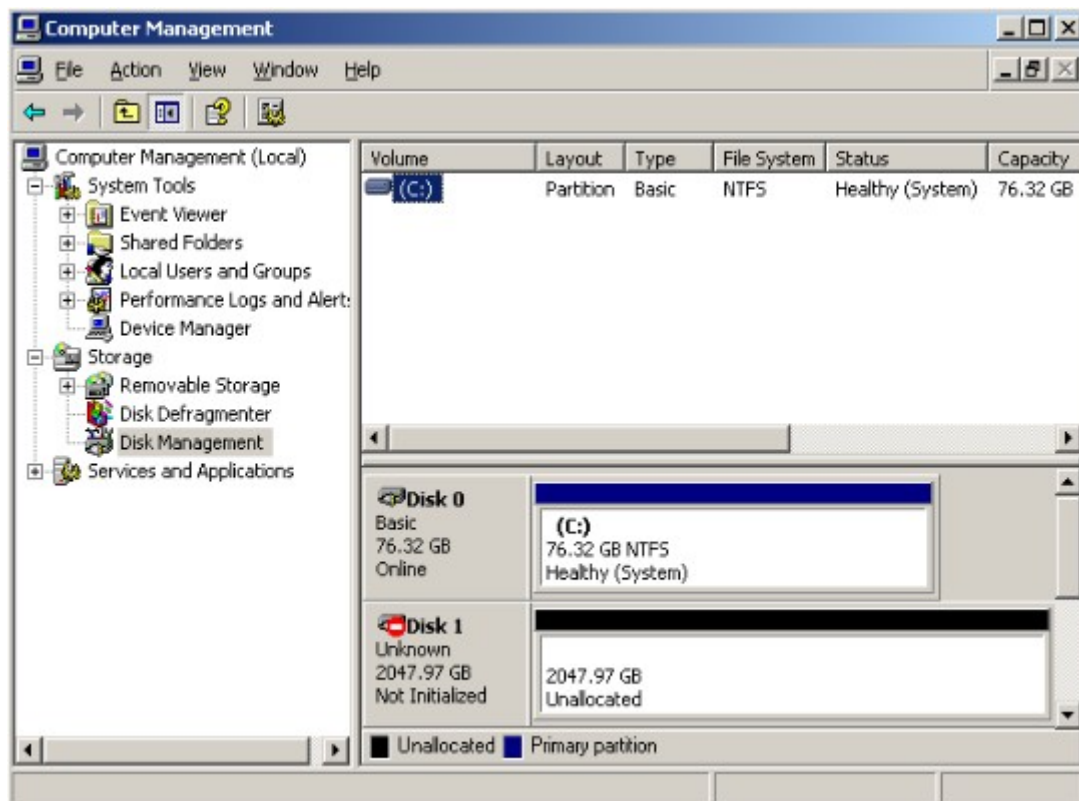
1.6 Now, you can create a partition with greater than 2TB capacity.



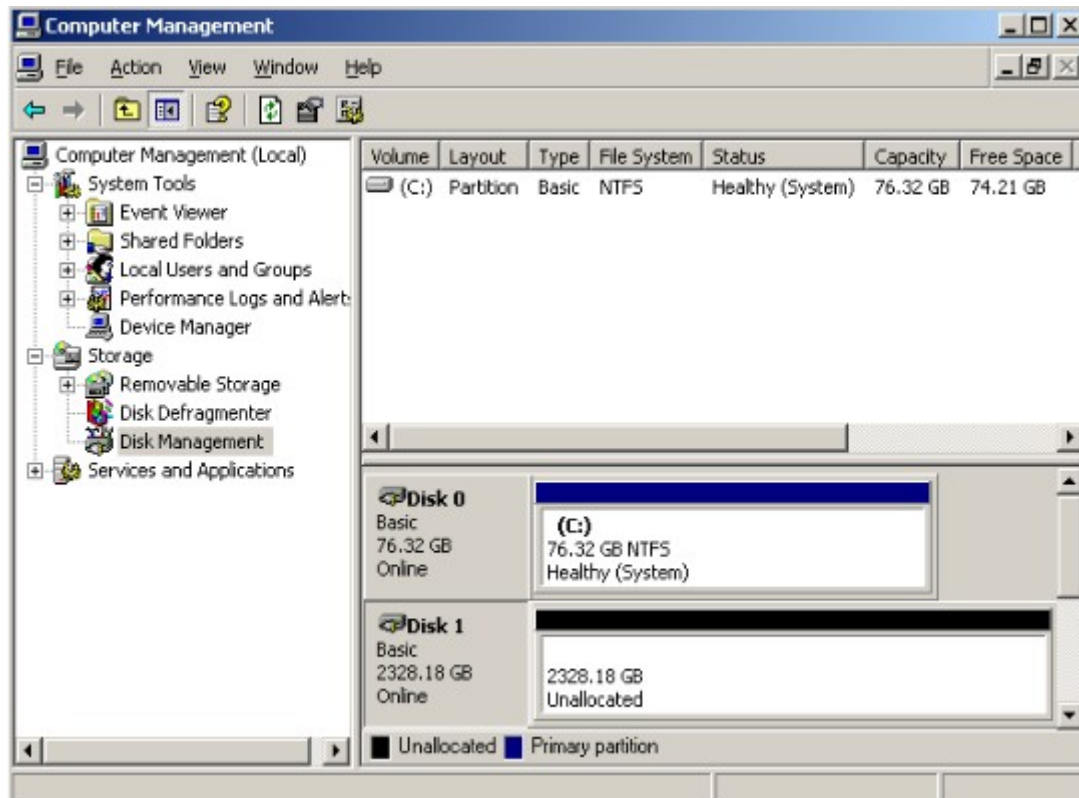
Note : The greater than 2TB Volume created by “4K Block (For Windows)” CAN NOT be converted to Dynamic Disk. Because 4K is not a standard sector size.

2. Using “For Windows” function to support greater 2TB volume in Windows 2000, XP :

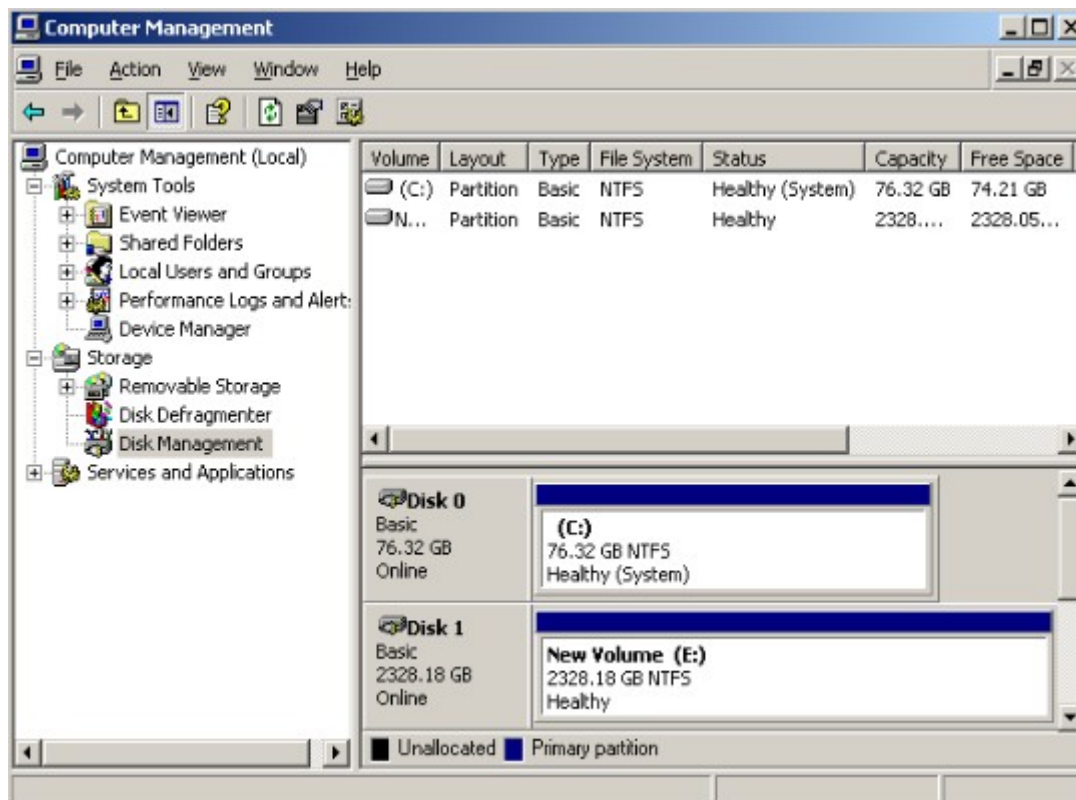
2.1 The greater than 2TB volume will listed as an unknown device in Disk Management :



2.2 After initialize this disk, the device will listed with full capacity.



2.3 Now, you can create a partition with greater than 2TB capacity.

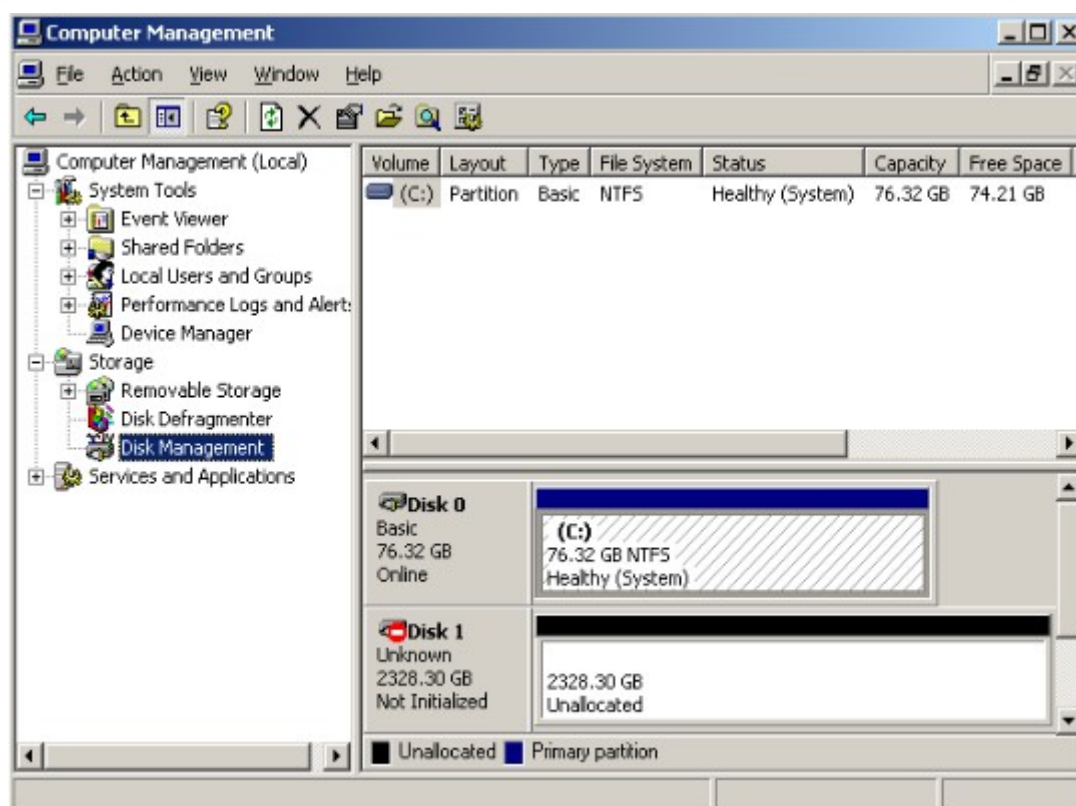


Note : The greater than 2TB Volume created by “4K Block (For Windows)” **CAN NOT** be converted to Dynamic Disk. Because 4K is not a standard sector size.

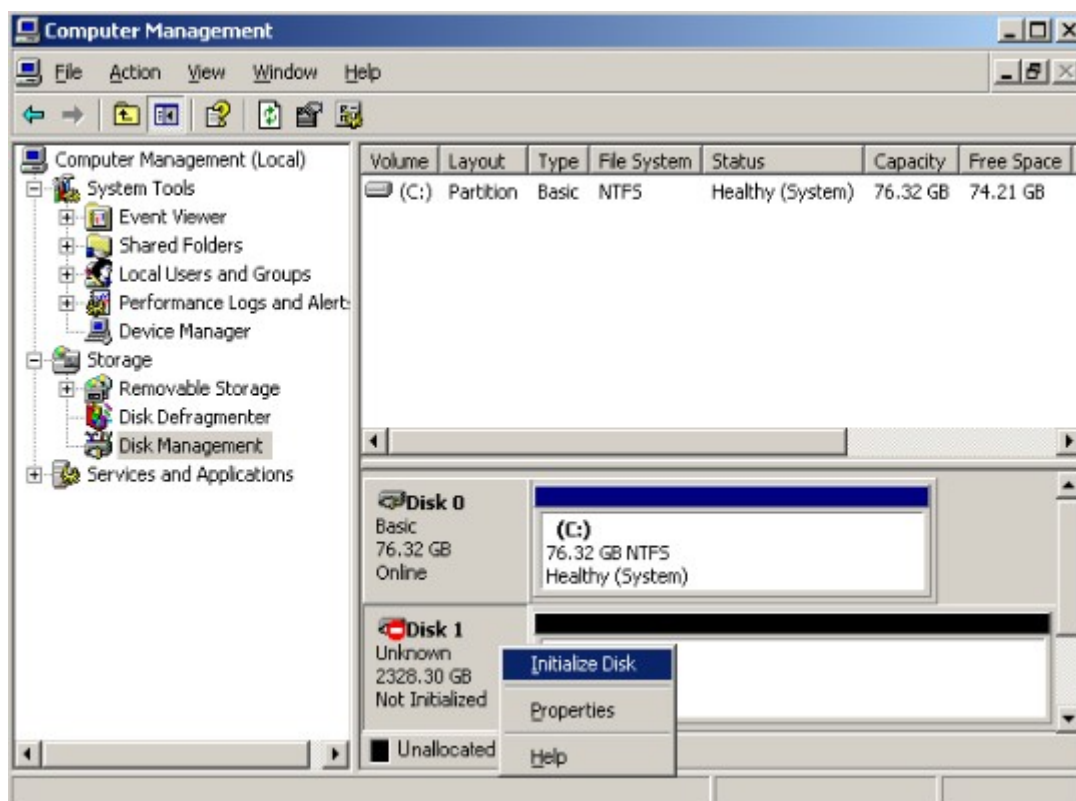
Using greater than 2TB volume in Windows by “LBA 64” option

1. Using “LBA 64” function to support greater 2TB volume in Windows 2003 SP1:

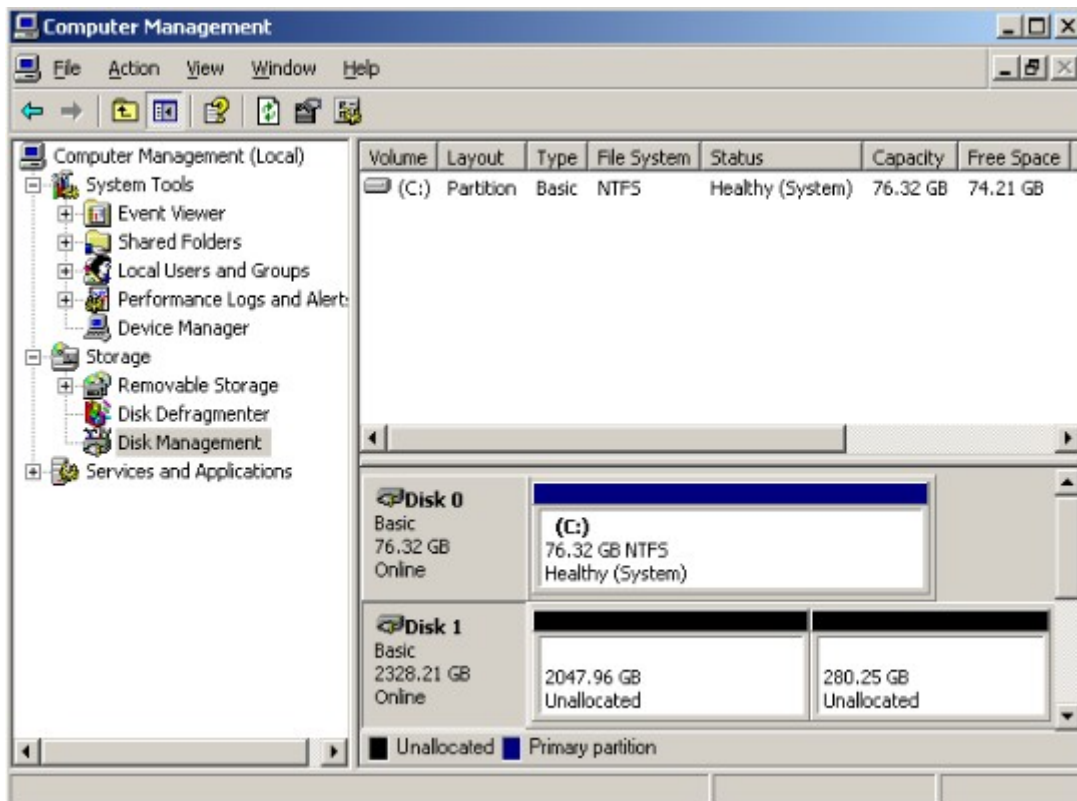
1.1 The greater than 2TB volume will listed as an unknown device in Disk Management :



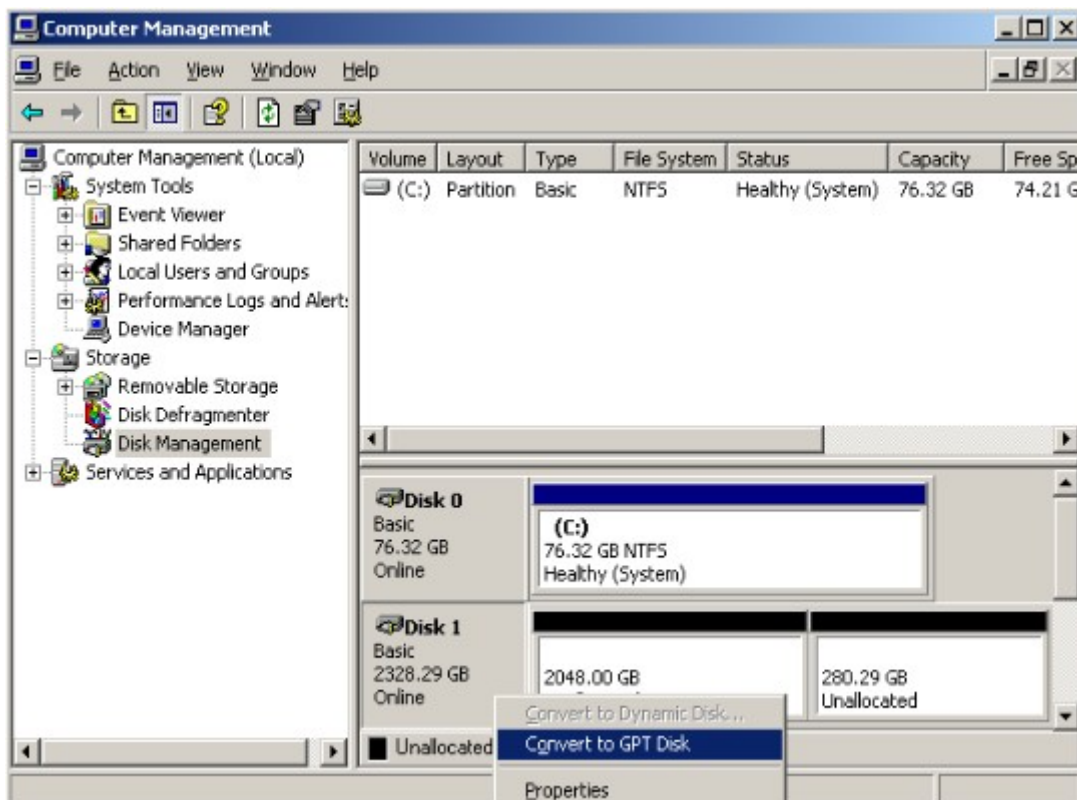
1.2 You need to execute **Initialize Disk** first.



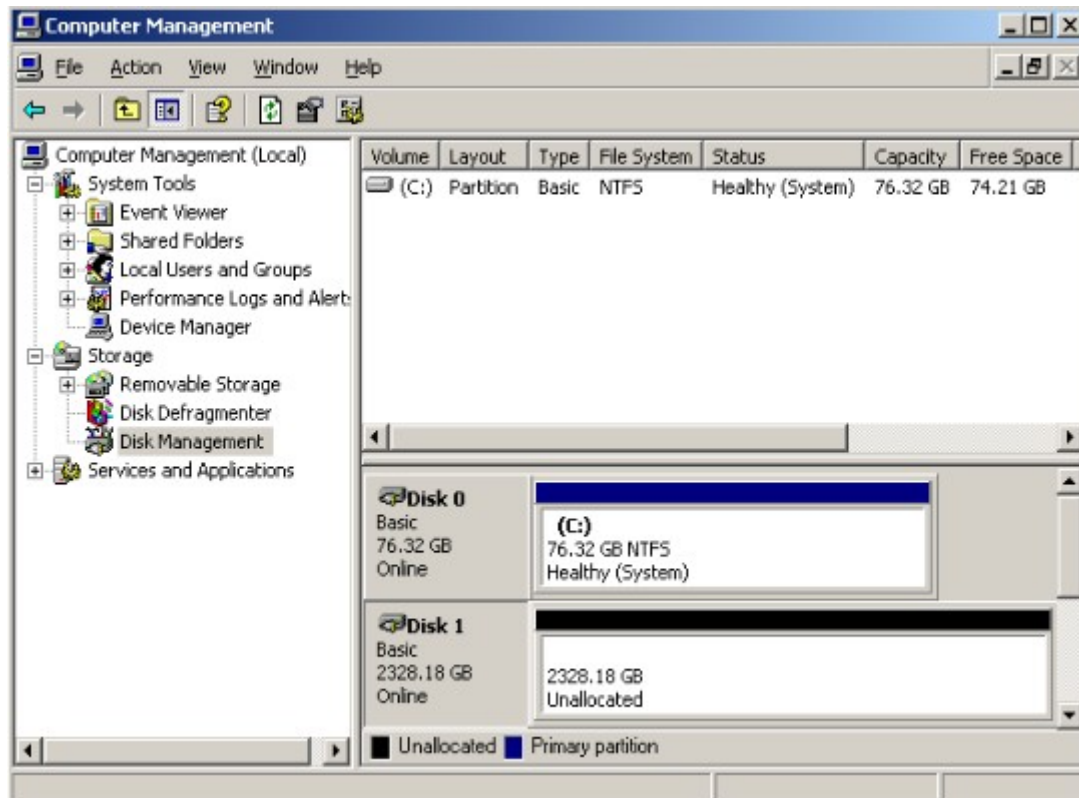
1.3 After initialize this disk, you will able to see a device divided to 2 parts : a 2TB space and the rest :



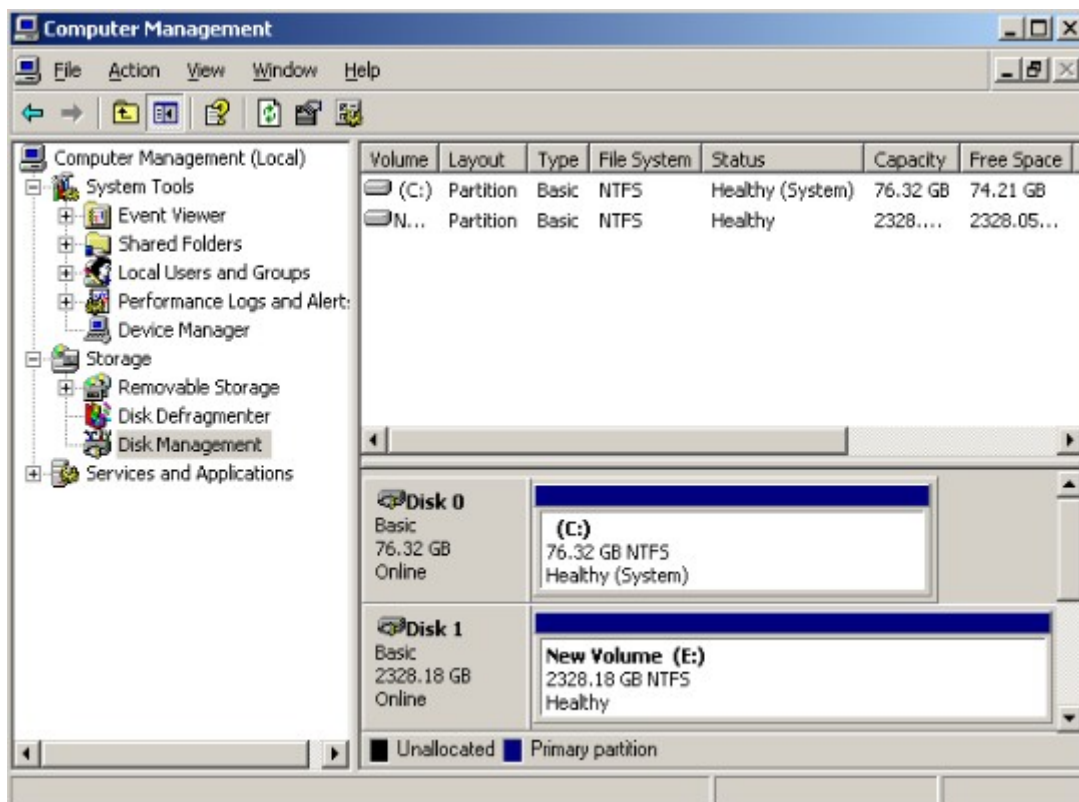
1.4 If you want to configure a partition with greater than 2TB capacity, you need convert this disk to GPT Disk. :



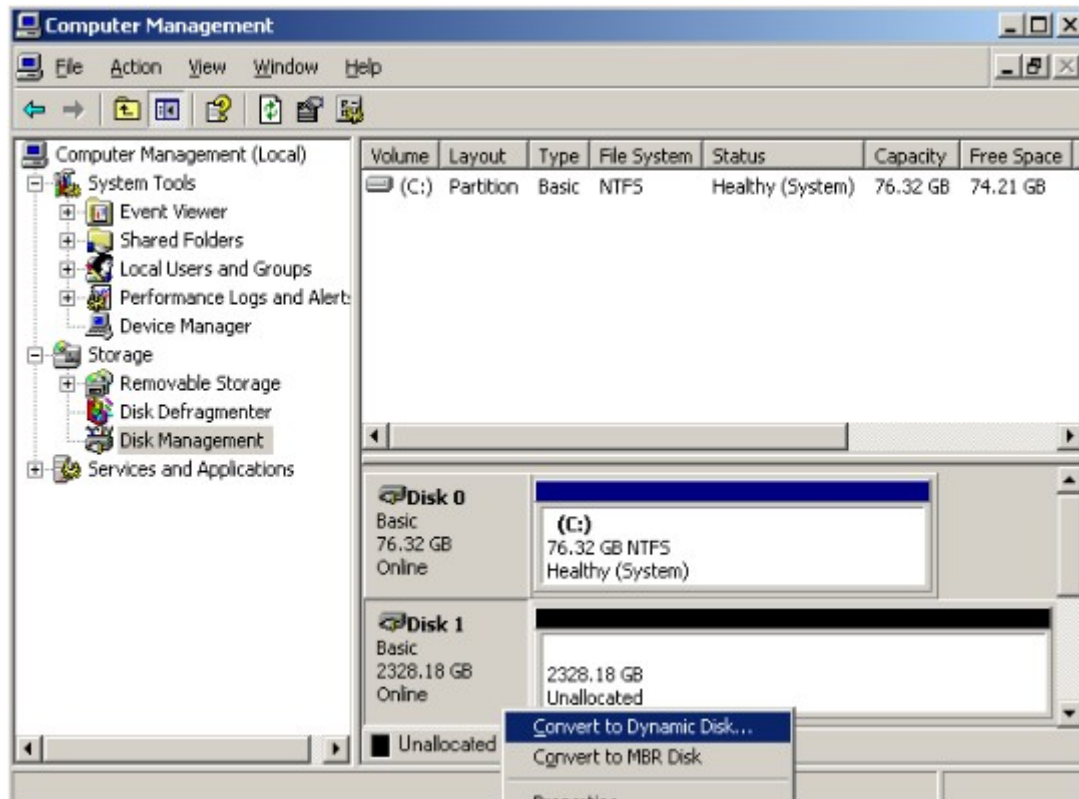
1.5 After you convert it to GPT Disk, the device will become a single disk with full capacity.



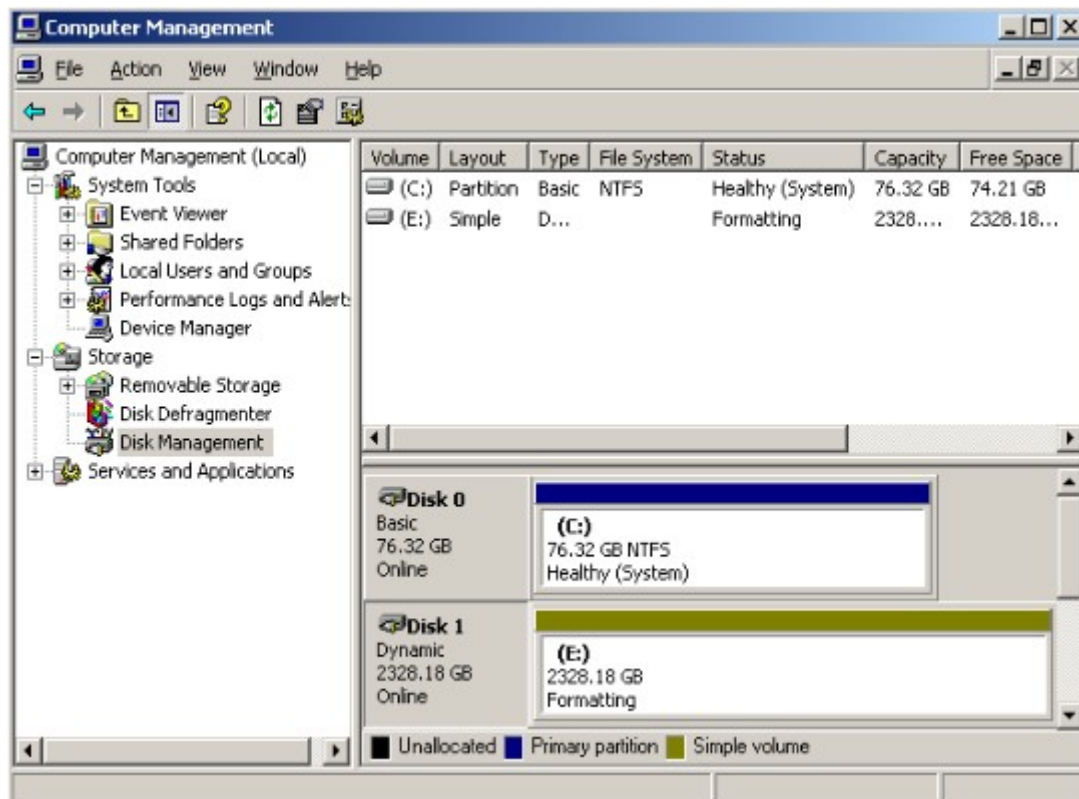
1.6 Now, you can create a partition with greater than 2TB capacity.



1.7 If you need, you can also convert this volume to a **Dynamic Disk** before partition it :



1.8 After convert to **Dynamic Disk**, you can create a partition with greater than 2TB capacity.



Note : Windows 2000 w/SP4 and Windows XP w/SP2 NOT support LBA64 yet. So you will not able to use LBA64 Volume with them.

Using greater than 2TB volume in Linux by “LBA 64” option

Notice : Please make sure that your system have ‘Large Block Device’ support enabled.

To enable it, open a Terminal Command console, then type :

```
# cd /usr/src/linux2.6
```

```
# make menuconfig
```

A configuration menu will open on the screen. Select ‘Device Drivers’ > ‘Block Devices’. There has a item called ‘Large Block Device’. Put an * on it. Save and exit. Then type :

```
# make
```

```
# make modules
```

```
# make modules_install
```

```
# install
```

When it is all done, reboot system. You will find a new option for custom kernel in menu. Boot from this new kernel.

To make a new kernel support ‘Large Block Device’, please make sure your system already installed kernel source and compiler tools.

1. Using “LBA 64” function to support greater 2TB volume in Linux Kernel 2.6.x

You can use “parted” command to partition a greater than 2TB device, for example :
(using a 2.5TB volume)

Open a Terminal Command console, then type :

```
# parted /dev/sda
```

GNU Parted 1.6.9

Copyright (C) 1998, 1999, 2000, 2001, 2002, 2003 Free Software Foundation, Inc.

This program is free software, covered by the GNU General Public License.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

Using /dev/sda

Information: The operating system thinks the geometry on /dev/sda is 303940/255/63. Therefore, cylinder 1024 ends at 8032.499M.

(parted)

In parted prompt type :

```
(parted) mklabel gpt
```


This command will create a gpt partition label.

Then type :

(parted) p

Disk geometry for /dev/sda: 0.000-2384185.000 megabytes

Disk label type: gpt

<i>Minor</i>	<i>Start</i>	<i>End</i>	<i>Filesystem</i>	<i>Name</i>	<i>Flags</i>
--------------	--------------	------------	-------------------	-------------	--------------

The disk geometry and partition table will be displayed.

Next, type “mkpart” command to make a partition :

(parted) mkpart

Partition type? [primary]?

File system type? [ext2]?

Start? 0

End? 2384185

After that, type “q” to exit parted :

(parted) q

Information: Don't forget to update /etc/fstab, if necessary.

The partition is created, now you can use mkfs to format it.:

mkfs.ext3 /dev/sda1

mke2fs 1.35 (28-Feb-2004)

.....

.....

.....

Now you can mount this partition and use it :

mount /dev/sda1 /mnt

df

<i>Filesystem</i>	<i>1K-blocks</i>	<i>Used</i>	<i>Available</i>	<i>Use%</i>	<i>Mounted on</i>
<i>/dev/hda2</i>	<i>77649156</i>	<i>6204996</i>	<i>67499804</i>	<i>9%</i>	<i>/</i>
<i>/dev/hda1</i>	<i>101086</i>	<i>10954</i>	<i>84913</i>	<i>12%</i>	<i>/boot</i>
<i>none</i>	<i>256940</i>	<i>0</i>	<i>256940</i>	<i>0%</i>	<i>/dev/shm</i>
<i>/dev/sda1</i>	<i>2403095944</i>	<i>32828</i>	<i>2280992848</i>	<i>1%</i>	<i>/mnt</i>

Notice : When you display the partition table in parted, the End value is incorrect, but the Disk geometry value is correct. For example :

(parted) p

Disk geometry for /dev/sda: 0.000-2384185.000 megabytes

Disk label type: gpt

<i>Minor</i>	<i>Start</i>	<i>End</i>	<i>Filesystem</i>	<i>Name</i>	<i>Flags</i>
<i>1</i>	<i>0.017</i>	<i>287032.983</i>			

The Disk geometry value “2384185: is correct for megabytes, but the End value “287032.983” is incorrect.
2,500,000,000,000 bytes = 2,441,406,250 kbytes = 2,384,185 megabytes (divided by 1024)

Something about file system in Linux

A. file system limitation :

file system	filesystem limitation	file size limitation	Block size	
ext2	4TB	2GB~2TB *1	1KB~4KB	
ext3	4TB	2GB~2TB *1	1KB~4KB	
reiserfs	16TB	8TB	4KB	

*1 The file size limitation depends on the block size you used. If you use 4KB block size, the file size limitation will be 2TB.

Using greater than 2TB volume in FreeBSD by “LBA 64” option

1. Using “LBA 64” function to support greater 2TB volume in FreeBSD 5.2.1 and later

You can use “gpt” command to partition a greater than 2TB device, for example :
(using a 2.375TB volume)

Open a Console, then type :

```
# gpt create -f /dev/da1
```

This command will create a gpt partition table.

Then you can check the table by type in :

```
# gpt show /dev/da1
```

<i>start</i>	<i>size</i>	<i>index</i>	<i>contents</i>
0		1	PMBR
1		1	Pri GPT header
2		32	Pri GPT table
34	4638668733		
4638668767		32	Sec GPT table
4638668799		1	Sec GPT header

The partition table will be displayed.

Next, type “gpt addt” command to make a partition :

```
# gpt add /dev/da1
```

This command will create a gpt partition (the default file system is UFS).

Then you can check the partition table by type in :

```
# gpt show /dev/da1
```

<i>start</i>	<i>size</i>	<i>index</i>	<i>contents</i>
0		1	PMBR
1		1	Pri GPT header
2		32	Pri GPT table
34	4638668733	1	GPT part - FreeBSD UFS/UFS2
4638668767		32	Sec GPT table
4638668799		1	Sec GPT header

The partition is created, now you can use newfs to format it.:

newfs /dev/da1p1

/dev/da1p1: 2264975.0MB (4638668732 sectors) block size 16384, fragment size 2048

using 12326 cylinder groups of 183.77MB, 11761 blks, 23552 inodes.

super-block backups (for fsck -b #) at:

*160, 376512, 752864, 1129216, 1505568, 1881920, 2258272, 2634624, 3010976,
3387328, 3763680, 4140032, 4516384, 4892736, 5269088, 5645440, 6021792,
6398144, 6774496, 7150848, 7527200, 7903552, 8279904, 8656256, 9032608,*

... ..

... ..

... ..

*4633645984, 4634022336, 4634398688, 4634775040, 4635151392, 4635527744,
4635904096, 4636280448, 4636656800, 4637033152, 4637409504, 4637785856,
4638162208, 4638538560*

Now you can mount this partition and use it :

mount /dev/da1p1 /mnt

df -l

<i>Filesystem</i>	<i>1K-blocks</i>	<i>Used</i>	<i>Avail</i>	<i>Capacity</i>	<i>Mounted on</i>
<i>/dev/ad0s1a</i>	<i>253678</i>	<i>35870</i>	<i>197514</i>	<i>15%</i>	<i>/</i>
<i>devfs</i>	<i>1</i>	<i>1</i>	<i>0</i>	<i>100%</i>	<i>/dev</i>
<i>/dev/ad0s1e</i>	<i>253678</i>	<i>12</i>	<i>233372</i>	<i>0%</i>	<i>/tmp</i>
<i>/dev/ad0s1f</i>	<i>36090964</i>	<i>1295058</i>	<i>31908630</i>	<i>4%</i>	<i>/usr</i>
<i>/dev/ad0s1d</i>	<i>253678</i>	<i>4206</i>	<i>229178</i>	<i>2%</i>	<i>/var</i>
<i>/dev/da1p1</i>	<i>2246364172</i>	<i>4 2066655036</i>	<i>0%</i>	<i>/mnt</i>	