
Microsoft Windows File-System Alignment

An EMC Best Practice for Microsoft Windows Servers
on Symmetrix and CLARiiON

Abstract

Default installation and configuration of Windows Servers result in misaligned file systems on EMC storage arrays. The misalignment with EMC Symmetrix and CLARiiON internal architectures can impact optimal performance. This white paper describes this issue and details the procedure to create aligned file systems on the Microsoft Windows Server platform.

Published 4/29/2005

Copyright © 2005 EMC Corporation. All rights reserved.

EMC believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED “AS IS.” EMC CORPORATION MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Use, copying, and distribution of any EMC software described in this publication requires an applicable software license.

EMC², EMC, ApplicationXtender, Celerra, CentraStar, CLARiiON, CLARAlert, Connectrix, Dantz, Direct Matrix Architecture, DiskXtender, Documentum, EmailXtender, EmailXtract, HighRoad, Legato, Navisphere, RepliStor, PowerPath, ResourcePak, Retrospect, Smarts, SnapView/IP, SRDF, Symmetrix, TimeFinder, VisualSAN, and where information lives are registered trademarks and EMC ControlCenter, EMC Developers Program, EMC OnCourse, EMC Proven, EMC Snap, EMC Storage Administrator, Access Logix, ArchiveXtender, Automated Resource Manager, AutoSwap, AVALONidm, C-Clip, Celerra Replicator, Centera, CLARevent, CopyCross, CopyPoint, DatabaseXtender, Direct Matrix, DiskXtender 2000, EDM, E-Lab, EmailXaminer, Enginuity, eRoom, FarPoint, FLARE, InfoMover, MirrorView, NetWin, NetWorker, OnAlert, OpenScale, Powerlink, RepliCare, SafeLine, SAN Advisor, SAN Copy, SAN Manager, SDMS, SnapSure, SnapView, StorageScope, SupportMate, SymmAPI, SymmEnabler, Symmetrix DMX, and VisualSRM are trademarks of EMC Corporation. All other trademarks used herein are the property of their respective owners.

Part Number H1531

Table of Contents

Introduction	4
Viewing Partition Misalignment	5
Why this Presents an Issue with EMC Arrays	6
RAID 5 Characteristics of Symmetrix Storage	8
Fixing the Issue by Aligning Windows File Systems	9
Basic Disks	9
Initializing RAW Disk Devices	9
Using DISKPAR.EXE	9
Creating Additional Volumes on a Basic Disk	12
Dynamic Disks	12
Creating a Filler Partition to Force Alignment	12
Verifying Alignment with Dynamic Disks	14
Creating Additional Volumes on a Dynamic Disk	16
Creating the Windows File System	17
References	19
Default Cluster (Allocation Unit) Size for FAT and NTFS	19
Microsoft SQL Server NTFS Allocation Unit Size	19
Dynamic Disks and Volumes	19

Introduction

Current versions of Microsoft Windows Server do not correctly detect disk-drive geometry of devices presented from storage area network (SAN) arrays such as EMC® Symmetrix® and CLARiiON®. The detection results in a geometry of 63 sectors per track, whereas the correct form of geometry should be 64 sectors per track. Creating partitions and the volumes that they represent without modification will result in suboptimal I/O processes.

It is the intent of this white paper to present supportable methods of resolving misalignment for Windows Servers by creating partitions that are aligned on 64 KB boundaries.

This white paper deals with current versions of NTFS on Windows Server 32-bit environments. It does not cover specific recommendations for 64-bit environments with MBR or GTP formats. Additional recommendations for Windows Server 2003 SP1 will be provided in a subsequent update to this document.

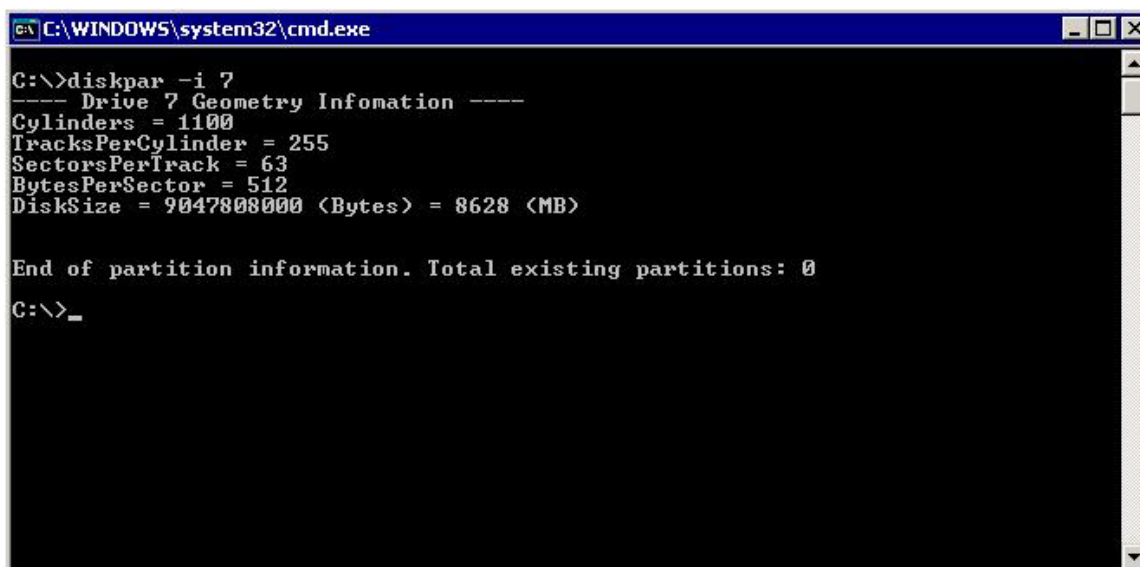
In shared storage arrays, inefficient utilization of storage components may not only adversely affect the performance of the host generating the I/O, but also other hosts that share common components. Optimally configured arrays ensure that all components are efficiently used.

Misaligned Windows file systems may result in significant performance degradation. It is not uncommon in a Symmetrix environment to see a performance improvement of 20 percent or more (depending on layout and I/O load). As a result of resolving misaligned I/O and its overheads, a more efficient utilization of system resources will occur. This can translate into positive improvements in I/O response for other servers on the shared array.

Throughout this white paper, the terms *partition* and *volume* are used. Partitions are logical allocations of storage on a disk or LUN. A volume is a formatted partition that can be subsequently mounted at a drive letter or mount point, as applicable.

Viewing Partition Misalignment

You can use commonly available disk tools to detail the detected geometry. In Figure 1, the Microsoft Windows 2000 Resource Kit utility DISKPAR.EXE is used to list the details of the newly presented disk device.



```
C:\WINDOWS\system32\cmd.exe
C:\>diskpart -i 7
----- Drive 7 Geometry Information -----
Cylinders = 1100
TracksPerCylinder = 255
SectorsPerTrack = 63
BytesPerSector = 512
DiskSize = 9047808000 <Bytes> = 8628 <MB>

End of partition information. Total existing partitions: 0
C:\>_
```

Figure 1. Symmetrix Device Detected Geometry

File systems created on this incorrectly detected geometry will result in misalignment with respect to the internal Symmetrix and CLARiiON architectures. During the creation of a partition, Windows will allocate a system area for the Master Boot Record (MBR), which contains partition information; by default, this space is padded out to the size of a single track. In this case, the track is calculated based on the geometry of the drive.

By default, the creation of a partition will calculate the reserved system area to be 1 x track x sector size. In the case of a Symmetrix or CLARiiON device, this will be 1 x 63 x 512 = 32256 bytes. This number is not truly 32 KB (32768 bytes). The calculated value is actually 512 bytes less than 32 KB, and results in misalignment issues on the storage arrays.

Figure 2 demonstrates this misalignment effect, again using the DISKPAR utility to display the same disk device as in Figure 1. In this case, Windows Disk Management had been used to create a single Basic volume using the entire available space. Note the starting offset (shown as `StartingOffset` in DISKPAR) is 32256 as discussed, resulting in misalignment.

```

C:\>diskpart -i 7
---- Drive 7 Geometry Information ----
Cylinders = 1100
TracksPerCylinder = 255
SectorsPerTrack = 63
BytesPerSector = 512
DiskSize = 9047808000 <Bytes> = 8628 <MB>

---- Drive Partition 0 Information ----
StartingOffset = 32256
PartitionLength = 9047775744
HiddenSectors = 63
PartitionNumber = 1
PartitionType = 7

End of partition information. Total existing partitions: 1
C:\>_

```

Figure 2. Partition Information after Default Creation with Windows Disk Manager

Why this Presents an Issue with EMC Arrays

Generally, Symmetrix storage arrays use a cache slot size of 32 KB (32768 bytes). CLARiiON storage arrays use a configurable element size, which by default is 128 sectors (65536 bytes)—this value is set at LUN creation and cannot be changed. CLARiiON arrays do have a capability to internally compensate for the geometry issue described in this document; the CLARiiON LUN offset is discussed later in this document. Either the LUN offset should be used, or the corrective action of partition alignment should be made, but *not* both.

The Windows partition is a logical representation on top of the virtualized device provided by the Symmetrix or CLARiiON array. This virtualized device (typically referred to as a *disk*) is viewed by the Windows Server as a logical unit, commonly referred to as a LUN.

All input/output (I/O) operations can essentially be considered to do a reference to that LUN with an offset. Thus a single page of data exists on the partition defined on the LUN and is further indirected at an offset to that partition. I/O activity will typically occur at a size equivalent to the allocation unit size defined by the NTFS volume on the partition; the most common default value for that page size is 4 KB (4096 bytes). Microsoft Windows Servers can dynamically select an allocation unit size for the volume being formatted, although administrators can override this default option. Refer to Microsoft Knowledge Base article 140365 for additional information on the default settings.

If we represent the Symmetrix device as being defined by a range of consecutive 32 KB (32768byte) tracks, and correlated that to the default partition created previously in our example, you would see a representation as follows.

While the following examples illustrate the issues from a Symmetrix perspective, a similar issue exists with CLARiiON structures. The applicability of the resolution is for both storage arrays.

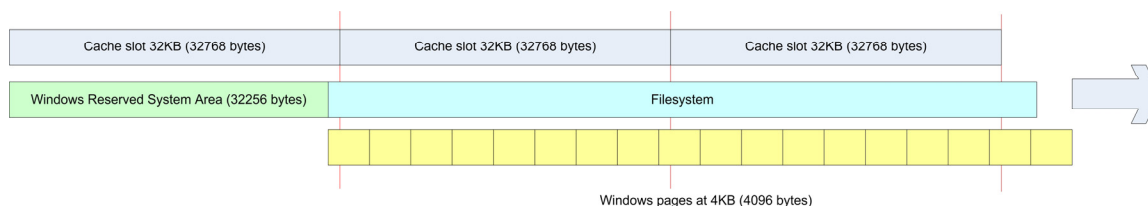


Figure 3. Representation of Symmetrix Cache to Windows File System and 4 KB Pages

In Figure 3, the reserved area creates a situation where the file-system usable space begins at a location which is 512 bytes offset from the beginning of a logical cache slot. This creates an inefficient situation for internal transfers for this data. A similar issue exists with internal CLARiiON structures where host file-system structures result in misalignment internally within the CLARiiON array. In both cases, multiple disk transfers must be executed rather than a single transfer. Note that is a result of the virtualized view that the host has of the object it sees as a LUN. The inefficiencies are caused by spanning structures within the arrays (such as cache slots) and, when the virtualized device is striped within the array, crossing of stripe boundaries.

Additionally Symmetrix arrays maintain data in the form of sectors. Symmetrix sectors are 4 KB (4096 bytes) in size. This means that a cache slot may actually be thought of as represented by eight of these sectors.

Removing the partition component from our example, and introducing sectors, we can compare how the pages within the Windows file system relate to Symmetrix cache slots.

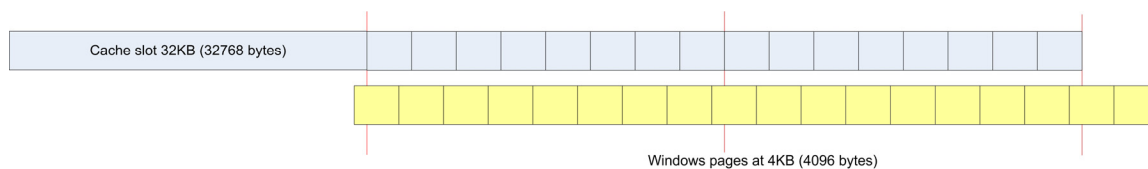


Figure 4. Representation of 4 KB Windows Pages to Symmetrix Sectors

It is at this page level where it can be seen that there are severe alignment issues. Each logical Windows page is offset by the same amount (512 bytes) from the Symmetrix sector. As a result, each Windows page is sector-misaligned. Some of those pages are also considered track-misaligned, where they cross a cache slot. This level of misalignment and random I/O behavior from the host can cause the array to process multiple back-end transfers to service a single I/O request.

Looking at the same situation with a Windows allocation unit size of 8 KB (8192 bytes), the problem may be considered reduced to the extent that there are fewer Windows pages, but the inherent problem persists. This is true of all larger supported allocation unit sizes since whenever they start at a misaligned sector location, all subsequent pages will be misaligned.

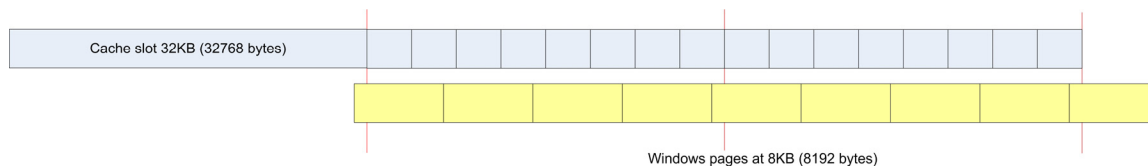


Figure 5. Representation of 8 KB Windows Pages to Symmetrix Sectors

It is clear in Figure 5 that a single Windows 8 KB (8192 bytes) page includes a partial sector, one full sector, and then another partial sector.

Misaligned file-system I/O operations can cause additional workloads for Symmetrix components, including fibre adapters (FAs), and subsequently provide inefficient utilization of other components including, but not limited to, internal buses and disk resources. For both Symmetrix and CLARiiON arrays, the additional workload not only affects the Windows host itself, but other hosts that share the same

components. It is in the best interest of all connected hosts that alignment for Windows servers be carried out.

While some inefficiencies can be dealt with by internal Symmetrix and CLARiiON mechanisms, it is much more effective to remove this condition.

The ultimate goal is to create partitions—and the volumes that they represent—to be aligned to storage-array structures such that the logical alignment is correct across all components.

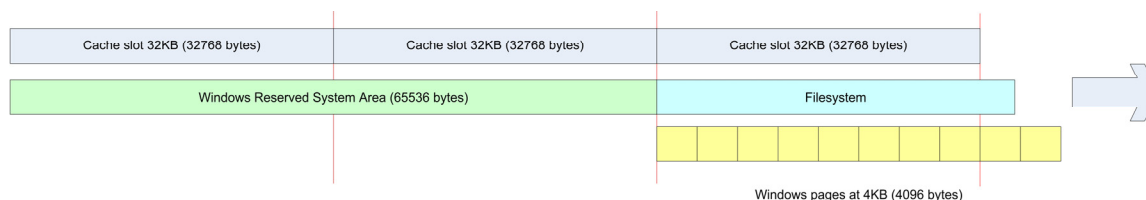


Figure 6. File-System Structures Aligned to Storage-Array Components

RAID 5 Characteristics of Symmetrix Storage

With the advent of RAID 5 support with Symmetrix storage, there is an additional concern regarding alignment. RAID 5 configurations result in a slightly different form of striping at the RAID 5 disk-group level. Within a RAID 5 set, striping occurs at four-track intervals. A Symmetrix track, as previously indicated, is 32 KB (35536 bytes), thus a *stripe* for a RAID 5 group is 128 KB.

Ensuring that any I/O operations are aligned within this stripe depth will optimize I/O operations for a RAID 5 set. Subsequently, ensuring that the partition alignment is done to a 64 KB offset ensures that all I/O operations are aligned with the RAID 5 stripe.

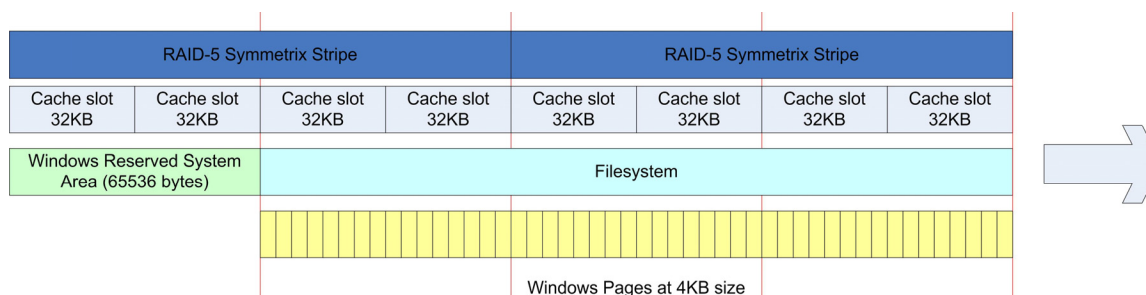


Figure 7. Symmetrix RAID 5 Stripe Depth and Windows Alignment

Misalignment within a given RAID 5 stripe would result in multiple back-end access across RAID 5 members. Correct alignment ensures optimal back-end utilization.

Fixing the Issue by Aligning Windows File Systems

Tools and methods other than the default Windows Disk Manager must be used to create aligned file systems on Windows Server 2000 and Windows Server 2003.

When a raw disk is first detected by Windows Servers, it must be initialized. This function must first be carried out by Windows Disk Manager. You must do this as normal, by using the Disk Management snap-in to write a signature to the disk device (LUN) itself. Once initialized, you may optionally configure the new disk device to be either a Basic disk or a Dynamic disk.

Basic disks and Dynamic disks differ in several ways. The major difference is the ability of Dynamic disks to support spanned or striped disk sets. This allows the system to aggregate the available volume space into a single logical volume, and thus create a form of RAID device. Typically, the advantages of host-based logical volumes are mitigated by the functionality provided by the array itself, thereby leaving the system to service the applications rather than maintaining RAID configurations.

Windows does not support the use of Dynamic disks created from the provided Logical Volume Manager (LVM) in a Windows Cluster Service environment. A third-party logical volume manager is required for this purpose. Additionally, several EMC software products do not support being used against volume sets created by the base LVM.

The following sections provide guidance on how to create aligned file systems on both Basic and Dynamic disks.

Basic Disks

The recommended tool for this purpose is the previously mentioned DISKPAR.EXE tool from the Microsoft Windows 2000 Resource Kit. Since there is no equivalent tool currently available within the Windows 2003 product set or its Resource Kit, the Windows 2000 version of DISKPAR.EXE must be used. Testing has shown that DISKPAR.EXE is fully compatible with the Windows Server 2003 platform.

Microsoft has indicated that within the Service Pack 1 (SP1) release of Windows 2003, the DISKPART.EXE tool, will allow for this functionality. It is not available in the DISKPART.EXE available with the base release of the operating system.

Initializing RAW Disk Devices

When first presented to a Windows Server, a physical disk or LUN must be initialized. Windows writes data to the disk to identify it—this includes writing a Disk Signature to the device. This step must be performed before any partitions are created on the device itself. After the initialization, disks are presented as Basic disk devices to the Windows Server.

Using DISKPAR.EXE

DISKPAR.EXE has only two primary functions: to query the selected disk (-i) and to create a disk partition (-s). Figures 1 and 2 demonstrated the use of the query interface. The only additional information required to create or view partition information is the disk number for the device itself.

To obtain the disk number, you can use the Disk Management component. In Figure 8, Disk Management can be seen to show the disk number of the device previously used in these examples. Disk Management is also used to first initialize the raw device for Windows usage. In the following examples, the disks have been configured as Basic disks. Basic disks can be identified by the word `Basic` appearing in the `Type` column of the Disk Management console, and also by `Basic` being displayed in the more-detailed lower-right frame.

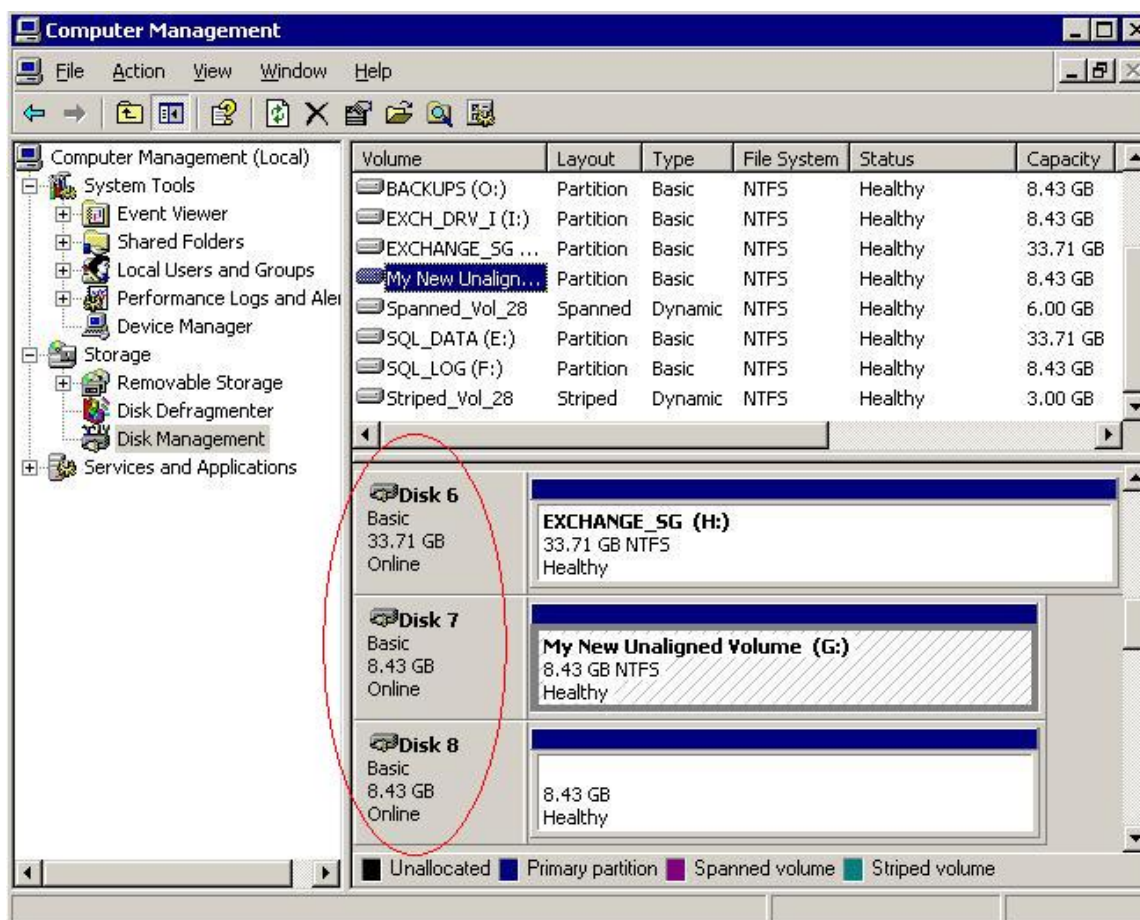


Figure 8. Disk Management Displaying Disk Numbers

In the previous example, *My New Unaligned Volume* was created using Disk Management, and was shown to be misaligned (Figure 2).

The DISKPAR tool should only be used against raw devices; that is, no partitions should exist on the disk. But the disk itself must have been initialized for Windows use, as previously discussed. Be certain of the disk device with which you are dealing. Selecting the incorrect disk device and using DISKPAR to create an aligned partition will overwrite existing partition information and will lead to data loss.

The previously created volume has been deleted, and the DISKPAR utility is executed in the following example.

```

C:\>diskpart -s 7
Set partition can only be done on a raw drive.
You can use Disk Manager to delete all existing partitions
Are you sure drive 7 is a raw device without any partition? <Y/N> Y

---- Drive 7 Geometry Information ----
Cylinders = 1100
TracksPerCylinder = 255
SectorsPerTrack = 63
BytesPerSector = 512
DiskSize = 9047808000 <Bytes> = 8628 <MB>

We are going to set the new disk partition.
All data on this drive will be lost. continue <Y/N>? Y

Please specify starting offset <in sectors>: 128
Please specify partition length <in MB> <Max = 8628>: 8628

Done setting partition.
---- New Partition information ----
StartingOffset = 65536
PartitionLength = 9047113728
HiddenSectors = 128
PartitionNumber = 1
PartitionType = 7

You now should use Disk Manager to format this partition
C:\>_

```

Figure 9. Using DISKPAR to Create an Aligned Partition

DISKPAR is being used to create a single aligned partition on the device referenced as *Disk 7* from Disk Management. The yellow circles in Figure 9 indicate areas of user input required, and are provided in this example.

DISKPAR will warn you that its actions are destructive. It will then display the device geometry and any existing partition information. You should once again ensure that you have correctly selected the device. If any existing partitions had been removed as previously mentioned, there is no listing of partition information in the DISKPAR output, as in the example in Figure 9.

DISKPAR will ask for the starting offset for the new partition. This value must be 64 KB (65536 bytes). The value requested is in sectors—the sector is a Windows sector and is 512 bytes. Therefore, the starting offset should be 128 sectors.

$$\begin{array}{rcl}
 128 & \times & 512 \\
 \text{Offset in sectors} & \times & \text{bytes per sector} \\
 & = & 65536 \\
 & = & 64 \text{ KB aligned}
 \end{array}$$

Once the correct alignment has been entered, you must specify the size of the partition in megabytes. DISKPAR will display the size of the device in megabytes. If you want to use the entire space for a single partition, you should simply enter the value provided by DISKPAR. There is no default value.

When all information has been entered, DISKPAR will create the partition and display details about it. In the previous example, you can see that the partition information lists the starting offset (shown as *StartingOffset* in DISKPAR) as being 65536, which is the value required.

HiddenSectors displays the amount of space allocated to the Windows System Reserved space on the disk device and should match the value that we previously entered as the starting offset (128 sectors). *PartitionNumber* should always be 1 for the first partition on a disk device. A *PartitionType* of 7 is the default created by DISKPAR and represents that this is a Windows partition.

DISKPAR does not create a file system on the partition; DISKPAR is only responsible for creating the partition itself.

Creating Additional Volumes on a Basic Disk

DISKPAR is not able to create additional volumes on a Basic disk, only a single volume is able to be created. While it is possible to create a volume using DISKPAR which does not consume the entire space on the LUN, it is not possible to create a subsequent volume after the DISKPAR created volume which is aligned.

As subsequent volumes cannot be created with correct alignment, it is recommended to only create a single volume on a Basic disk.

Dynamic Disks

Dynamic disks with Microsoft Windows Servers allow administrators to expand upon the limitations imposed by physical disk devices. Dynamic disks provide the ability to create many more partitions/volumes on a single LUN than Basic disks, and to extend those volumes across multiple physical disks in various RAID methods. Dynamic disks also allow for the extension of a given volume to include additional storage should that be required.

EMC arrays do have the capabilities to extend their space allocation. EMC provides documentation on methods expanding the underlying storage allocation, and then extending Basic disk volumes into the additional space. Refer to additional documentation for this specific functionality.

When disks are first presented as raw devices to a Windows Server, they must be initialized by the server, as described previously. This allows for writing unique identification information on the device itself, as well as preparing the device to be managed by the operating system. After this initialization, the device is seen as a Basic disk device. Basic disks can only contain primary partitions or extended partitions (which can then contain multiple logical partitions). Basic disks can be upgraded to Dynamic disks as long as they meet the minimal criteria required which includes at least 1 MB of free space on the disk itself. Refer to the References section on page 19 for locations of additional information on Basic and Dynamic disk functionality, requirements, and restrictions.

Read the Microsoft Best Practices documentation for using Dynamic disks to ensure correct configuration and operational requirements. These recommendations are documented in Knowledge Base article 816307.

At this time, it is not possible to cause a Dynamic disk to create aligned volumes using standard tools. Rather, we need to force the alignment by creating an initial aligned volume on the Basic disk and then upgrading the disk to be Dynamic. This will create subsequent volumes that are aligned. Only volumes that are created when the disk is Dynamic can be dynamic volumes. Volumes that are *upgraded* from a Basic disk are not dynamic, and therefore do not have the attributes of true dynamic volumes. Thus, to provide the alignment functionality and reduce space allocated, we will create a minimal *filler* partition to provide the alignment. This ensures that subsequent volumes are aligned and possess all the dynamic-volume attributes.

Creating a Filler Partition to Force Alignment

This process is identical to the procedure for creating a basic disk partition using DISKPAR.EXE. However, in this instance the goal is to simply create a mechanism that results in subsequent dynamic volumes being aligned. To do this, a small partition must be created to offset any subsequent partitions and ensure that they remain aligned.

The smallest size partition that can be created using DISKPAR.EXE is 1 MB. This is sufficient to create the required alignment for subsequent partitions. The resulting 1 MB volume is not actually useable by the operating system since it is far too small to be formatted for any reasonable use. After the subsequent Dynamic volumes are created, the filler partitions can be removed if required.

When using EMC replication products, the filler partition must be removed. This ensures that the filler partition does not interfere with replication functions that may stem from partition numbering.

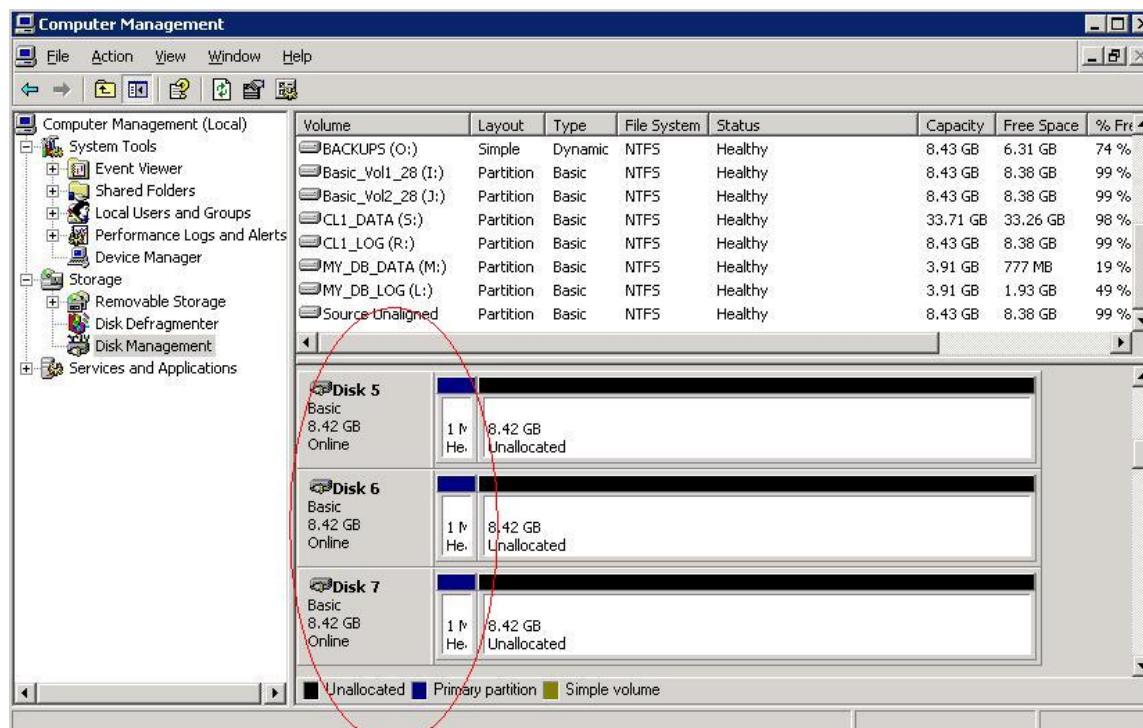


Figure 10. Set of Basic Disks with 1 MB Filler Partition

Once all the filler partitions have been created and the devices upgraded to Dynamic disks, it is possible to create a disk volume as required. The filler partition will ensure that the new volume is aligned.

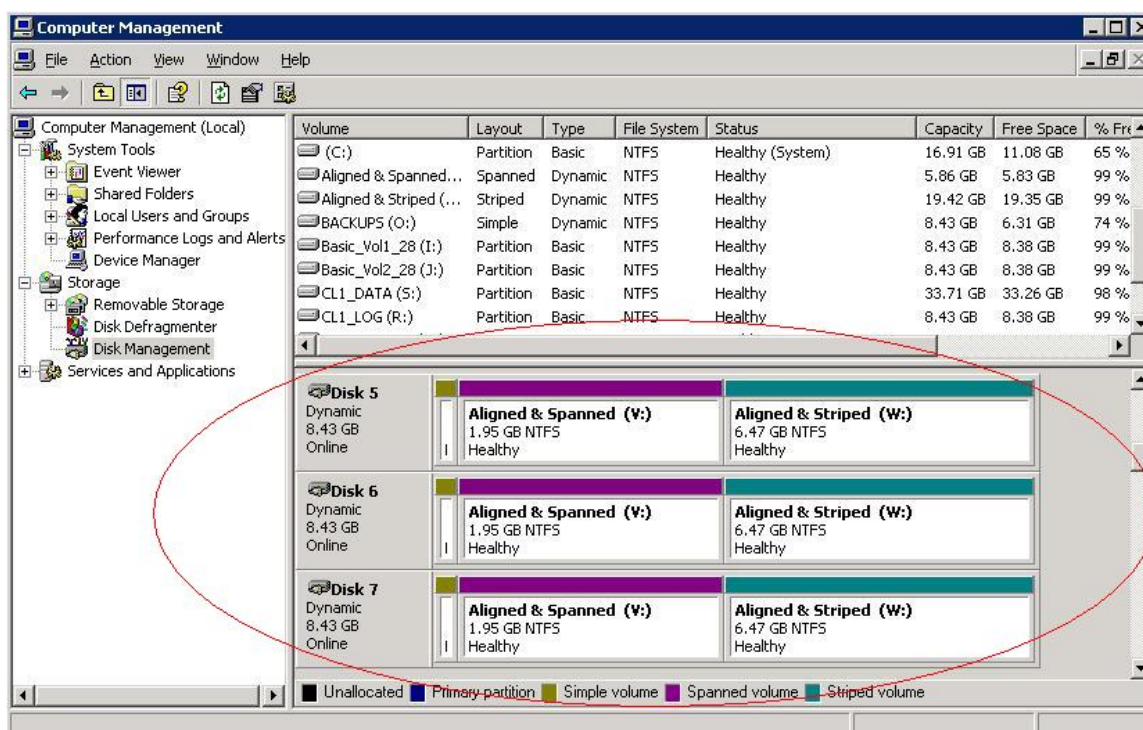


Figure 11. Dynamic Volumes Created after Disk Upgrade (Filler Partition Still in Place)

Verifying Alignment with Dynamic Disks

The DISKPAR.EXE tool does not function correctly with Dynamic disks, and can provide inconsistent results when it is used to look at a disk of this type. To validate Dynamic disk configurations and to obtain accurate alignment information, another tool from the Windows Resource Pack must be utilized. The DMDIAG.EXE tool provides reports on the structure of Dynamic disk configurations. Using this tool, it is possible to look at both Basic and Dynamic disks.

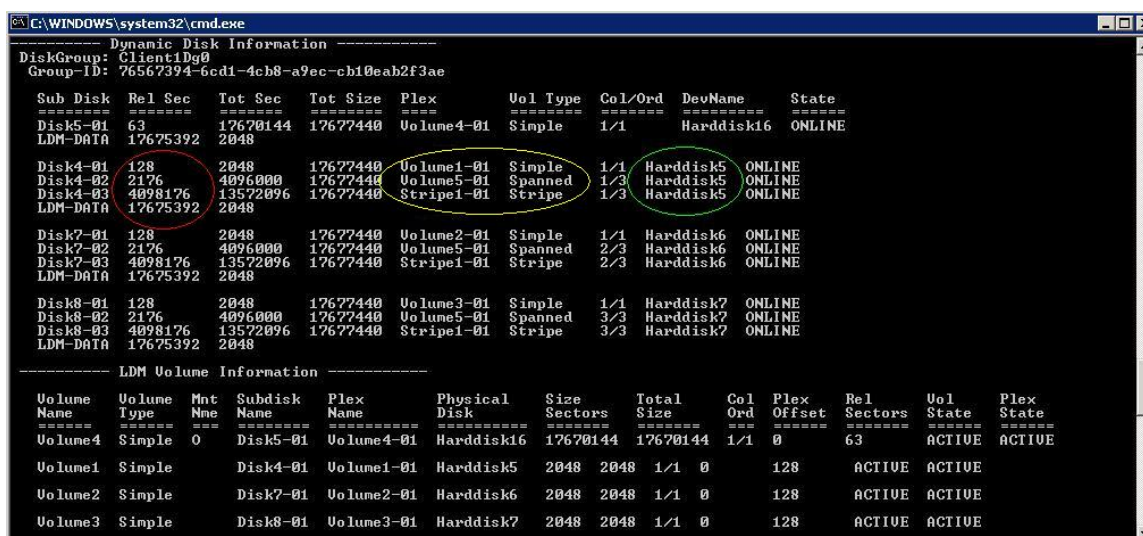


Figure 12. Partial Output from DMDIAG

In Figure 12, the red circle (left) highlights the starting offset of volumes on the disk. The green circle (right) highlights the disk number, which maps to earlier examples, as this is the disk number as displayed within Disk Management. The yellow circle (center) highlights the default name given to the volume as

well as the type of dynamic volume created. The other members of this volume set appear below the highlighted entry; these are hard disks 6 and 7.

Of most interest is the `Rel Sec` column, which shows the starting sector of the partition. The first partition (or filler) starts at sector 128, which is the value provided to DISKPAR during the filler partition creation. The size of this volume is listed as 2048 sectors or 1048576 bytes—more commonly referred to as 1 MB. The second volume starts at an offset of 2176 sectors. This value is a factor of 64 KB and, as such, means that the volume is 64 KB aligned. The third volume is also 64 KB aligned, as its starting sector is 4098176, and this too is a factor of 64 KB.

To check the alignment, multiply the starting offset (in sectors) by 512 bytes. If this number is divisible by 65536 bytes (64 KB), then the partition (and thus the volume itself) is aligned.

Once the required volumes have been created, the filler partition can easily be deleted using Disk Management as shown in Figure 13.

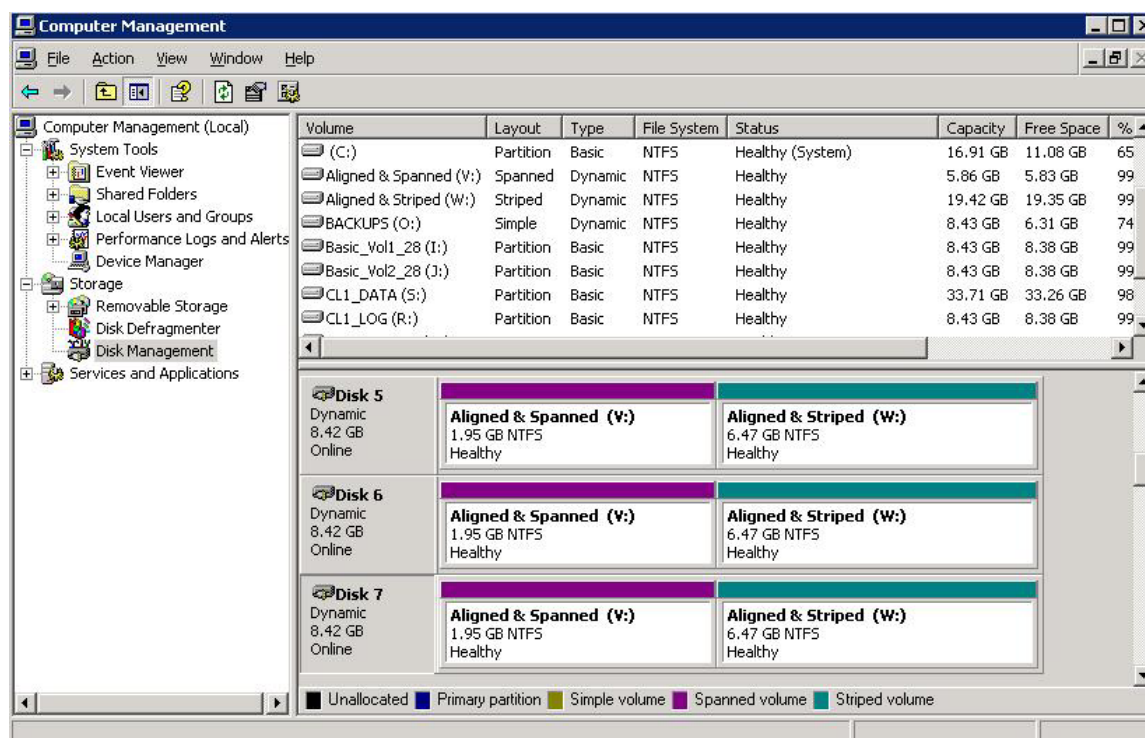


Figure 13. Disk Management after Deletion of the 1 MB Filler Partition

You can use DMDIAG to verify that the removal of the filler partition in no way affects the alignment of the remaining dynamic volumes. Figure 14 displays the output from DMDIAG after the removal of the filler partition.

C:\WINDOWS\system32\cmd.exe

```

----- Dynamic Disk Information -----
DiskGroup: Client1Dg0
Group-ID: 76567394-6cd1-4cb8-a9ec-cb10eab2f3ae

```

Sub Disk	Rel Sec	Tot Sec	Tot Size	Plex	Vol Type	Col/Ord	DevName	State
=====	=====	=====	=====	=====	=====	=====	=====	=====
Disk5-01	63	17670144	17677440	Volume4-01	Simple	1/1	Harddisk16	ONLINE
LDM-DATA	17675392	2048						
Disk4-02	2176	4096000	17677440	Volume5-01	Spanned	1/3	Harddisk5	ONLINE
Disk4-03	4098176	13572096	17677440	Stripe1-01	Stripe	1/3	Harddisk5	ONLINE
LDM-DATA	17675392	2048						
Disk7-02	2176	4096000	17677440	Volume5-01	Spanned	2/3	Harddisk6	ONLINE
Disk7-03	4098176	13572096	17677440	Stripe1-01	Stripe	2/3	Harddisk6	ONLINE
LDM-DATA	17675392	2048						
Disk8-02	2176	4096000	17677440	Volume5-01	Spanned	3/3	Harddisk7	ONLINE
Disk8-03	4098176	13572096	17677440	Stripe1-01	Stripe	3/3	Harddisk7	ONLINE
LDM-DATA	17675392	2048						

Figure 14. DMDIAG Output after Deletion of the 1 MB Filler Partition

Clearly, the volumes remain aligned after the removal of the filler partition. Thus, we have a method of creating dynamic volumes that are aligned on 64 KB boundaries for optimal performance.

Creating Additional Volumes on a Dynamic Disk

After the creation of the filler partition, and conversion to Dynamic disk format, it is possible to create subsequent volumes which are correctly aligned. This process was detailed in the Dynamic disk examples.

Creating the Windows File System

Once the partition that has been created is aligned to Symmetrix internal structures, the file system must be created. The file-system creation will also lead to the drive letter or mount-point assignment. Disk Management may be used to create the file system; optionally, you may create the file system using the Windows FORMAT utility. For information regarding the FORMAT command and its options, refer to the Windows help system.

During the file system creation, Disk Management will require an allocation unit size. This allocation unit size determines the minimum amount of space allocated within the file system when a file is created—space is always allocated as a factor of this allocation unit size.

The default allocation unit size for Windows is based on the size of the partition to be formatted. Refer to *References* on page 19 for the relevant Microsoft documentation for this default value.

Microsoft makes specific recommendations regarding the optimal Allocation Unit size for some of their application environments. Specifically, they provide guidance for file systems to be used for Microsoft SQL Server and for Microsoft Exchange Server.

The guidelines are as follows:

- Microsoft SQL Server for both database and log locations: 64 KB
- Microsoft Exchange Server for all areas: 4 KB

EMC has tested various configurations of the allocation unit size with Exchange environments. It has been observed that utilizing a 64 KB allocation unit size for file systems that host Exchange database and transaction log areas do increase the performance of certain maintenance operations such as defragmentation and backup. Increasing the allocation unit size to 64 KB in this configuration has not been observed to produce a negative effect during general user operations.

References to Microsoft-specific documentation and website locations providing this information are located in the *References* section on page 19. Customers should always confer with their Microsoft consultants where possible to ensure that they implement the current recommendation, which may supersede the information presented in this document.

The value of the allocation unit size to use for the file system being formatted is set through the allocation unit size entry as shown in Figure 15. This value can also be provided using the Windows FORMAT command.

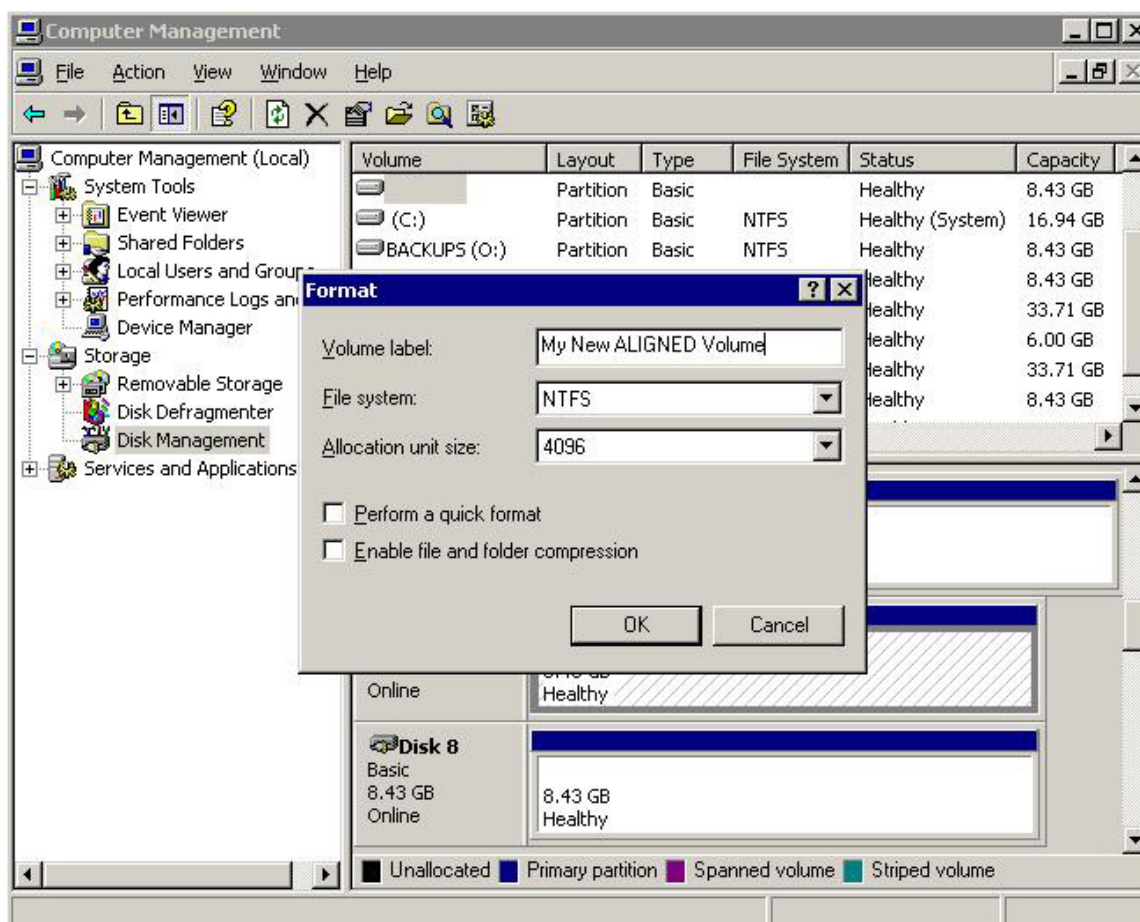


Figure 15. Formatting Windows File System on New Aligned Partition

If we were to now look at the way that file system has been created with respect to the Symmetrix cache as we have done previously, we can see that the file system and all pages are aligned to Symmetrix sectors and thus to cache slots.

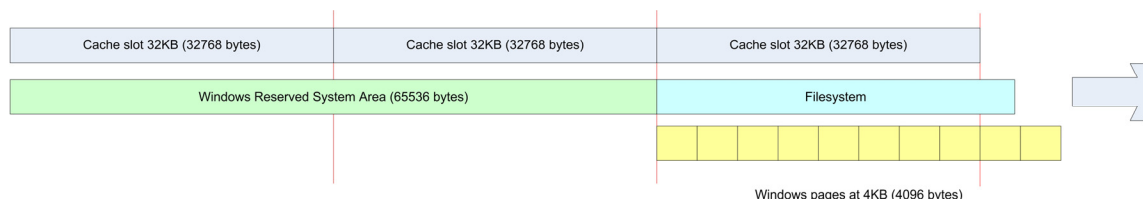


Figure 16. New Aligned File System and Windows Pages

This new aligned file system is optimal for Symmetrix operations. It is the recommended alignment scenario for all Windows file systems created on Symmetrix devices.

References

Default Cluster (Allocation Unit) Size for FAT and NTFS

<http://support.microsoft.com/default.aspx?scid=kb;en-us;140365>

Microsoft SQL Server NTFS Allocation Unit Size

<http://www.microsoft.com/technet/prodtechnol/sql/2000/maintain/sqlops6.msp>

Dynamic Disks and Volumes

http://www.microsoft.com/resources/documentation/WindowsServ/2003/standard/proddocs/en-us/dm_dynamic_overview.asp