MSSQLTips.com
brought to you by Edgewood Solutions

Your daily source for SQL Server Tips

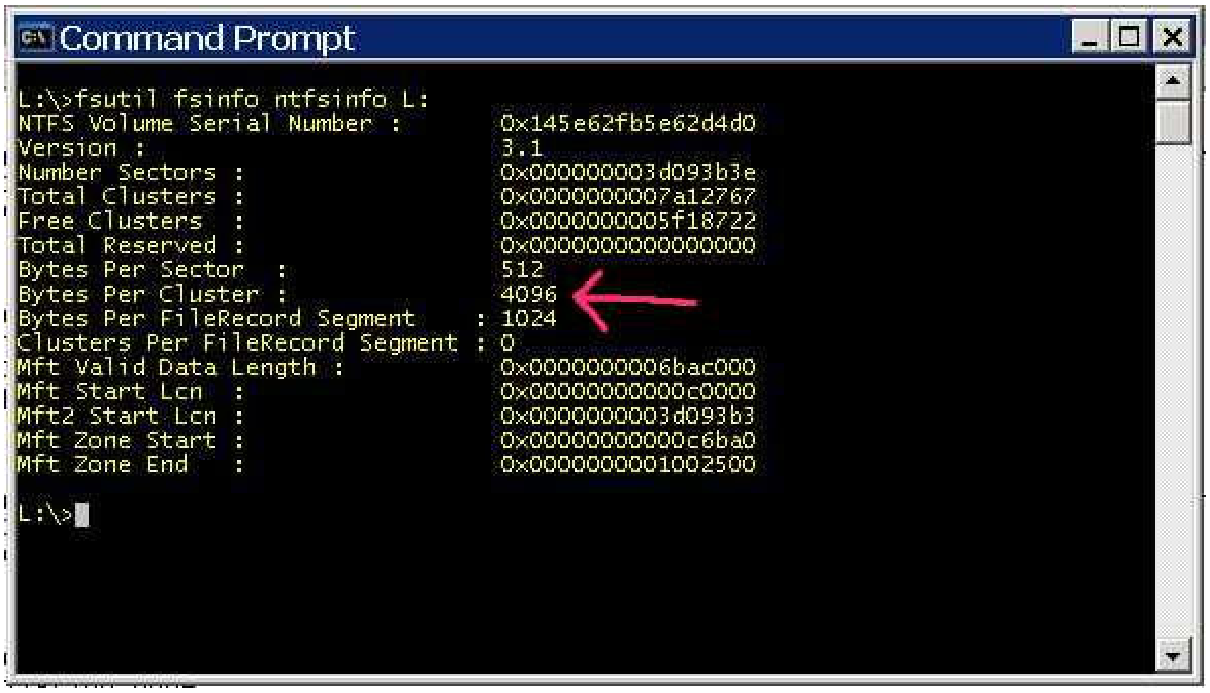
Format drives with correct allocation and offset for maximum SQL Server performance

Written By: Andy Novick -- 10/11/2010

Problem
Disk performance is critical to the performance of SQL Server. Creating partitions with the correct offset and formatting drives with the correct allocation unit size is essential to getting the most out of the drives that you have. I've always been told that the drive's partition offset must be set to 32K and the allocation unit size set to 64K for partitions that hold data and 8K for partitions that hold logs. How does one set these parameters correctly?

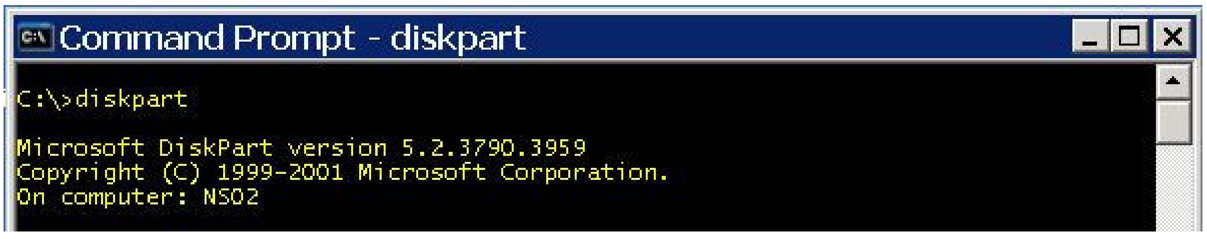
Solution
In the article Partition offset and allocation unit size of a disk for SQL Server, I showed how to get both the partition offset and allocation unit size. Allocation unit size is also know as cluster size. In this article I'll show you how to set them according to best practices. The configuration that I'll use is suggested by Microsoft. It's a starting point and each disk system should be tested to verify that optimal performance is achieved. The article Benchmarking SQL Server IO with SQLIO shows how to get started on benchmarking.

The drive that I'm going to be working with is my Disk 1. It's a set of SATA drives directly attached to my server and bound into a hardware (controller) RAID 5 set. Let's take a look at the allocation unit size before changing it:

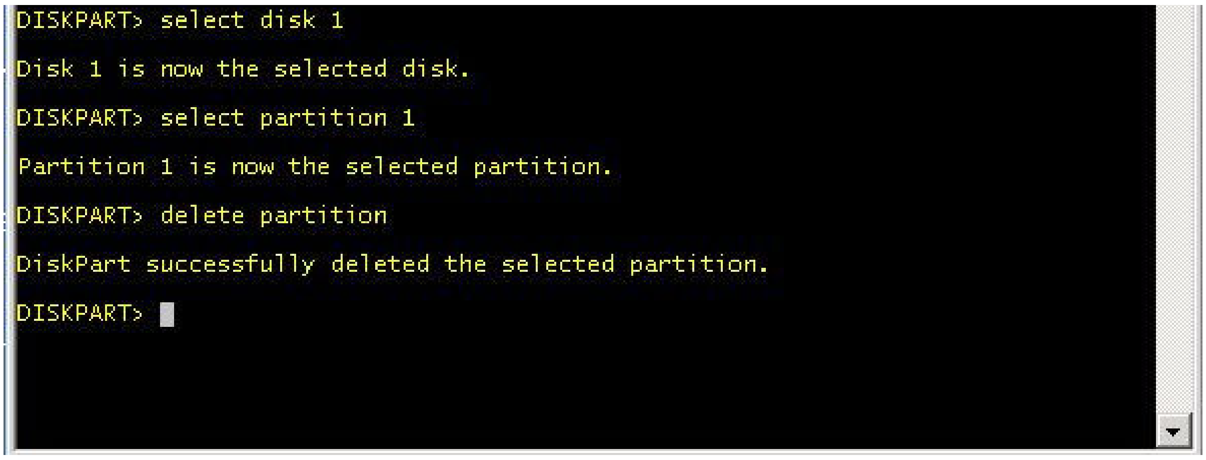


```
L:\>fsutil fsinfo ntfsinfo L:
NTFS Volume Serial Number :       0x145e62fb5e62d4d0
Version :                         3.1
Number Sectors :                  0x000000003d093b3e
Total Clusters :                  0x0000000007a12767
Free Clusters  :                  0x0000000005f18722
Total Reserved :                  0x0000000000000000
Bytes Per Sector  :               512
Bytes Per Cluster :               4096
Bytes Per FileRecord Segment    : 1024
Clusters Per FileRecord Segment : 0
Mft Valid Data Length :           0x0000000006bac000
Mft Start Lcn  :                  0x00000000000c0000
Mft2 Start Lcn :                  0x0000000003d093b3
Mft Zone Start :                  0x00000000000c6ba0
Mft Zone End   :                  0x0000000001002500

L:\>
```

Before creating the partition, the original has to be deleted of course, after you've backed up or moved any data. The Computer Management snap-in for MMC can do that or it can be done with the DISKPART tool. Here the DISKPART command is used to delete partition 1 from disk 1. This was the L: drive.

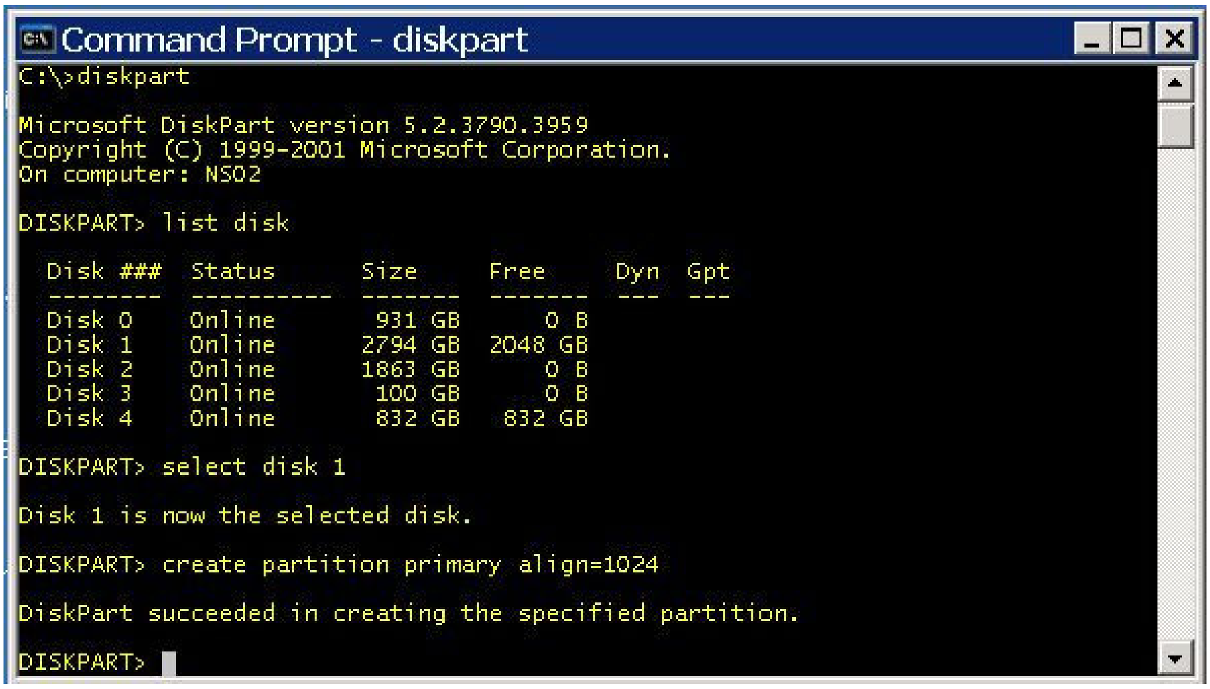


```
C:\>diskpart

Microsoft DiskPart version 5.2.3790.3959
Copyright (C) 1999-2001 Microsoft Corporation.
On computer: NS02
```



```
DISKPART> select disk 1

Disk 1 is now the selected disk.

DISKPART> select partition 1

Partition 1 is now the selected partition.

DISKPART> delete partition

DiskPart successfully deleted the selected partition.

DISKPART>
```

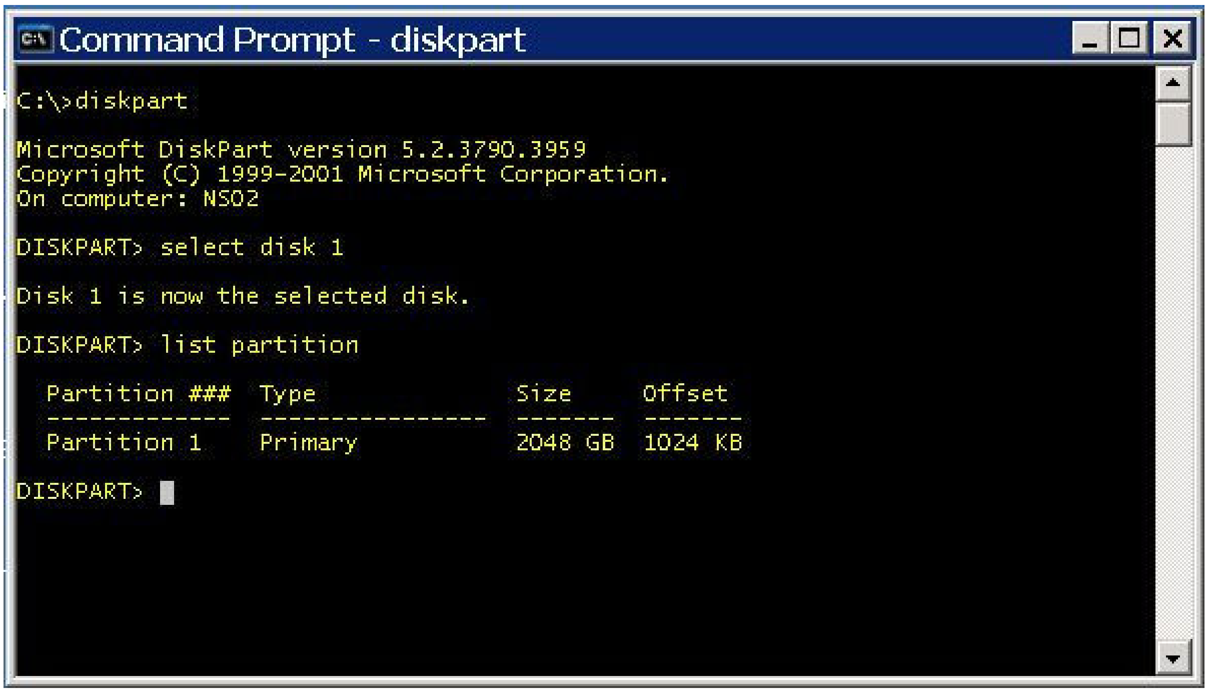
Now it's time to create the new partition. In Windows 2008 Microsoft changed the default partition offset to 1024K. This number is supposed to align well with RAID arrays and SANS. I'll use 1024K instead of 32K.



Command Prompt - diskpart

```
C:\>diskpart

Microsoft DiskPart version 5.2.3790.3959
Copyright (C) 1999-2001 Microsoft Corporation.
On computer: NS02

DISKPART> list disk

  Disk ###  Status      Size     Free     Dyn  Gpt
  --------  ----------  -------  -------  ---  ---
  Disk 0    Online       931 GB      0 B
  Disk 1    Online      2794 GB  2048 GB
  Disk 2    Online      1863 GB      0 B
  Disk 3    Online       100 GB      0 B
  Disk 4    Online       832 GB   832 GB

DISKPART> select disk 1

Disk 1 is now the selected disk.

DISKPART> create partition primary align=1024

DiskPart succeeded in creating the specified partition.

DISKPART>
```

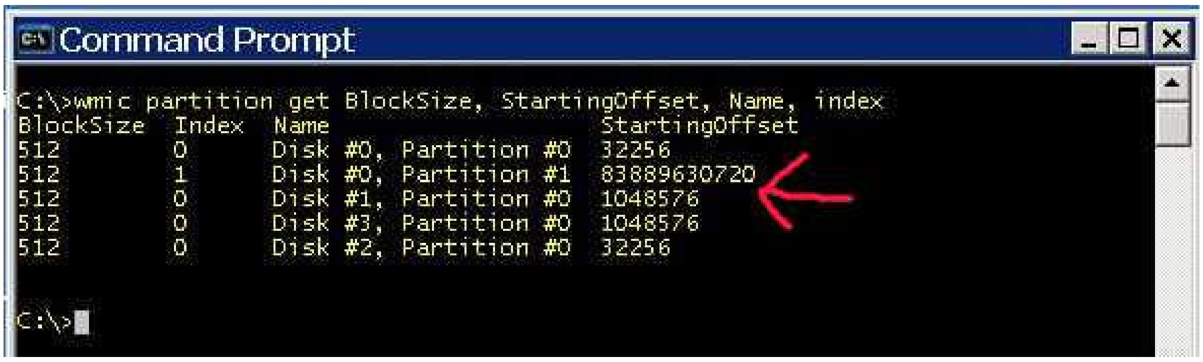
DISKPART shows the partition at the 1024 KB offset:



Command Prompt - diskpart

```
C:\>diskpart

Microsoft DiskPart version 5.2.3790.3959
Copyright (C) 1999-2001 Microsoft Corporation.
On computer: NS02

DISKPART> select disk 1

Disk 1 is now the selected disk.

DISKPART> list partition

  Partition ###  Type              Size     Offset
  -------------  ----------------  -------  -------
  Partition 1    Primary           2048 GB  1024 KB

DISKPART>
```

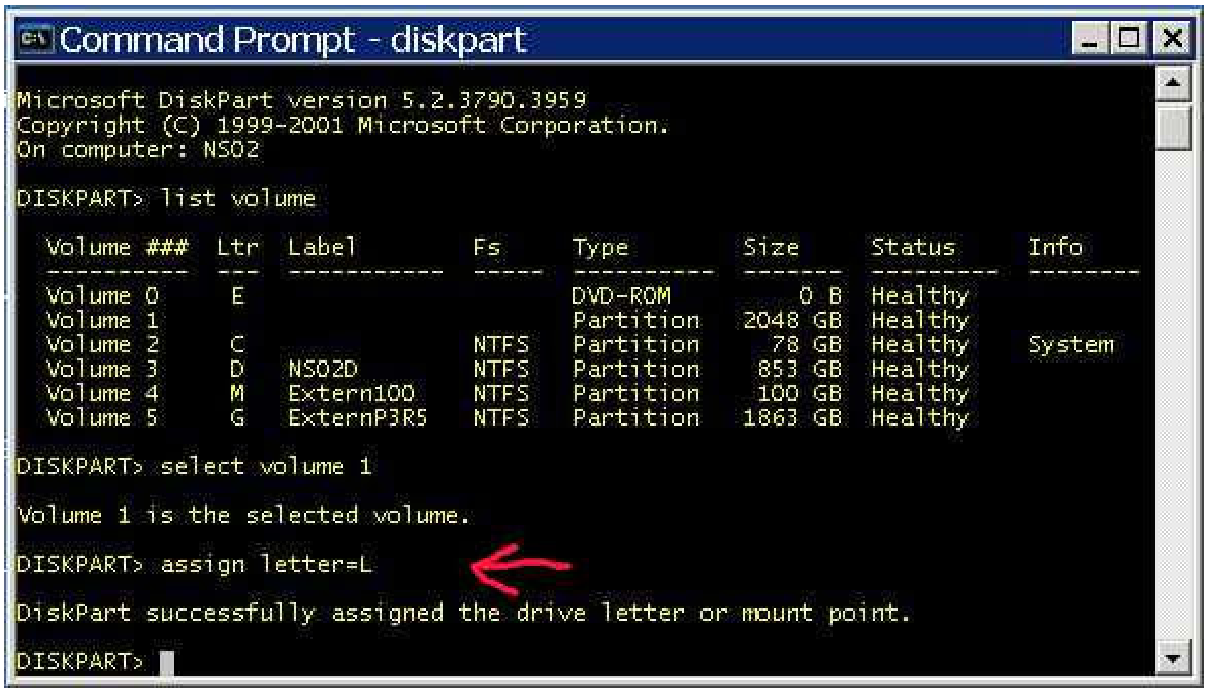
However, since DISKPART rounds the offset, the correct tool to use is WMIC, specifically the query "**wmic partition get BlockSize, StartingOffset, Name, Index**", shown here:



```
C:\>wmic partition get BlockSize, StartingOffset, Name, index
BlockSize  Index  Name                  StartingOffset
512        0      Disk #0, Partition #0  32256
512        1      Disk #0, Partition #1  83889630720
512        0      Disk #1, Partition #0  1048576
512        0      Disk #3, Partition #0  1048576
512        0      Disk #2, Partition #0  32256

C:\>
```

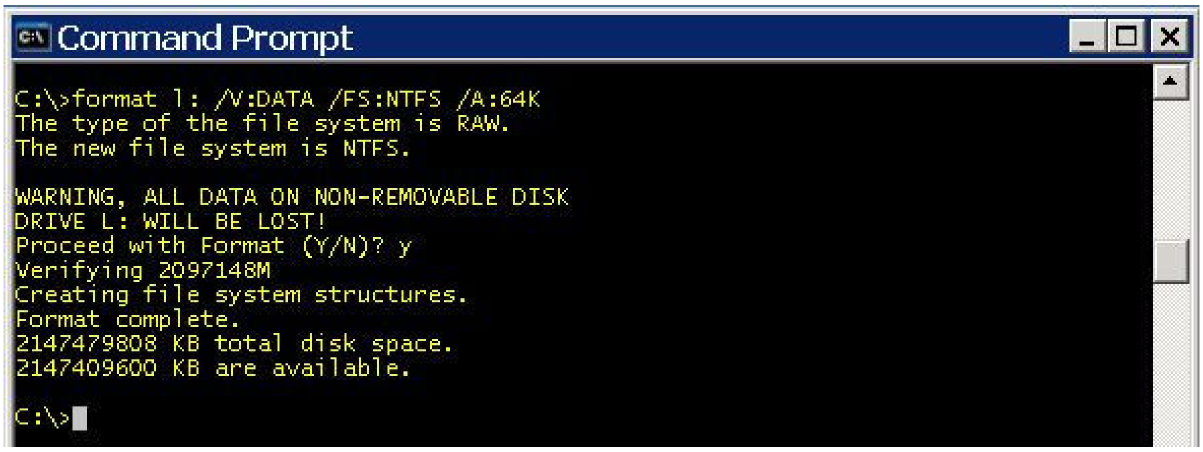
The value 1048576 is exactly one megabyte and is the proper alignment for most purposes most of the time. Other hardware, such as SANS, might need a different alignment and you'll have to consult the vendor about what's best for their hardware.

Next assign a drive letter to the volume with DISKPART. DISKPART's "list volume" subcommand first shows us the available volumes. The new volume is #1. This is selected and then assigned the letter L.



```
Microsoft DiskPart version 5.2.3790.3959
Copyright (C) 1999-2001 Microsoft Corporation.
On computer: NS02

DISKPART> list volume

  Volume ###  Ltr  Label        Fs     Type        Size     Status     Info
  ----------  ---  -----------  -----  ----------  -------  ---------  --------
  Volume 0     E                       DVD-ROM        0 B   Healthy
  Volume 1                             Partition   2048 GB  Healthy
  Volume 2     C                NTFS   Partition     78 GB  Healthy    System
  Volume 3     D   NS02D        NTFS   Partition    853 GB  Healthy
  Volume 4     M   Extern100    NTFS   Partition    100 GB  Healthy
  Volume 5     G   ExternP3R5   NTFS   Partition   1863 GB  Healthy

DISKPART> select volume 1

Volume 1 is the selected volume.

DISKPART> assign letter=L

DiskPart successfully assigned the drive letter or mount point.

DISKPART>
```

Finally, format the drive with the desired allocation unit size of 64 kilobytes. Of course, the file system is NTFS. I use DATA as the volume name because L: is going to be a data drive. Here's the output from the Format:



```
C:\>format l: /V:DATA /FS:NTFS /A:64K
The type of the file system is RAW.
The new file system is NTFS.

WARNING, ALL DATA ON NON-REMOVABLE DISK
DRIVE L: WILL BE LOST!
Proceed with Format (Y/N)? y
Verifying 2097148M
Creating file system structures.
Format complete.
2147479808 KB total disk space.
2147409600 KB are available.

C:\>
```

The L: drive now has the offset and formatting that I want. Has the performance improved? Knowing that, is going to take running SQLIO on the drive.

One warning. When you format without the /Q switch Windows zeros the blocks on the drive, which can take a long time. Add the /Q switch if you don't want to wait.

Next Steps

- Correct the partition offset and allocation unit size of drives where performance might be improved.
- Repeat the benchmarking process on the drives to verify that the changes have the desired effect.