

EMC Symmetrix VMAX Virtual Provisioning Space Reclamation and Application Considerations

Abstract

This white paper discusses EMC® Symmetrix® VMAX™ Virtual Provisioning™ space reclamation functionality in the context of server applications, detailing the scenarios in which applications and data migration tools write zeros that can subsequently be reclaimed from virtually provisioned devices.

January 2013

Copyright © 2013 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.

For the most up-to-date listing of EMC product names, see EMC Corporation Trademarks on EMC.com.

VMware are registered trademarks or trademarks of VMware, Inc. in the United States and/or other jurisdictions. All other trademarks used herein are the property of their respective owners.

Part Number h6730.5

Table of Contents

Executive summary.....	5
Audience.....	5
Introduction	6
Terminology	6
Zero space reclamation	7
Executing zero space reclamation	8
Solutions Enabler	8
EMC Unisphere for VMAX.....	9
Space reclamation from previously used devices.....	13
Online reclamation of previously used space.....	13
Storreclaim.....	13
Veritas Storage Foundation	14
vSphere 5.0 U1	14
Windows Server 2012	14
Zero reclamation of previously used space.....	14
Application considerations	16
VMware vSphere	16
“Eagerzeroedthick” with ESXi 4.1+ and Enginuity 5875	17
Considerations for zero space reclamation in VMware environments.....	18
Reclaiming dead space with vSphere 5.0 U1 with Enginuity 5876.159.102 and later ...	19
Reclaiming space on VMFS volumes without Dead Space Reclamation support.....	22
Oracle	26
REDO log files.....	26
Datafiles.....	26
ASM considerations	27
Reclaiming space from deleted Oracle objects	27
Summary of Oracle considerations ASM considerations	27
Microsoft SQL Server	27
Transaction log files	27
Database files	28
Microsoft Exchange Server	30
Microsoft Exchange log files	30
Microsoft Exchange database files	30
Windows Server 2012	31
Microsoft Hyper-V.....	33
Migration and replication.....	35
File system considerations	35

Open Migrator/LM	35
EMC Open Replicator	36
TimeFinder/Clone thick to thin replication	37
Symmetrix Remote Data Facility (SRDF)	37
PowerPath Migration Enabler	38
Veritas Storage Foundation SmartMove	39
Conclusion	40
References	41
White papers	41
Technical notes	41
Feature sheet	41
Product documentation	41

Executive summary

Organizations continually search for ways to both simplify storage management processes and improve storage capacity utilization. When provisioning storage, administrators must consider and otherwise estimate an application's future capacity requirements in addition to current needs. Estimating storage requirements is a difficult process and can lead to miscalculations in the allocation of storage. Administrators frequently choose to overallocate in order to avoid the disruptions and management overhead of exhausting available capacity. Overallocation results in unused storage space, which carries inefficiencies and unnecessary costs.

EMC® Virtual Provisioning™ helps to address these challenges by providing a mechanism of presenting large “thin” LUNs to a host while consuming physical storage from a shared pool only as needed. Symmetrix Virtual Provisioning can improve storage capacity utilization and simplify storage management by presenting an application with sufficient future capacity, reducing the need to frequently provision new storage, while also avoiding costly allocated but unused storage. Virtual Provisioning also simplifies data layout, with automated wide striping that enables organizations to achieve equivalent performance to standard provisioning, with less planning and labor required.

One of the goals when deploying Virtual Provisioning is to ensure the applications and migration tools used to store or to move data from standard or fully provisioned environments to thin environments do not cause unneeded storage allocations. In some cases an application may write a contiguous series of zeros to represent available or initialized but unused space for a volume or file in a file system. When a standard environment is overallocated and a datafile or volume has a high amount of unused space, an application that creates or copies this unused space to a thin device in the form of zeros can cause inefficient space utilization.

To help address this specific concern, Symmetrix® Virtual Provisioning now enables Symmetrix VMAX™ users to automatically reclaim "chunks" containing all zeros. This is most beneficial after migrating from standard volumes to thin volumes but can also prove beneficial in some application scenarios, in order to reduce capacity requirements and TCO.

In addition to zero space reclamation, the Symmetrix VMAX offers a facility for host based applications to request the reclamation of previously used space. This online ability to reclaim previously used areas of a file system further enhances the efficiency of Virtual Provisioning and adds flexibility for changing environments.

Audience

This white paper is intended for server and application administrators, storage architects, customers, and EMC field personnel who want to understand application considerations for Virtual Provisioning space reclamation on the Symmetrix VMAX.

Introduction

Many applications have the potential to write zeros to free space as a part of standard initialization, allocation, or migration processes. Depending on the way zeros are written, there is the potential to reclaim the storage space allocated due to these processes. This white paper discusses some of the most common applications and environment scenarios in which zeros will be written to storage devices. In addition, file system usage patterns can lead to free space within a file system that is consumed within the Virtual Provisioning environment with non-zero data. This paper will outline when it is possible to reclaim all-zero space using EMC Solutions Enabler or the Symmetrix Management Console, and how to reclaim previously used and now non-zero space with host based applications

Terminology

Term	Description
Device	A logical unit of storage defined within a Symmetrix array.
Thin Device	A host accessible device that has no storage directly associated with it.
Data Device	An internal device that provides storage capacity to be used by thin devices.
Thin Device Extent	The minimum quantum of storage that must be mapped at a time to a thin device. Extent also is known as a "chunk."
Data Device Extent	The minimum quantum of storage that is allocated at a time when dedicating storage from a thin pool for use with a specific thin device
Thin Pool	A collection of data devices that provide storage capacity for thin devices.
Bind	The process by which one or more thin devices are associated to a thin pool.
Unbind	The process by which a thin device is disassociated from a given thin pool. When unbound, all previous extent allocations from the data devices are erased and returned for reuse.
Pre-allocating or Pre-provisioning	User specified operation performed against a thin device for the purposes of reducing the operational impact of allocating extents, or for guaranteeing a specified amount of storage for a thin device in a thin pool

Zero space reclamation

Solutions Enabler 7.1, in conjunction with the Enginuity™ 5874 Q4 service release, was the first Symmetrix software release to provide the ability to free up, also referred to as “de-allocate,” storage ranges found to contain all zeros. Space reclamation was designed to be used primarily following a replication or migration activity from a regular device to a thin device, in which software tools such as Open Replicator and Open Migrator (for certain environments) copy all-zero, unused space to the target thin volume. Enginuity 5875 further extended the Virtual Provisioning space reclamation capabilities of the VMAX platform to include on-the-fly zero detection during Open Replicator and Symmetrix Remote Data Facility (SRDF®) migration scenarios.

Zero space reclamation is an extension of the existing Virtual Provisioning space de-allocation mechanism. Previous versions of Enginuity and Solutions Enabler allowed the reclamation of allocated but unused thin device space from a thin pool. Administrators now have the ability to reclaim all-zero space, including both host-unwritten extents/chunks and chunks that contain all zeros due to file system or database formatting methods. The space reclamation process is non-disruptive and can be executed with the targeted thin device ready and read/write to operating systems and applications.

Starting the space reclamation process will spawn a back-end disk director (DA) task that will examine the allocated thin device extents on specified thin devices. A thin device extent is 768 KB (or 12 tracks) in size and is the default unit of storage at which data is allocated for a thin device. For each allocated extent, all 12 tracks will be brought into Symmetrix cache and examined to see if they contain all zero data. If the entire extent contains all zero data, the extent will be de-allocated and added back into the pool, making it available for a new extent allocation operation. Should any portion of the extent contain non-zero data, none of the extent will be reclaimed.

Currently there are some restrictions for zero space reclamation with Symmetrix-based local and remote replication. Prior to Enginuity 5876.159.102, zero space reclamation was not supported on actively replicating SRDF volumes. The link needed to be suspended prior to running the reclamation operation. With 5876.159.102 and later, however, the SRDF relationship can remain active and the reclaim operations will occur against the devices at both sites. All Symmetrix arrays in the SRDF relationship must be at 5876.159.102. Additionally, zero space reclamation will not be performed on tracks that participate in a local replication session, including TimeFinder®/Clone and TimeFinder/Snap.

Executing zero space reclamation

Solutions Enabler

Virtual Provisioning space reclamation can be executed using the Solutions Enabler version 7.1 or later “symconfigure” command line interface (CLI). With older version of Solutions Enabler the symconfigure command “free tdev” could be used with the thin device being the target for reclamation. As a part of the free tdev syntax, a range of cylinders representative of the thin device needed to be specified, based on a beginning and end cylinder value, or size in cylinders, megabytes, or gigabytes. The most common scenario would be to target an entire thin device for reclamation, which can be specified by a starting cylinder of 0 and an ending cylinder of “last_cyl.”

The free tdev command offered two types of reclamation options. The first type, called “unwritten,” was available with previous versions of Solutions Enabler and Enginuity. Reclaiming unwritten space is equivalent to reclaiming pre-allocated but unused storage within a thin pool. Pre-allocating storage is a user-specified operation generally performed to guarantee space for a thin device in a pool, or to reduce the operational impact of allocating extents. Pre-allocation can be done when binding a thin device to a pool, or after a thin device is bound with the “allocate tdev” or “start allocate” symconfigure command.

The second form of reclamation is called “reclaim.” The reclaim operation will perform the previously mentioned reclamation of pre-allocated but unused (unwritten) space, as well as perform the reclamation of space where a thin device extent contains all zero data.

With more recent versions of Solutions Enabler, the symconfigure syntax has changed and it is recommended to instead use “start reclaim.” “Start reclaim” will perform the function of both reclaiming zero space and pre-allocated but unwritten space. Additionally it is not required to include a range of cylinders for the start reclaim syntax. The example below provides the appropriate syntax for starting the reclamation process against an entire thin device, 36d, using the symconfigure command.

```
symconfigure -cmd "start reclaim on tdev 36d;" commit -sid 769
```

The symconfigure command will kick off a background process to perform the specified reclamation. Once initiated the status of the thin device will change from a “bound” state to a “reclaiming” state. One method to view the status of the reclamation process and validate the reclaiming state is to look at the detail of the thin pool where the thin device resides. This can be accomplished by running the symcfg show command against a thin pool with the “-detail” option. The syntax of the command is below, with the resulting output shown in Figure 1.

```
symcfg show -thin -pool FC -detail -sid 769 -gb
```

The *symcfg show* output may differ depending on the Solutions Enabler version and may not show the “Status” column. As an alternative, the ‘*symcfg list -tdev -pool*

`<poolname>`' command can be used to query the Status column or Status flag, which may also differ depending on the version.

```
Pool Bound Thin Devices(14):
{
-----
Sym  FLG      Pool      Pool      Total      Compressed
Dev  T      Total Sub  Allocated  Written  Size/Ratio
-----
036D  R      961.9  8      27.8  3      27.3  3      27.8  0
0398  B      12.0  0      0.2  2      0.2  2      0.2  0
0422  B      10.0  0      0.1  1      0.1  1      0.1  0
0423  B      10.0  0      0.0  0      0.0  0      0.0  0
0424  B      10.0  0      0.0  0      0.0  0      0.0  0
0425  B      10.0  0      0.0  0      0.0  0      0.0  0
0426  B      10.0  0      0.0  0      0.0  0      0.0  0
0460  B      10.0  0      0.0  0      0.0  0      0.0  0
0461  B      10.0  0      0.0  0      0.0  0      0.0  0
0462  B      10.0  0      0.0  0      0.0  0      0.0  0
0463  B      10.0  0      0.0  0      0.0  0      0.0  0
0464  B      10.0  0      0.0  0      0.0  0      0.0  0
0465  B      10.0  0      0.0  0      0.0  0      0.0  0
0466  B      10.0  0      0.1  1      0.1  1      0.1  0
-----
GBs      1093.9  9      28.2  3      27.7  3      28.2  0
}

No Other-Pool Bound Thin Devices have allocations in Device Pool FC

Legend:
Enabled devices FLG:
(S)hared Tracks : X = Shared Tracks , . = No Shared Tracks
Bound Devices FLG:
S(T)atus      : B = Bound, I = Binding, U = Unbinding, A = Allocating,
                D = Deallocating, R = Reclaiming, C = Compressing,
                N = Uncompressing, . = Unbound,
```

Figure 1. `symcfg show` command to display "reclaiming" status

The reclamation process can also be queried by using the “`symcfg verify`” command. The verify command has been enhanced to include a “reclaiming” parameter for the purposes of verifying whether a thin device has entered a reclaiming state. The verify command can be useful when scripting operations, which will check to ensure a given thin device has begun reclaiming. An example for running the `symcfg verify` command is as follows:

```
symcfg verify -tdev -reclaiming -dev 36d -sid 769
```

Once a thin device has completed the reclaim background task it will change status from reclaiming back to bound. The same `symcfg show` and `symcfg verify` commands as previously mentioned can also be used to query and otherwise verify that a thin device has completed reclamation and returned to a bound state.

EMC Unisphere for VMAX

Unisphere for VMAX can also be used to initiate and monitor the Virtual Provisioning space reclamation process. To start space reclamation first select the Symmetrix system, then go to **Storage > Volumes** to open the **Volumes Dashboard**. From within the dashboard expand “Virtual Volume,” select “TDEV” and click “view” as shown in Figure 2.

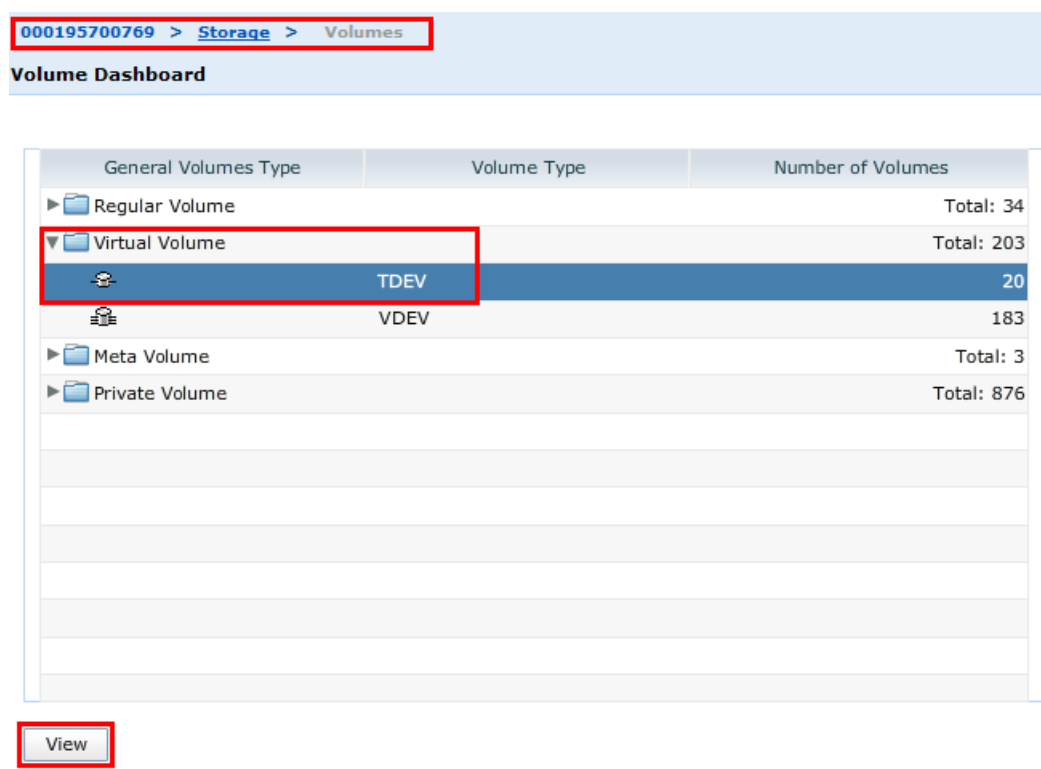


Figure 2. Unisphere for VMAX Volumes Dashboard

From within the **Thin Volumes** view, right click on the appropriate thin device and choose “Start Allocate/Free/Reclaim ...” as shown in Figure 3.

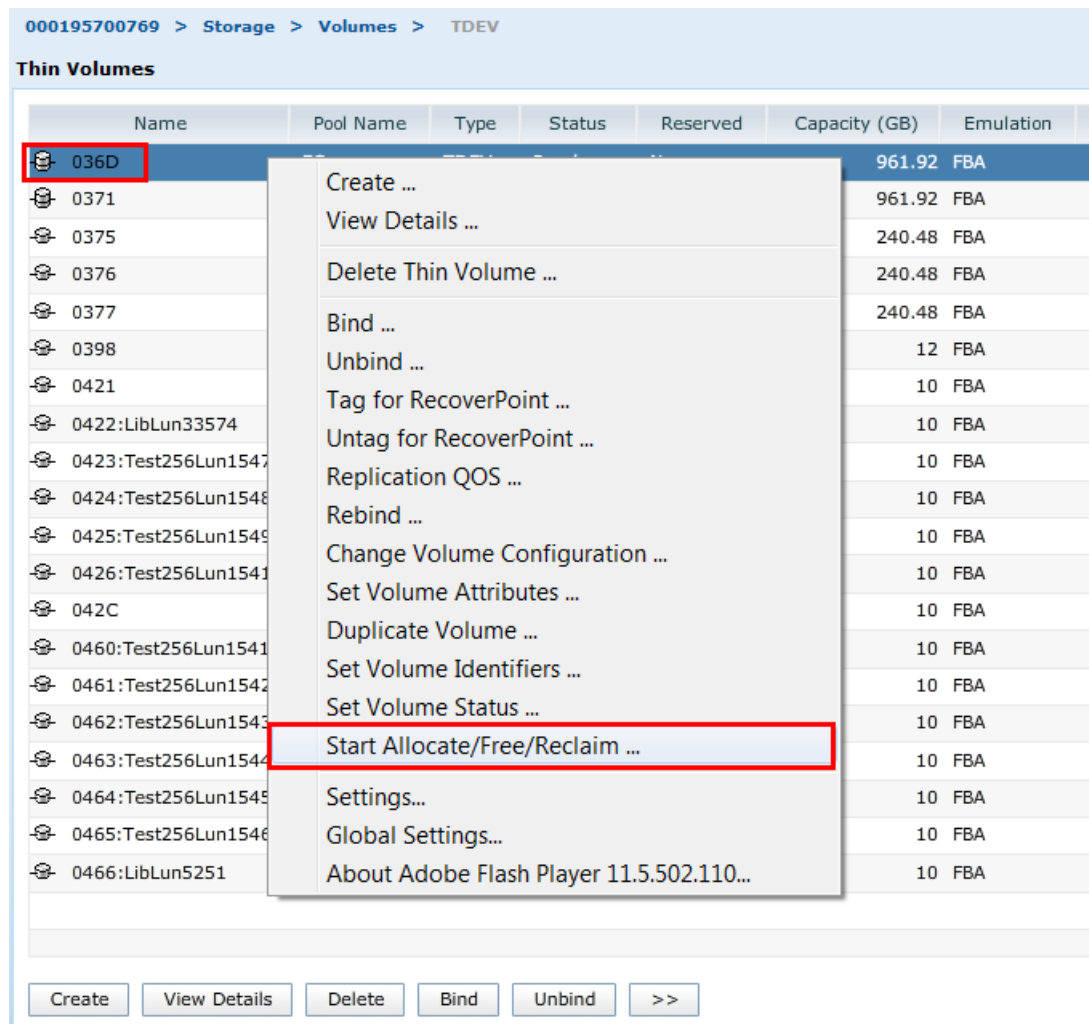


Figure 3. Allocate/Free/Reclaim Thin Device selection

From within the Start Allocate/Free/Reclaim menu, choose “Start Reclaim” and “Full Volume.” Optionally select to “reclaim persistent capacity.” Then choose to “add to job list” or “run now,” as seen in Figure 4, to begin the reclaim background process. If add to job list was chosen, the job will need to be scheduled or started from under **System > Job List**

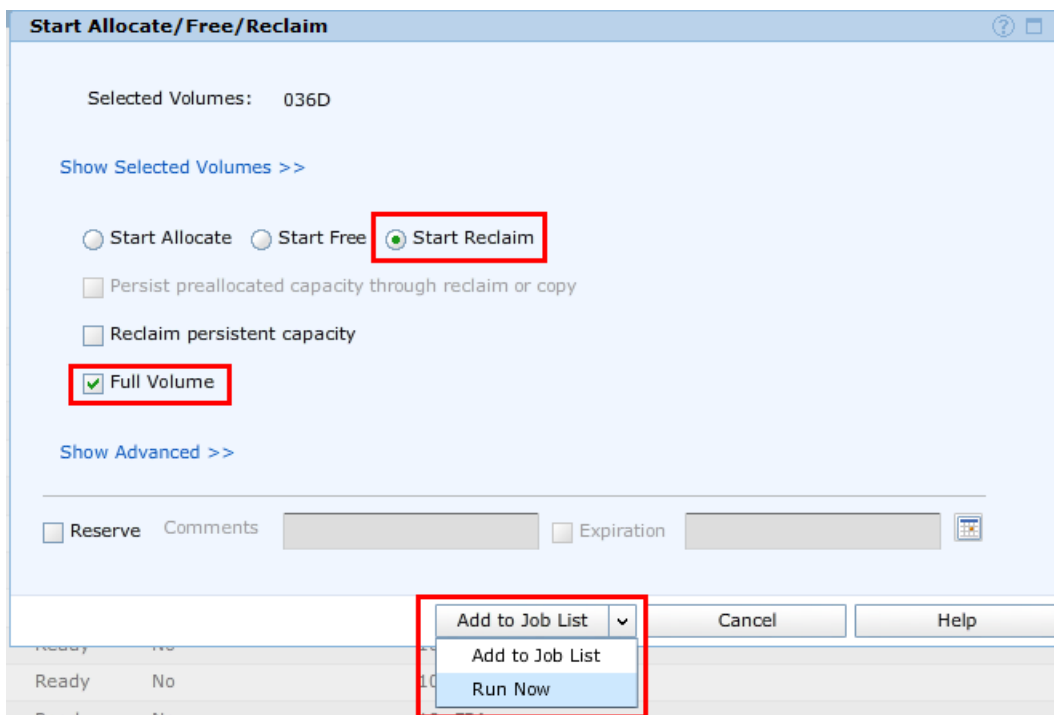


Figure 4. Reclaim options and execution

After the reclaim job is started, the reclamation process will operate as a background task on the Symmetrix. To verify that reclamation has started for a given thin device, navigate to the **Thin Pools** view under **Storage**. Select the appropriate pool and click “View Details.” From the details window, look for “related” objects” and click on the “Bound Volumes” link. As shown in Figure 5 the bound volumes view has a “Status” column that will show the reclaiming state of the thin device.

000195700769 > Storage > Thin Pools > FC > Bound Volumes										
Bound Volumes For Thin Pool										
Name	Config...	Emula...	Volume Alloc...	Capacity ...	Pool Allocatio...	Total Sub...	Total Writ...	Shared...	Persistent...	Status
036D	TDEV	FBA		961.92	27.79	8 %	27.27	No	None	Reclaiming
0398	TDEV	FBA		12	0.25	0 %	0.24	No	None	Bound
0422	TDEV	FBA		10	0.08	0 %	0.07	No	None	Bound
0423	TDEV	FBA		10	0	0 %	0	No	None	Bound
0424	TDEV	FBA		10	0	0 %	0	No	None	Bound

Figure 5. Viewing the reclaiming status of a thin device

Once the reclamation process is complete, the status of the device will return to bound. Refreshing the view in Figure 5 will reflect the change in status upon completion.

Space reclamation from previously used devices

Online reclamation of previously used space

Engenuity 5875 introduced support for the T10 industry-standard SCSI command called WRITE_SAME unmap. This SCSI command allows for operating systems and applications to communicate to the Symmetrix VMAX a range of logical block addresses (LBAs) that are no longer needed. If the LBA range covers an entire Virtual Provisioning extent (12 tracks or 768 KB), then the extent can be reclaimed.

With Engenuity 5876.159.102, support for the T10 industry-standard UNMAP command was added. UNMAP also allows for operating systems and application to communicate non-zero LBA ranges that are no longer needed and can be deallocated.

There are some caveats with UNMAP use where certain device types/configurations may cause the UNMAP command to be blocked or not valid for the entire device. These are:

1. Not supported on devices in use by Open Replicator for Symmetrix (ORS)
2. Not supported with devices in use by RecoverPoint
3. Not supported for encapsulated FTS devices
4. Can be partially blocked by thin devices that have persistent allocations. The VMAX accepts the UNMAP command but respects persistent allocations. If the specified UNMAP range has persistent and non-persistent allocations, the VMAX will de-allocate the non-persistent allocations and leave the persistent allocations.
5. Not supported with virtual devices (TimeFinder/Snap targets)
6. Not supported with devices currently involved in duplicate write sessions (this is a temporary state; these sessions are used during meta reconfiguration and expansion)
7. Not supported with TimeFinder/VP Snap targets

UNMAP to a thin SRDF device is accepted and supported on the R1 and is propagated to its remote replicas (R21, R2) as long as the links are active and the remote boxes are also running 5876.159.102.

Applications that utilize the UNMAP or WRITE_SAME unmap capabilities of Symmetrix VMAX help to enable the reclamation of previously used, but now deleted, areas of a thin device, without the need to zero out areas of a file system or otherwise unbind the thin device.

Storreclaim

For Windows and Linux environments, the *storreclaim* utility has been written by EMC to take advantage of the WRITE_SAME unmap command. The *storreclaim* command

is intended to be run against file systems that contain previously written but now deleted files. The utility will create sparse files within the free space of the file system, map out the extents of the newly created files, and subsequently issue a WRITE_SAME unmap command for those ranges of extents. Symmetrix VMAX will then reclaim the ranges of previously allocated and written space within the thin pool. The *storreclaim* command is nondisruptive and can be run while the file system being affected is mounted and in use.

The *storreclaim* command was initially released as part of the Global Services ToolKit. EMC employees can now find more details regarding the *storreclaim* command on Powerlink under the Support and Product and Diagnostic Tools heading. Customers should contact an EMC representative for more information.

Veritas Storage Foundation

Veritas Storage Foundation also offers a reclamation facility that utilizes the WRITE_SAME unmap specification inherent in the Symmetrix VMAX with Engenuity 5875. The reclamation capabilities of Storage Foundation span a wide range of operating systems and file systems. For more information regarding the supported platforms for thin reclamation with Veritas Storage Foundation, please see the appropriate hardware compatibility lists on <http://www.symantec.com>.

vSphere 5.0 U1

Referred to as Dead Space Reclamation by VMware, ESXi 5.0 U1 introduced support for the SCSI UNMAP command which issues requests to de-allocate extents on a thin device. Dead Space Reclamation offers the ability to reclaim blocks of a thin-provisioned LUN on the array via a VMware-provided command line utility. Please see the VMware vSphere section for additional details.

Windows Server 2012

Windows Server 2012 provides identification for thin (virtually) provisioned storage as well as file system aware reclamation capabilities, commonly referred to as TRIM within Microsoft documentation. When a file is deleted from within a file system, Windows Server 2012 can issue reclaim requests to the underlying storage. Windows will also issue real-time reclaim operations when formatting a file system on a thin LUN. Windows Server 2012 utilizes the UNMAP specification when performing reclaim operations against a Symmetrix running 5876.159.102 or higher. Please see the Windows Server 2012 and Microsoft Hyper-V sections for additional details.

Zero reclamation of previously used space

In some cases devices are no longer needed and can be re-used for other applications. Such devices would continue to retain stale non-zero data from prior uses and will not allow reclamation of unused storage space. Better capacity utilization and cost savings will be realized if the devices are zeroed out first using operating system tools prior to re-provisioning. As a caution, prior to using such tools administrators must make sure that the devices don't contain any active application data.

On UNIX and Linux systems, a command like “dd” can be used to zero out the disk blocks. If an entire partition is re-provisioned then “dd” can be used to zero out the whole partition. Once allocated space on the device has zeros in contiguous blocks, space reclamation can be used to reclaim the zeroed (and un-used) storage in the thin pool. For example, `dd if=/dev/zero of=/dev/emcpowerdf1 bs=1024 count=1024000`, will zero out 1 GB space on the first partition on device `/dev/emcpowerdf`.

On Windows 2003 and 2008 systems the diskpart “clean all” command can be issued to zero out all areas of a specified disk. With Windows 2008 only, a non-quick or full format will also write zeros to the targeted volume.

Additionally, as detailed in the “File system considerations” section, external tools such as sdelete for Windows can be used to zero out contiguous ranges of a device.

The use case for zeroing out a standard device is most interesting in the context of future replication from thick to thin volumes, like with Open Replicator or TimeFinder/Clone. The target thin devices in these scenarios will be able to reclaim the now zero space as replicated from the source.

If the source is a thin device and the entire device is re-provisioned, then the most efficient method for reclamation would be to unbind and re-bind to the thin pool to automatically reclaim space allocated by the device. For example, if the thin device 0297 on Symmetrix 191 is added to a thin pool Oracle1 and masked to a host then the following commands can be used to reclaim space.

1. Change device state to “not_ready” to prevent host access:
 - a. `symdev -sid 191 -RANGE 0297:0297 not_ready:`
2. Unbind the device from the thin pool:
 - a. `symconfigure -sid 191 -cmd “unbind tdev 0297 from pool Oracle1 ;”
commit`
3. Rebind the device to the thin pool:
 - a. `symconfigure -sid 191 -cmd “bind tdev 0297 to pool Oracle1 ;”
commit`

Application considerations

At times of initialization or as a part of cleanup operations, applications may write contiguous areas of zeros to datafiles or file systems. The following sections discuss when commonly used applications will write zeros and whether those zeros can be reclaimed with Virtual Provisioning space reclamation.

VMware vSphere

In a VMware environment, the zeroing out of virtual disks can lead to unnecessary physical storage allocation on Symmetrix thin pools, a situation that is inherently detrimental to the space-saving benefit of Symmetrix Virtual Provisioning. The ability to perform space reclamation on thin devices hosting these zeroed-out virtual disks alleviates this issue. In VMware vSphere™ there are multiple situations where performing space reclamation on thin devices will be helpful.

VMware vSphere offers multiple ways of formatting virtual disks and has integrated these options into VMware vCenter™. For new virtual disk creation, the formats eagerzeroedthick, zeroedthick, and thin are offered as options.

All of the allocation mechanisms that the VMware vSphere Client provides for creating virtual disks are listed in Table 2.

Table 1 Allocation policies when creating new virtual disks on a VMware datastore

Allocation mechanism (Virtual Disk format)	VMware kernel behavior	Effect of zero reclaim
zeroedthick	All space is allocated at creation but is not initialized with zeros. However, the allocated space is wiped clean of any previous contents of the physical media. This is the default policy when creating new virtual disks.	No effect unless guest OS itself has written contiguous zeros
eagerzeroedthick	This allocation mechanism allocates all of the space and initializes all of the blocks with zeros. This option performs a write to every block of the virtual disk, and hence results in equivalent storage use in the thin pool.	Will reclaim all portions of the virtual disk not written to by the guest OS. Portions of the virtual disk that contain contiguous zeros written by the guest OS will be reclaimed as well.
thin	This allocation mechanism does not reserve any space on the VMware VMFS on creation of the virtual disk. The space is allocated and zeroed on demand.	No effect unless guest OS itself has written contiguous zeros

In VMware vSphere, the ability to choose virtual disk allocation mechanisms allows administrators to deploy virtual machines using the thin, zeroedthick, or eagerzeroedthick mechanisms. This enables an administrator to deploy virtual disks in a manner friendly to the spacing saving benefit of Symmetrix Virtual Provisioning.

As shown in Figure 6, if a new virtual machine is created with the box **Thick Provision Eager Zeroed** selected, the virtual machine will be created using the “eagerzeroedthick” allocation mechanism and if placed on a thin device, will result in a complete allocation due to the zeroing of the virtual disk. Space reclamation can be used in this instance to alleviate the consumed space on the thin pool. However, if the virtual machine is targeted to be used with VMware Fault Tolerance or cannot stand the small additional latency associated with the first write to a reclaimed (unallocated) Symmetrix VP extent, it is not recommended to reclaim those zeros. In these cases, the virtual disks of the virtual machine(s) were pre-allocated purposefully and those zeros (and more importantly the VP allocations) are not meant to be removed.

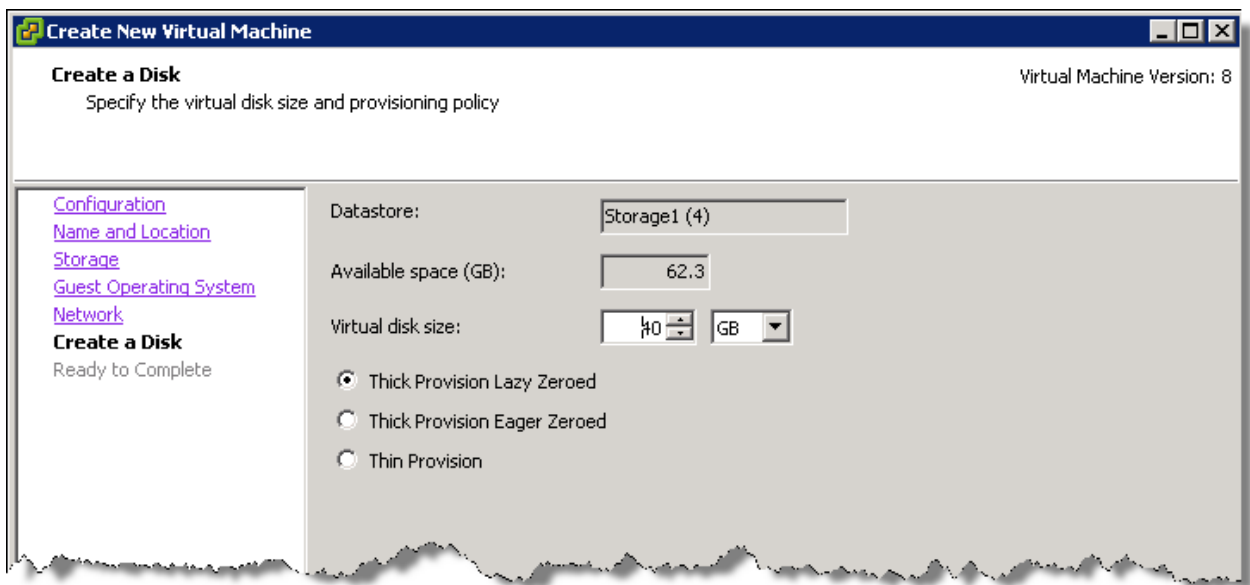


Figure 6. Virtual disk allocation mechanism options in vSphere 5.0

“Eagerzeroedthick” with ESXi 4.1+ and Enginuity 5875

VMware vSphere 4.1/5.x offers a variety of VMware vStorage APIs for Array Integration (VAAI) that provides the capability to offload specific storage operations to EMC Symmetrix VMAX to increase both overall system performance and efficiency. Symmetrix VMAX with Enginuity 5875 supports the following of VMware's new vStorage APIs - Full Copy and Block Zero and Hardware-Assisted Locking.

The Block Zero primitive delivers hardware-accelerated zero initialization, greatly reducing common input/output tasks such as creating new virtual machines. This feature is especially beneficial when creating fault-tolerant (FT)-enabled virtual machines. Having the array complete bulk zeroing of a virtual disk instead of the host, speeds up the standard initialization process. Without the block zeroing primitive, the command is not complete until the host has completed the zeroing process. For a very large disk this could take a long time. The block zeroing primitive, which uses the write same SCSI command (0x93), enables the disk array to return the cursor to the

requesting service as though the process of writing the zeros has been completed and then finish the job of zeroing out those blocks in the background on the array.

This primitive has a profound effect on Symmetrix Virtual Provisioning when deploying "eagerzeroedthick" virtual disks. When this feature is enabled, combined with Symmetrix Virtual Provisioning, the typical host zeroing that occurs when deploying disks using this allocation mechanism, will be offloaded to the array. Furthermore, the Symmetrix will not write the zeros to disk but instead simply set the track as "Never Written By Host" and allocate the respective tracks on the thin pool. This behavior invalidates the need for zero reclaim for newly-created eagerzeroedthick virtual disks. Since no zeros are written, no zeros need be reclaimed. However, these allocated, but unwritten, tracks can be de-allocated using the unwritten reclamation process. The same recommendations apply in this situation as ordinary eagerzeroedthick disks. Since the VP extents are still allocated with Block Zero, the first write latency for the guest OS does not occur. If it is decided that this is not an issue, the allocation can be reclaimed. Otherwise, do not reclaim these allocations.

Block Zero is enabled by default on both the Symmetrix VMAX running 5875 Engenuity and later, and on a properly licensed ESX(i) 4.1+ host.

Considerations for zero space reclamation in VMware environments

For vSphere, there are important considerations that should be evaluated before performing reclaims.

If a thin device stores multiple virtual machines, a zero reclaim function cannot be performed on just one of the virtual disks. It is currently not possible to determine the specific blocks each virtual disk consumes on the thin device and therefore it is not possible to limit a reclaim to a single virtual disk out of many on a given Virtual Machine file system. Consequently, a reclaim should only be performed on a thin device if all virtual machines hosted by the VMware file system volume on that thin device can and should be reclaimed. In particular, in vSphere environments, virtual machines that are set up in a Fault Tolerant configuration or require the pre-allocation of eagerzeroedthick virtual disks should not share Symmetrix thin devices (or the Virtual Machine file systems that are hosted by them) with other virtual machines. This will reduce the risk of reclaiming the zeros from these virtual machines.

Additionally, zeroing out by guest operating systems must be taken into account before performing a zero reclaim. If the virtual machines' operating system or its hosted applications zero out files for a particular reason, care must be taken to consider any implications of removing those zeros before reclaiming them.

The use of EMC Virtual Storage Integrator Storage Viewer feature is highly recommended in order to easily map virtual machines and their corresponding VMFS volumes to the correct underlying Symmetrix thin device. Double-checking storage mapping information with the EMC VSI Storage Viewer will eliminate the possibility of performing zero reclamation on the incorrect thin device.

Reclaiming dead space with vSphere 5.0 U1 with Enginuity 5876.159.102 and later

Referred to as Dead Space Reclamation by VMware, ESXi 5.0 U1 introduced support for the SCSI UNMAP command (0x42) which issues requests to de-allocate extents on a thin device. Dead Space Reclamation offers the ability to reclaim blocks of a thin-provisioned LUN on the array via a VMware-provided command line utility.

It is important to note that the Symmetrix only supports this functionality with Enginuity 5876.159.102 and higher. If a customer is running Enginuity 5876.82 or lower, they must therefore upgrade Enginuity to leverage this functionality or utilize the workaround process described in the next section *Reclaiming space on VMFS volumes without Dead Space Reclamation support*.

A Dead Space Reclamation should be executed after a virtual disk is deleted or is migrated to a different datastore. As shown in the previous section, when virtual machines were migrated from a datastore, the blocks used by the virtual machine prior to the migration were still being reported as “in use” by the array. With this new VMware Storage API for Array Integration primitive, the Symmetrix will be informed that the blocks are no longer in use, resulting in better reporting of disk space consumption and a much more efficient use of resources.

To confirm that the Enginuity level is correct via the ESXi host and that UNMAP is in fact supported on a LUN, the following command can be issued to query the device:

```
esxcli storage core device vaai status get -d <naa>
```

Example output:

```
naa.<naa>
```

```
    VAAI Plugin Name: VMW_VAAI_SYMM
```

```
    ATS Status: supported
```

```
    Clone Status: supported
```

```
    Zero Status: supported
```

```
    Delete Status: supported
```

A device displaying **Delete Status** as supported means that the device is capable of receiving SCSI UNMAP commands or, in other words, the Symmetrix is indeed running the correct level of Enginuity to support UNMAP from an ESXi host.

In order to reclaim space, ESXi 5.0 Update 1 includes an updated version of `vmkfstools` that provides an option (`-y`) to issue the UNMAP operation. This command should be issued against the device after navigating to the root directory of the VMFS volume residing on it:

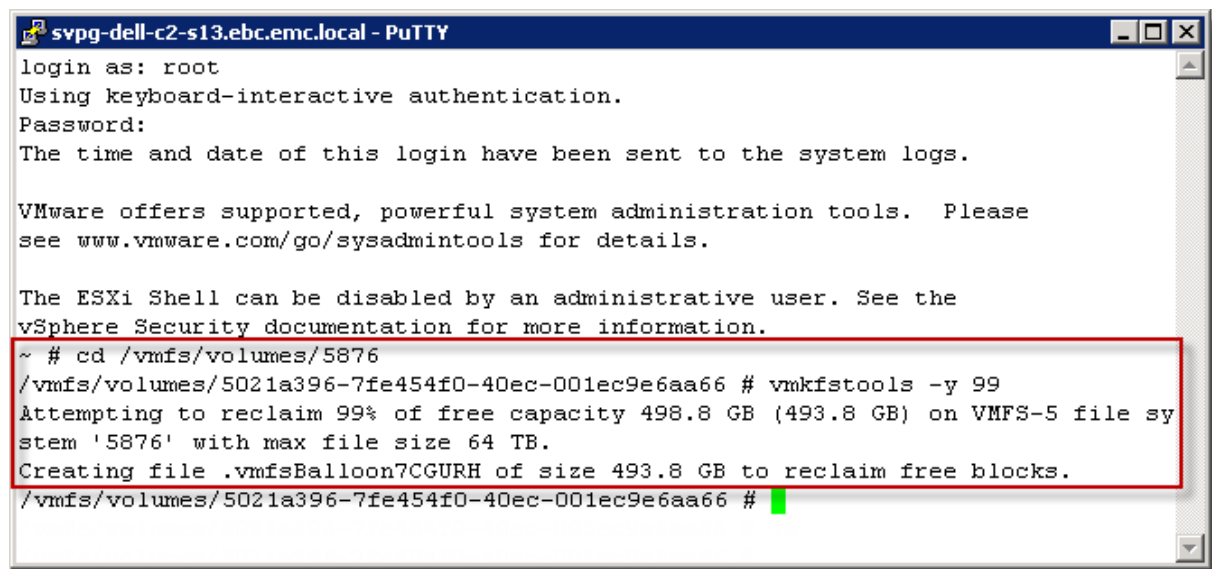
```
cd /vmfs/volumes/<volume-name>
```

```
vmkfstools -y <percentage of deleted block to reclaim>
```

The percentage of deleted blocks can be configured from anywhere between 1 to 99%.

This command creates a temporary “balloon” file at the top level of the datastore. This file can be as large as the aggregate size of blocks being reclaimed as seen in

Figure 7. If the reclaim operation is somehow interrupted, this temporary file may not be deleted automatically and users may need to delete them manually.



```
svpg-dell-c2-s13.ebc.emc.local - PuTTY
login as: root
Using keyboard-interactive authentication.
Password:
The time and date of this login have been sent to the system logs.

VMware offers supported, powerful system administration tools. Please
see www.vmware.com/go/sysadmintools for details.

The ESXi Shell can be disabled by an administrative user. See the
vSphere Security documentation for more information.
~ # cd /vmfs/volumes/5876
/vmfs/volumes/5021a396-7fe454f0-40ec-001ec9e6aa66 # vmkfstools -y 99
Attempting to reclaim 99% of free capacity 498.8 GB (493.8 GB) on VMFS-5 file sy
stem '5876' with max file size 64 TB.
Creating file .vmfsBalloon7CGURH of size 493.8 GB to reclaim free blocks.
/vmfs/volumes/5021a396-7fe454f0-40ec-001ec9e6aa66 #
```

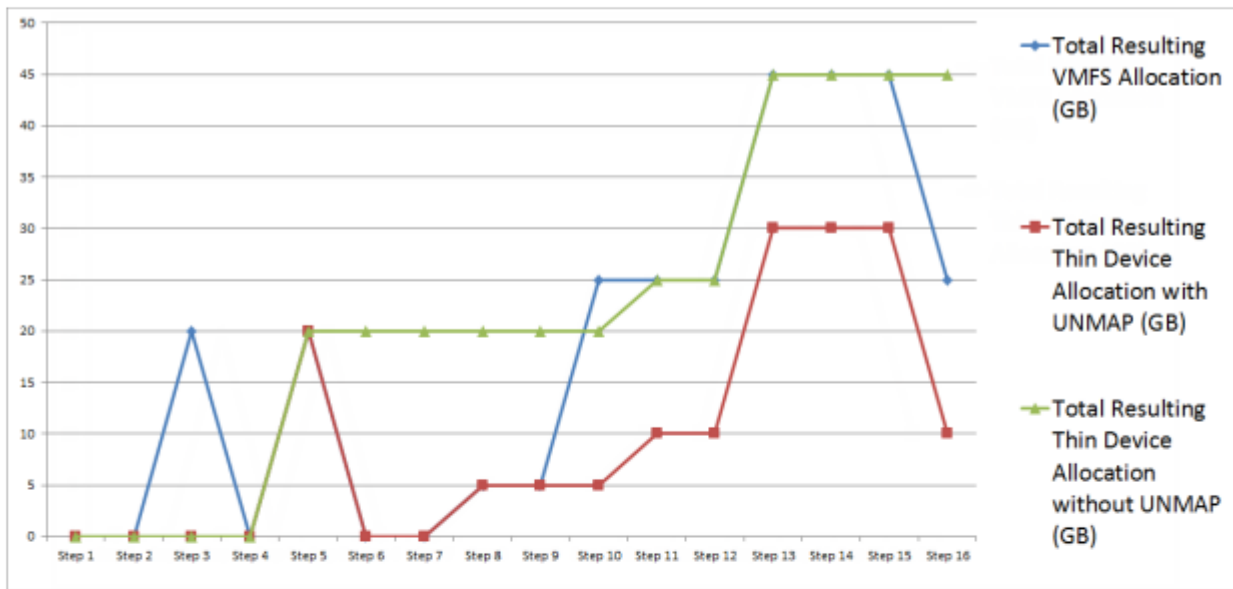
Figure 7. SSH session to ESXi server to reclaim dead space with vmkfstools

It is recommended to not use too large of a percentage in the `vmkfstools` reclaim command as resulting balloon file temporarily consumes and reserves space on the datastore. If a new virtual disk needs to be deployed or an existing one needs to be grown, it is possible that a large enough temporary balloon file will block these actions. Therefore, it is recommended to use a reclaim percentage closer to 60% and to avoid using the maximum of 99%.

There is no practical way of knowing how long a `vmkfstools -y` operation will take to complete. It can run for a few minutes or even for a few hours depending on the size of the datastore, the amount of dead space and the available resources on the array to execute the operation. Reclamation on the Symmetrix is a low-priority task (in order for it to not negatively affect host I/O performance) so a heavy workload on the device¹, pool or the array itself can throttle the process and slow the reclamation.

Figure 8 shows the results from a series of tests conducted to determine the behavior of the VMFS when virtual disks are created and deleted and when files within those virtual disks are created or deleted. A Dead Space Reclamation operation (`vmkfstools -y`) is executed after each delete operation. The graph shows the used capacity in GB of the VMFS volume as vSphere sees it (blue line) and the actual GBs allocated on the thin pool after the operation (red line). This chart also shows with the green line what the TDEV allocation would be if UNMAP was not supported and the dead space was not returned to the pool.

¹ An in-progress VAAI XCOPY session to the device being reclaimed (due to a Storage vMotion or Clone operation) can cause the device to exceed device pending operation limits thereby throttling the UNMAP session to possibly substantially reduced rates until the threshold has been cleared.



Step #	Operation
Step 1	Create a 20 GB Thin virtual disk
Step 2	Delete the 20 GB Thin virtual disk created in step 1 and execute "vmkfstools -y"
Step 3	Create a 20 GB ZeroedThick virtual disk
Step 4	Delete the 20 GB ZeroedThick virtual disk created in step 3 and execute "vmkfstools -y"
Step 5	Create an EagerZeroedThick virtual disk
Step 6	Delete the EagerZeroedThick virtual disk created in step 5 and execute "vmkfstools -y"
Step 7	Create a 20 GB Thin virtual disk
Step 8	Write 5 GBs of data on to the virtual disk created in step 7
Step 9	Delete the 5 GBs of data written in step 8 and execute "vmkfstools -y"
Step 10	Create a 20 GB ZeroedThick virtual disk
Step 11	Write 5 GBs of data on to the virtual disk created in step 10
Step 12	Delete the 5 GBs of data written in step 11 and execute "vmkfstools -y"
Step 13	Create a 20GB EagerZeroedThick virtual disk
Step 14	Write 5 GBs of data on to the virtual disk created in step 13
Step 15	Delete the 5 GBs of data written in step 14 and execute "vmkfstools -y"
Step 16	Migrate the 20 GB EagerZeroedThick virtual disk created in step 13 off the datastore and execute "vmkfstools -y" on the source datastore

Figure 8. VMFS allocation mechanisms with Dead Space Reclamation

It can be seen from Figure 8 that the VMFS utilization drops when the virtual disks are deleted and vmkfstools is run but not when files within them are deleted no matter the type of virtual disk. Due to Dead Space Reclamation (UNMAP), the percent utilization of the thin pool changes when virtual disks containing actual data or eagerzeroedthick virtual disks are deleted. Also of note, the VMFS volume reuses the freed blocks as new files are created and deleted on the file system. VMFS, with the addition of Dead Space Reclamation, exhibits a behavior that is exceptionally beneficial when used with virtually provisioned devices.

Reclaiming space on VMFS volumes without Dead Space Reclamation support

As described earlier, in versions of vSphere prior to 5.0 U1 or Symmetrix arrays running Enginuity 5876.82 or earlier, thin device allocations cannot be reclaimed with the `vmkfstools` command. If an ESX and/or Enginuity upgrade is not possible, the following process is a workaround² to remove these allocations quickly and simply:

1. Log into the vSphere Client and ensure that the EMC Virtual Storage Integrator Storage Viewer plugin is installed.
2. Synchronize VSI. To make sure VSI has the most up-to-date information concerning thin pool allocations, it is important to synchronize the local database supporting VSI Storage Viewer. In the vSphere Client, navigate to Home > Solutions and Applications > EMC and click the Symmetrix Arrays panel. Click the “Sync All Arrays” link at the top of the panel as shown in Figure 9.

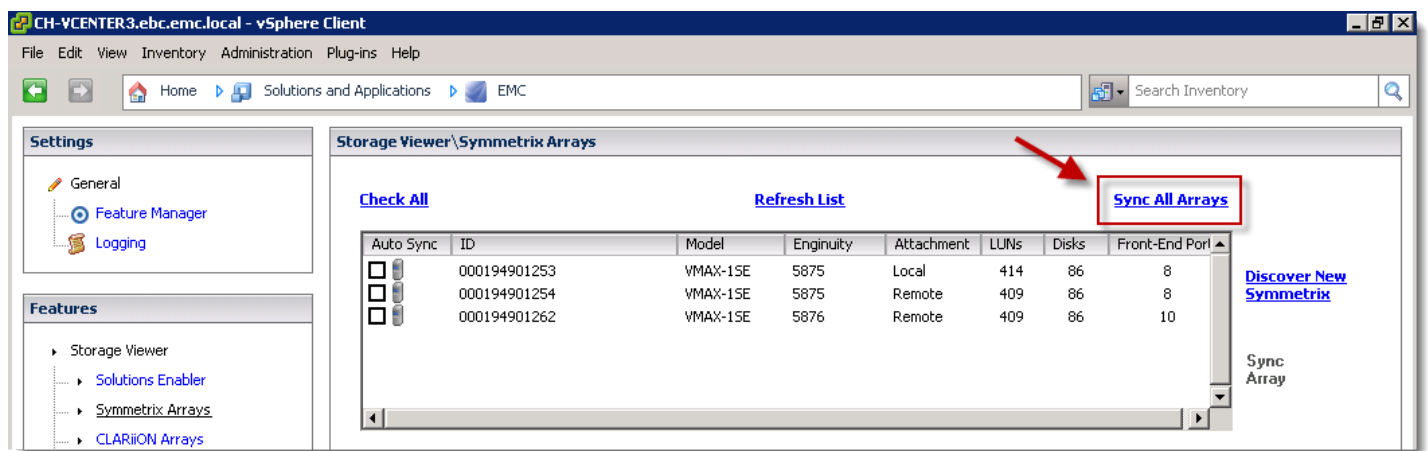


Figure 9. Synchronizing VSI information

3. Before space can be reclaimed, the previously used allocations must be overwritten with contiguous zeroes to allow the Symmetrix to be able to reclaim them. The simplest method of achieving this is by deploying an eagerzeroedthick virtual disk. When possible, the ESXi kernel will re-use previously allocated blocks before writing to new ones, which therefore makes this the most efficient method of zeroing out previously written segments. While users could simply create a virtual disk as large as possible, it is a quicker and far more efficient process if the user knows approximately how much space can be reclaimed so they can limit the size of the temporary virtual disk and the duration of this process. The maximum reclaimable space can be approximated with the use of the Storage Viewer feature of VSI. To find the reclaimable space:

² Since this process requires support for zero space reclamation, it can only be performed on Symmetrix VMAX arrays running Enginuity 5874.207 or higher.

1. Select a host with access to the VMFS volume that is to be reclaimed and click on the EMC VSI tab to open the VSI Storage Viewer.
2. Select the VMFS volume in the “Datastores” panel and then choose the “Storage Pools” button to see thin pool allocations of the thin device.
3. Find the value of “Free” column of the selected VMFS volume in the Datastores list
4. Identify the value of the “Free” capacity in the storage pool storage details screen of the thin device itself. Note that although there may be multiple storage pools listed, only the summary at the top contains the relevant information.

The third step shows the total free capacity of the VMFS datastore as vCenter observes it (space is freed when VMs are deleted or moved). The fourth step shows the free space of the thin device as the Symmetrix sees it (space is not freed as VMs are deleted or moved). The difference in these two numbers is the approximate amount of space that can be reclaimed and therefore should be the size of the eagerzeroedthick virtual disk that is deployed to zero out the VMFS volume dead space. The locations of the values are noted by red arrows in Figure 10.

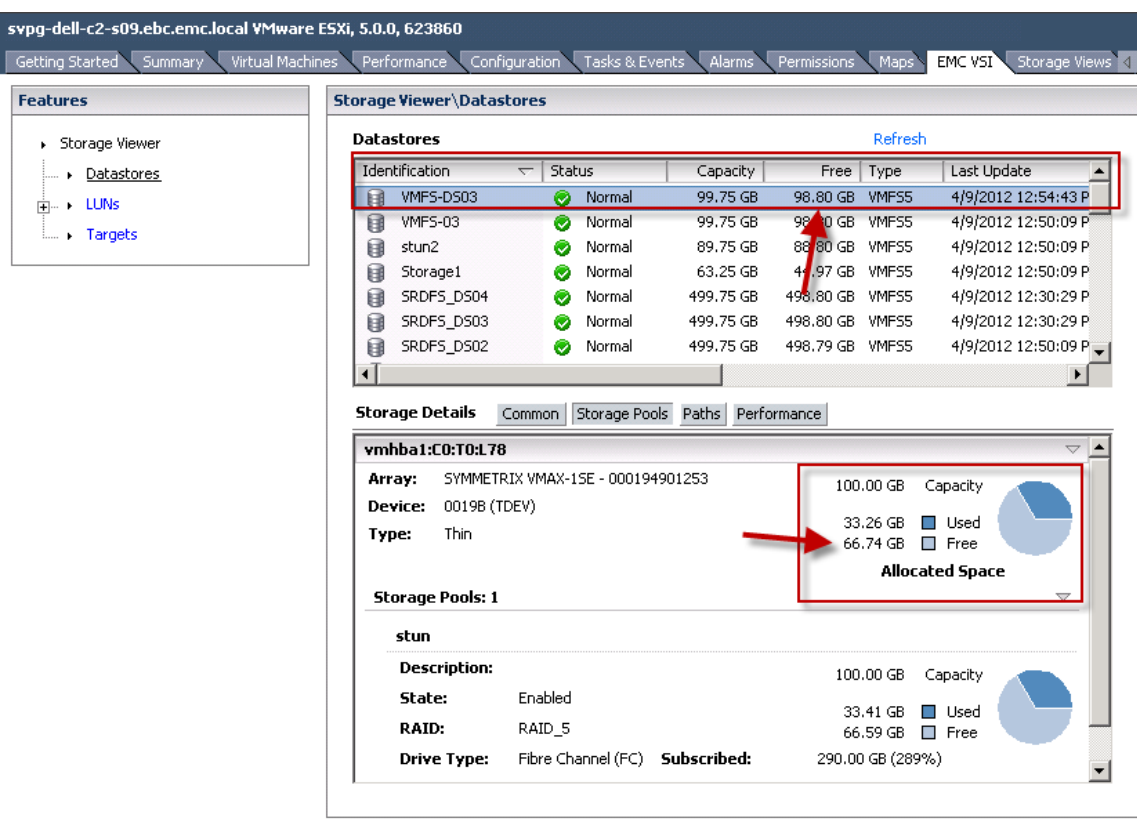


Figure 10. Identifying reclaimable space with VSI Storage Viewer

4. In the example shown in Figure 10 the reported free space on the VMFS volume is 98.80 GB and the reported free space on the thin device is 66.74 GB. This means that approximately 32.06 GB was at one point in use on the VMFS volume but has since been deleted (i.e. 32.06 GB of dead space). Therefore, up to 32.06 GB of space can be freed on the thin device.

NOTE: Due to the fact that this is merely a workaround, the efficiency of this process is not always 100%. This is because ESXi may not always perfectly re-use previously deleted space before creating new allocations. The ability to re-use blocks depends on a variety of factors (VMFS fragmentation etc.). The best practice therefore is to size the eagerzeroedthick virtual disk somewhat larger than the calculated free space to account for any discrepancy. In this case, to make up for any variations, the eagerzeroedthick virtual disk that will be deployed to zero the VMFS volume will be rounded up in size from 32.06 GB to 40 GB. This will increase the likelihood of reclaiming the maximum amount of space. In some situations the virtual disk might need to be increased by 100% or more to maximize reclaimed space.

5. At this point the temporary virtual disk can be deployed. The method of deployment (see step 5) is entirely at the discretion of the user. All that matters is the following:
 - a. The size of the virtual disk is adequately large.
 - b. The virtual disk is of type eagerzeroedthick.
 - c. The virtual disk is deployed on the correct VMFS volume.
6. It is important to note that this temporary virtual disk does not have to be associated with a virtual machine and can and should be deleted immediately after creation. However, since the vSphere Client does not offer a mechanism to create virtual disks that are not associated with a virtual machine, the virtual disk must be temporarily added to a virtual machine or a new temporary virtual machine must be created. However, VMware CLI tools such as vmkfstools can be used to create a virtual disk that is not attached to a virtual machine and also has the added benefit of being scriptable. These options are shown in Figure 11.

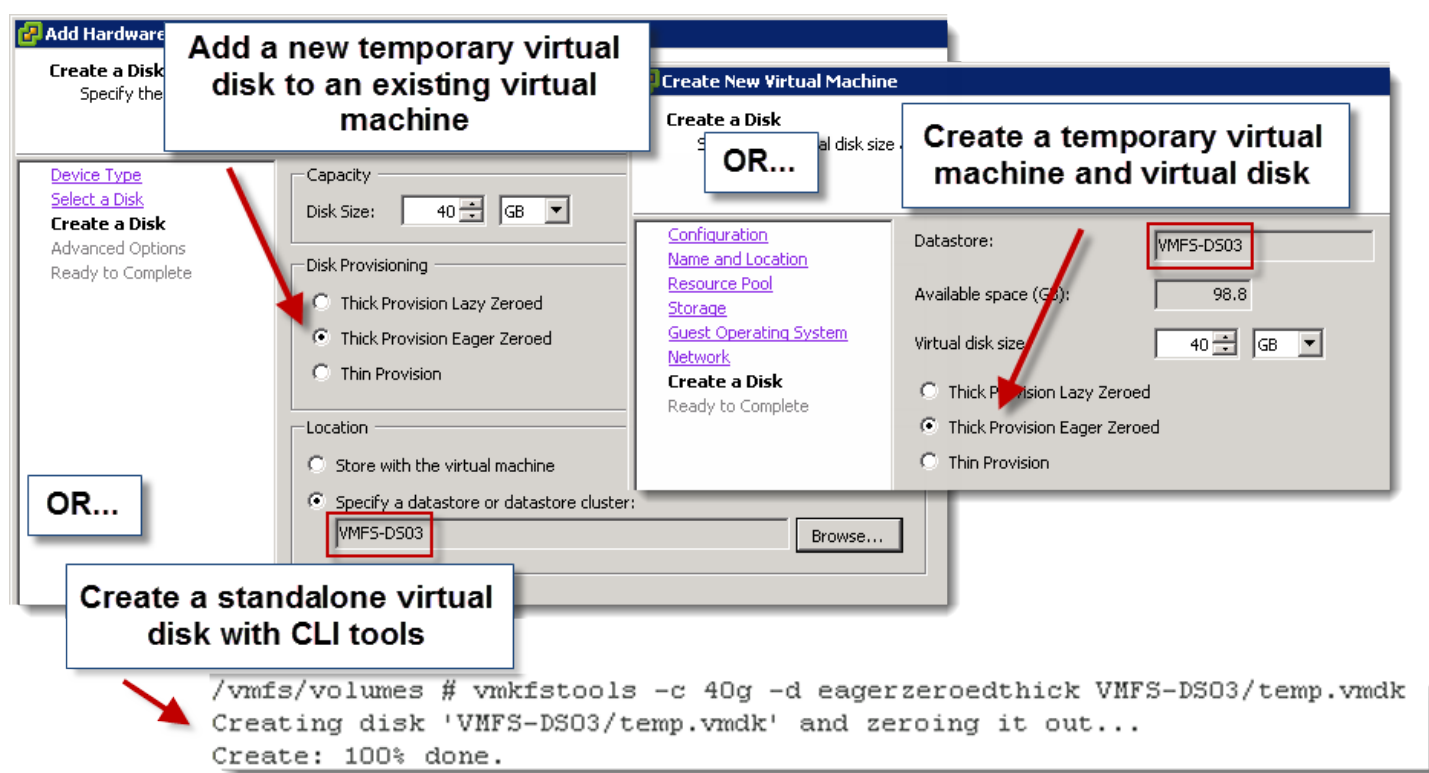


Figure 11. Virtual disk creation options

7. When a method has been chosen, the creation of the eagerzeroedthick virtual disk will zero out most, if not all, of the previously dead space on the thin device. For detailed information on what the virtual disk deployment process is doing refer to the section “Eagerzeroedthick” allocation format on page **Error! Bookmark not defined..**
8. The virtual disk (or virtual machine if a new one was created) should be deleted immediately after completion.
9. These newly-written contiguous zeroes can be detected by the Symmetrix and reclaimed with either Unisphere for VMAX or Solutions Enabler CLI. Unisphere is shown in Figure 12.

WARNING: If a storage administrator has purposely allocated space on this TDEV this process will de-allocate it unless they were persistent allocations. Furthermore, the zeroes of any eagerzeroedthick virtual disks on that VMFS volume will be reclaimed. Removing these zeroes and allocations could impact performance for that VMFS so it is important to understand what other activity is present on that datastore before executing a reclaim. Consequently, before running a reclaim, Storage vMotion any eagerzeroedthick virtual machines whose space you do not want reclaimed, off of the VMFS and move them back after the reclamation process has completed.

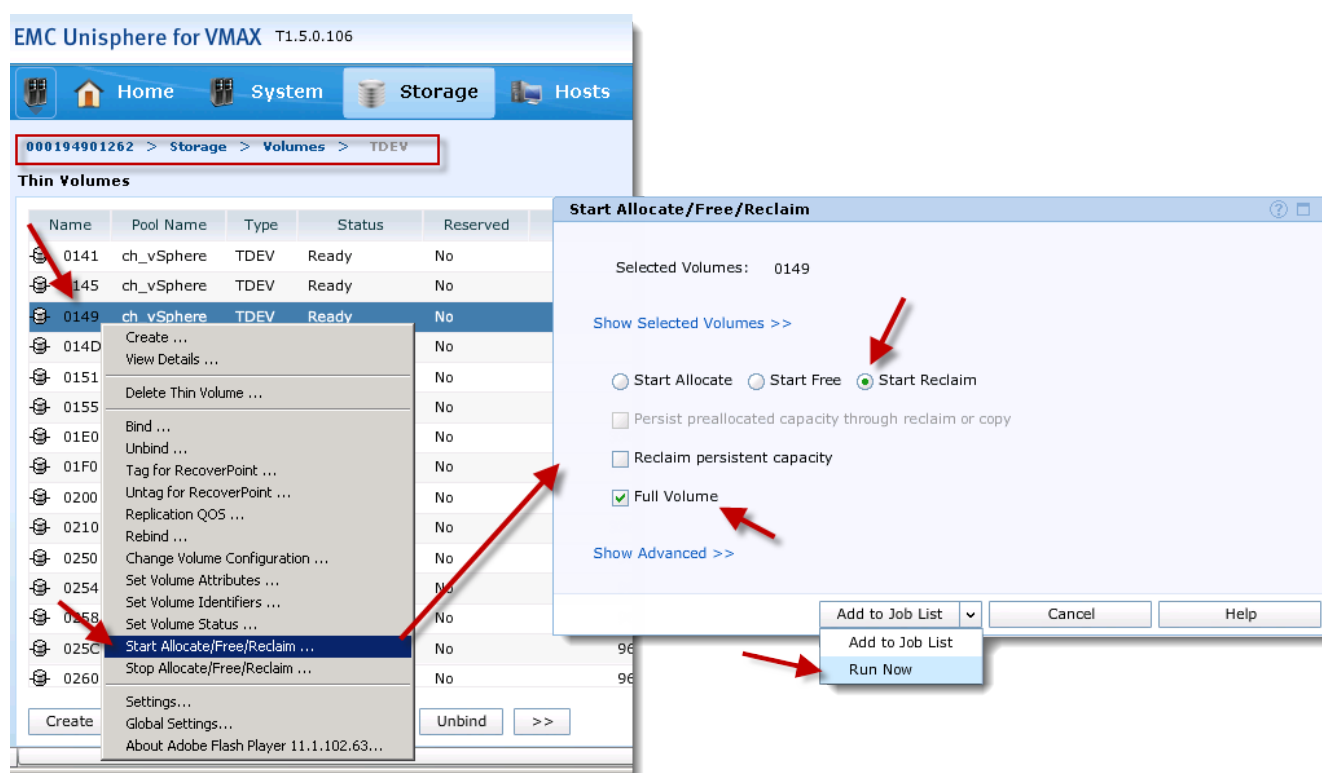


Figure 12. Reclaiming zeroes with Unisphere for VMAX

The reclamation process is not instantaneous as the entire device must first be scanned for thin extents that are filled with contiguous zeroes. For smaller devices this process may take minutes, for large devices perhaps hours. Furthermore the reclaim process is a low priority task on the Symmetrix directors, so it may slow down if the director(s) are under a heavy load from tasks with higher priority such as the servicing of significant host I/O.

Oracle

REDO log files

Oracle REDO log are pre-allocated files that store all the database change records before they are applied to datafiles. Oracle writes redo records to REDO log files in a circular fashion and therefore there is no reason for space reclamation on these devices.

Datafiles

When an Oracle datafile is first created, Oracle initializes the space in the file. Each initialized datafile contains a series of empty data blocks populated with a block header and other metadata. Due to the nature of file initialization in Oracle, there are no contiguous ranges of zero blocks with potential for space reclamation when targeting storage allocated by an initialized Oracle datafile.

In order to get improved capacity utilization benefits from a Virtual Provisioning infrastructure, a strategy for sizing datafiles, pools, and thin devices should be developed in accordance with EMC best practices.³

ASM considerations

When an ASM diskgroup is created, Oracle writes just a small amount of metadata but does not initialize with zeros or write to the whole newly created diskgroup space. Thus ASM is thin provisioning friendly and allows better storage utilization when used with thin LUNs (as described in EMC Virtual Provisioning best practices for Oracle and ASM). Therefore, there is no potential of zero space reclamation for devices provisioned to Oracle ASM outside of other scenarios described later in this white paper.

Reclaiming space from deleted Oracle objects

Whenever an Oracle object is dropped, the underlying storage is not zeroed out and cannot be reclaimed directly. ASM will, however, attempt to reuse space when new extents are added. This behavior reduces the need to reclaim deleted space in ASM. See also the “Space reclamation from previously used devices” section for reclaiming space on deleted Oracle objects on file systems.

Summary of Oracle considerations ASM considerations

Virtual Provisioning space reclamation improves storage utilization by releasing space consumed by contiguous zero blocks on the thin devices. Oracle file initialization and metadata limits such reclamation potential for Oracle database objects. However, Oracle applications can still improve storage capacity utilization by following the specific EMC best practices for Oracle in Virtual Provisioning environment, incorporate external tools to zero out previously used devices before they are given back to Oracle, or benefit from the TimeFinder/Clone thick-to-thin feature as described in the “Migration and replication” section of this white paper.

Microsoft SQL Server

Transaction log files

Active log files are formatted when a Microsoft SQL Server database is created. Every single page of the log files is written to at the time of creation so that the log files become fully provisioned when they are initialized. Because the log file will mostly contain contiguous zeros when first written, its storage has the potential to be reclaimed by Virtual Provisioning reclamation. Over time, as transactions are written to the log files and later truncated based on log backup or checkpoint operations, the log file will no longer contain all zero data. With this behavior in mind, it should not be expected that a log file allocated to a proper size and used for a given period of time will contain contiguous zeros sufficient enough to be reclaimed.

³ See the [Implementing Virtual Provisioning on EMC Symmetrix VMAX with Oracle 10g and Oracle 11g](#) white paper on EMC.com

Should a transaction log file be sufficiently oversized to where areas of the log were never used by SQL Server, there is the potential that these areas on disk could be reclaimed. While it is possible to reclaim unused portions of the SQL log file it is generally not recommended to reclaim zero space dedicated to a log device. Pre-written space on thin devices provides optimum write performance, important for the SQL log file. Additionally, pre-written space will guarantee thin pool allocations are dedicated and available for the SQL log file.

Database files

Prior to SQL Server 2005, SQL Server would fully initialize all datafile and transaction log file components, writing to every page in all database files and transaction log files. The process of writing to every page during initialization writes contiguous zeros to areas of the database not otherwise used by file metadata pages. If these zero pages remain untouched by SQL Server there is the potential that the areas on disk could be reclaimed.

As table and index information is written to a fully initialized datafile, areas of the database become allocated and used by non-zero user data. Assuming a database file was overallocated to where its initialized size is larger than the amount of user data in tables and indexes it may be possible to reclaim the remaining zero space, assuming it was never touched by SQL Server outside of the initialization process during creation or growth operations. Figure 13 provides a logical depiction of a file system and database file in this state.

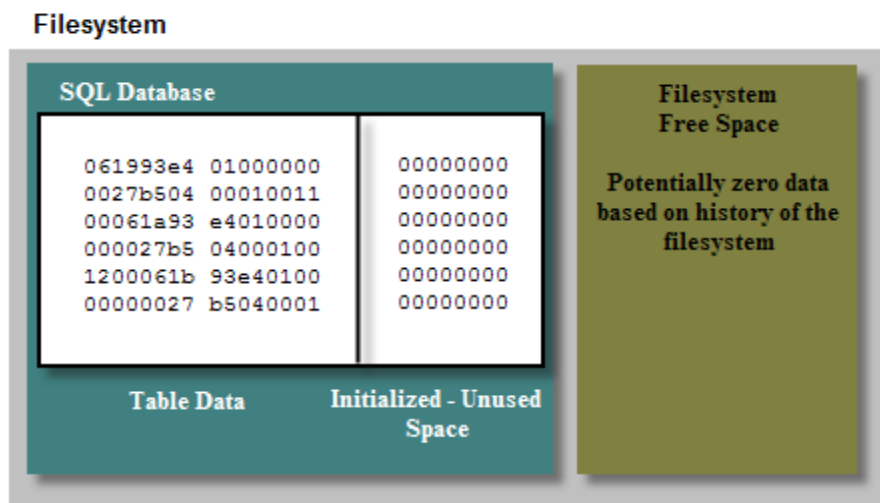


Figure 13. File system and SQL Server database file with initialized and unused space

Microsoft SQL Server 2005 introduced new functionality that altered the behavior during the database creation phase. As of SQL Server 2005, the database creation phase was no longer required to initialize all datafiles if Instant File Initialization could be utilized by the SQL Server environment.

Transaction log files will continue to be fully allocated with zeros even when using Instant File Initialization.

Instant File Initialization will be used automatically by SQL Server provided the service account, under which the SQL Server service is running, has the “Perform volume maintenance tasks” user rights assignment under the local security policy. By default, only users marked as Administrators have this permission.

Areas of a disk under which a sparse file is defined, as created by Instant File Initialization, are not zeroed. Whatever data existed on these areas of the disk, be it zeros or perhaps non-zeros based on the history of the file system and LUN, will remain. In the context of migrations, should a LUN-based copy be replicated to a target thin device from a source volume where a database was created with Instant File Initialization, the ability to reclaim zero written data will depend on the history of the LUN and file system. Figure 14 depicts this scenario.

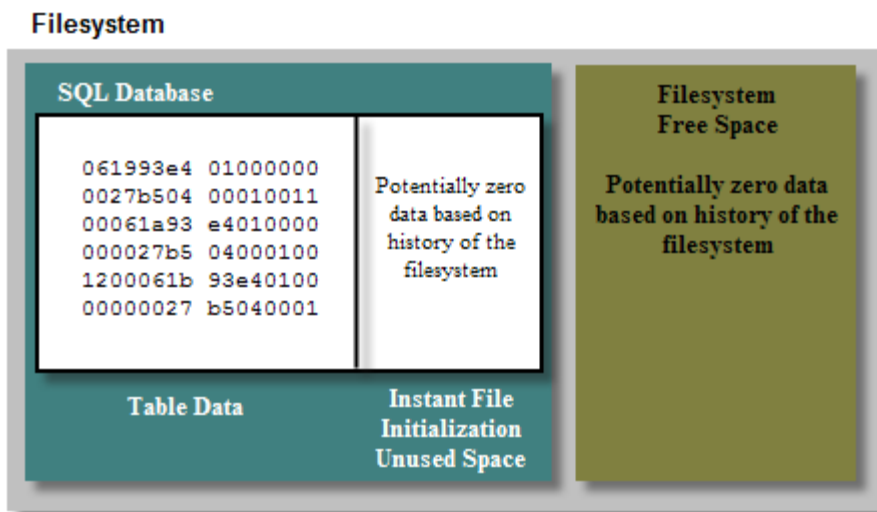


Figure 14. File system and SQL Server database created with Instant File Initialization

If user data ever resided in areas of the database file now considered free based on row or table delete operations, those areas will be non-zero and cannot be reclaimed. With SQL Server 2008 there is a stored procedure called `sp_clean_db_free_space`, which is intended to clean areas of a database file that are considered free. This procedure is also included with SQL Server 2005 SP3 but is undocumented. The `sp_clean_db_free_space` command will zero not only previously used areas of the database file, but also areas of the file that have never been used. This procedure will initiate writes and thin pool allocations on areas of the database file that have not been allocated where Instant File Initialization was used. As a part of the cleaning process each page touched by the procedure will leave behind database header information. While the majority of the page is zeroed, the page header left behind makes it impossible to reclaim the space cleaned by this utility. Because of this

behavior, it is not recommended to use `sp_clean_db_free_space` as a means to reclaim space within a SQL database file.

Additional information regarding Microsoft SQL Server and Symmetrix Virtual Provisioning can be found in the white paper *Implementing Virtual Provisioning on EMC Symmetrix with Microsoft SQL Server – Applied Technology* available on Powerlink®.

Microsoft Exchange Server

Microsoft Exchange log files

By default, Microsoft Exchange allocates log space in the form of either 5 MB files or 1 MB files, depending on the Exchange version. As the log files are filled, new log files are created. Over time, log files that have been committed to the database files will be deleted (truncated). This is normally performed following a successful backup operation. Log files are also deleted automatically should circular logging be enabled. As log files are deleted, space is freed within the NTFS volume; however, the space is not freed within the thin data pool.

For the thin pool space to be utilized efficiently, NTFS needs to reuse areas of the volume that had been previously written to and subsequently freed through the truncation process.

NTFS reuse rates may vary depending on operating system and service pack levels. Testing with Windows 2003 SP2 and Windows 2008 have shown more efficient use of newly freed log volume space on NTFS volumes when compared with earlier versions of Windows.

Microsoft Exchange database files

By default, an Exchange database file will start between 2 MB and 4 MB in size and will incrementally grow by 1 MB or 2 MB (depending on version) as additional space is needed. This auto-extend behavior is efficient from a thin pool perspective as only space immediately needed by the database file will be allocated. Based on this behavior, contiguous free or zero value space will generally only exist in the file system that holds the Exchange database file. How much of the file system and underlying LUN will contain zeros is dependent on the size of the database files and whether any other file system objects existed previously and were subsequently deleted.

One instance in which an Exchange database file may contain zero data is if page zeroing functionality is enabled. If page zeroing is enabled, once a deleted database object (mailbox messages, user mailboxes, etc.) has surpassed the date specified by the deleted item retention policy, the page on which it resided will be zeroed. Deleting data in this fashion ensures it is not recoverable, which may be a requirement in environments with sensitive data. Depending on the version of Exchange, page zeroing can be enabled and subsequently occur during a streaming backup operation, or as a part of the online maintenance window.

Testing with Exchange 2007, utilizing database page zeroing as a part of online maintenance, has shown that the pages cleaned by this process are not subsequently recoverable by Virtual Provisioning space reclamation. It should not be expected to be able to reclaim any data within an Exchange database file following page zeroing operations, even when large amounts of data are deleted.

Windows Server 2012

Windows Server 2012 supports the ability to detect thinly provisioned storage and issue T10 standard UNMAP or TRIM based reclaim commands against that storage. Windows Server 2012 utilizes the UNMAP specification when performing reclaim operations against a Symmetrix. Support of detection and reclamation with UNMAP has been added to Engenuity 5876.159.102.

Ensure the hotfix outlined in the following ETA is applied: ETA emc305807: Symmetrix VMAX, Symmetrix VMAXe: Compatibility requirement for Microsoft Windows Server 2012 TRIM support and Linux discard support at Engenuity 5876.

If a LUN is detected as a “Thin provisioned drive” in Windows Server 2012, by default reclaim operations will be performed under the following scenarios:

- When a volume residing on a thin provisioned drive is formatted with the quick option, the entire size of the volume will be requested to be reclaimed in real-time.
- When the “optimize” option has been selected for a volume as a part of a regularly scheduled operation, or as manually selected from the “Defragment and Optimize Drives” GUI. By default drives are automatically optimized on a weekly basis. Please note that Clustered Shared Volumes (CSV) cannot be optimized unless they are in redirected mode. An example of the optimize drives GUI can be seen in Figure 15.
- When the ‘optimize-volume’ powershell command is used with the “-retrim” option.
- When a file or groups of files are deleted from a filesystem, Windows will automatically issue reclaim commands for the area of the file system that was freed based on the file deletion. This holds true for CSV volumes as well, even if they are not in redirected mode. This automated method of reclamation reduces the need of running optimize operations; however to achieve full efficiency an optimize drive operation may still need to be run.

Windows Server 2012 supports reclaim operations against both NTFS and ReFS formatted volumes. Additionally, the new VHDX virtual disk format, native to Windows Server 2012, supports reclaim operations from within a Hyper-V virtual machine to a virtual disk. All reclaim operations supported on a physical LUN can be performed within and against a VHDX based virtual disk or against a pass-thru disk presented to a Hyper-V based virtual machine. Please see the Microsoft Hyper-V section for additional detail.

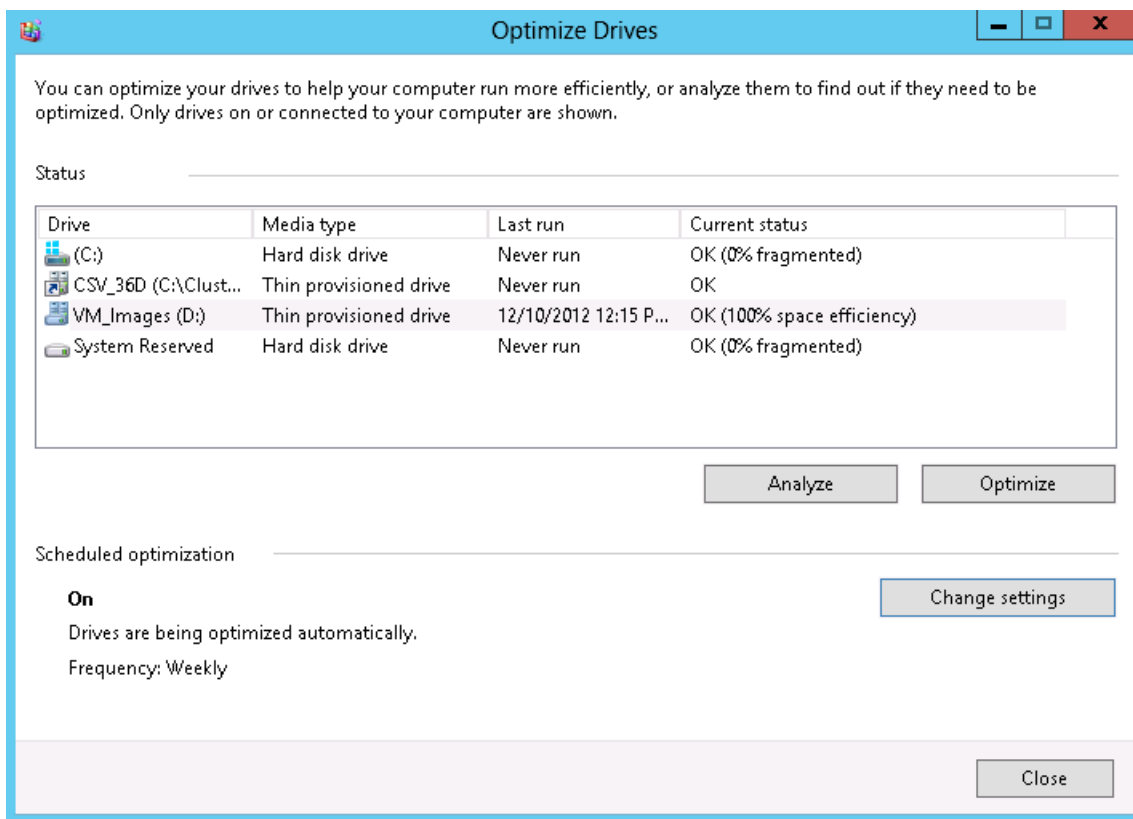


Figure 15. Windows Server 2012 Optimize Drives interface

The default behavior of issuing reclaim operations can be globally disabled on a Windows 2012 server. Specifically the “disabledeletenotify” parameter can be modified to prevent reclaim operations from being issued against all volumes on the server. This setting can be modified with the “fsutil” command line tool included with Windows Server 2012.

To disable reclaim operations run the following from an elevated command prompt:

Fsutil behavior set DisableDeleteNotify 1

To query the reclamation setting:

Fsutil behavior query DisableDeleteNotify

If DisableDeleteNotify = 0, this is the default and reclamation is enabled.

If DisableDeleteNotify = 1, space reclamation is disabled.

Lab testing has shown that both automatic reclamation and reclaim when running optimize volume operations are disabled when DisableDeleteNotify = 1. No reboot is required and the change goes into effect immediately.

Microsoft Hyper-V

Two primary representations of storage devices may be configured for a virtual machine in Microsoft Hyper-V environments. Either a storage device will be provisioned as a Virtual Hard Disk (VHD) and reside within an NTFS file system on the Hyper-V parent, or the device will be connected as a physical hard disk (a pass-thru storage device). The efficiency of pass-thru storage with regards to Virtual Provisioning as well as space reclamation will be fully dependent on the use of the hard disk by the virtual machine.

The efficiency of virtual hard disks from a Virtual Provisioning perspective, along with their use by a virtual machine, will be dependent on the kind of VHD format used. Three different types of VHD disks are available when configuring new or additional storage devices, represented in Figure 16. The choice between a “Fixed size” and “Dynamically expanding” format will generally be based on the storage utilization requirements, in as far as there is a difference in how storage is allocated for these two types. For this reason, the two selections will subsequently affect storage provisioning functionality such as that provided by Symmetrix Virtual Provisioning. A fixed sized VHD device will be fully written to at creation time. The writes that occur to a fixed size VHD will be contiguous zeros and can be reclaimed using Virtual Provisioning space reclamation dependent on the use of the VHD by the virtual machine.

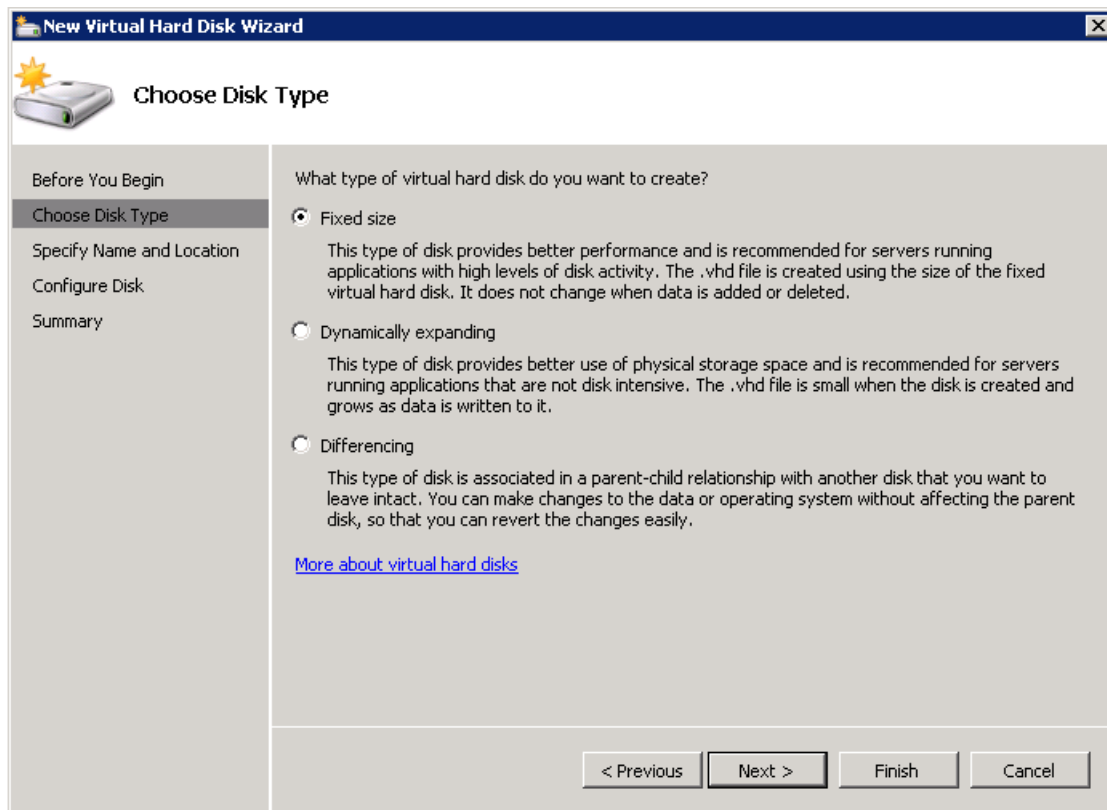


Figure 16. Hyper-V virtual hard disk options

Dynamically expanding VHD devices will not pre-allocate all storage defined for them; however these devices may suffer a slight degradation in performance as a result of the need to allocate actual storage when the operating system or applications within the virtual machine require more space. These storage allocations are simply those to acquire additional storage from the parent partition for the VHD device.

Beyond the format of the VHD files, the amount of space reclamation possible in a given Hyper-V environment will also depend on the NTFS volume that supports these files and its historical use. Please see the “File system considerations” section for additional details.

With Windows Server 2012 there is a new virtual hard disk format called VHDX. The VHDX format offers the same allocation methods as offered with the VHD format. As far as initial space allocation is concerned, “fixed” and “dynamically expanding” hard disks function the same with VHDX as was described earlier in this section for the VHD format.

What is unique about the VHDX format, however, is that it supports reclaim operations within the VHDX file. For example, if a fixed VHDX of 20 GB in size is created, 20 GB worth of zeros will be written to the thin device and marked as allocated and written in the thin pool. If that VHDX is then presented to a Windows Server 2012 based virtual machine and formatted with the quick option, the format process will issue reclaim commands to the VHDX file, which will be passed to the physical LUN and to the storage array. This will cause the full amount of the volume size to be reclaimed from within the fixed VHDX and from within the Symmetrix thin pool.

If this same 20 GB VHDX was presented to a Windows 2008 R2 virtual machine, the full 20 GB would still be allocated as Windows 2008 R2 does not issue UNMAP commands natively which could take advantage of the VHDX reclaim awareness.

Pass-thru storage to a Hyper-V virtual machine also supports reclaim operations.

Migration and replication

File system considerations

The efficiency of space reclamation following a migration process will be highly dependent on the nature of the data written to the original source file system. Most file systems are Virtual Provisioning friendly in that they will not write non-zero data to the entire contents of a volume during creation. Most allocations that occur will be due to user and application data being written to the partition or file system.

Depending on the use of the file system, objects may be added and deleted on a regular basis. Generally speaking, when a file is deleted from most file systems it is removed from being referenced or otherwise available for access; however the data on disk is still non-zero and represents the original contents of the file. Over time, as files are repeatedly added and deleted, it may be possible for the entire file system to contain non-zero data, even though from the operating system perspective there could be a large amount of free space available.

When devices containing file systems that exhibit a large number of add and delete operations are migrated, depending on the migration tool used, it is possible for the non-zero data representative of objects no longer referenced by the file system to be copied as a part of the migration process. The act of copying deleted but non-zero data will lead to lower space savings on thin devices targeted for migration and will also prevent Virtual Provisioning reclamation from freeing this space.

The ability of space reclamation to work efficiently following a migration from a thick to a thin device depends on the amount of contiguous zero blocks that may have been copied from the thick LUN. To maximize the zero blocks, external tools such as sdelete for Windows can be used on the thick LUN prior to a migration. Tools like sdelete will ensure a given area of a file system that was previously used and subsequently freed will contain zero data that can be reclaimed.

Open Migrator/LM

EMC Open Migrator/LM (OM) enables online data migration of Microsoft Windows, UNIX, or Linux volumes between any source and EMC storage. Open Migrator/LM host-based software helps to enhance data center efficiency by automating and simplifying data migration. Regardless of the purpose for the migration, be it consolidating servers, upgrading storage, or tuning performance, volumes remain online and fully available to critical applications during migration.

At the time of publication, versions of Open Migrator/LM supported on UNIX and Linux platforms perform a full volume copy from a source device to a target device. Should the target device be virtually provisioned, it should be expected that the target thin device will become fully allocated within the thin pool. The ability to reclaim the allocations done during the migration will be dependent on the amount of contiguous zero blocks that originally existed on the source device. The “File system considerations” section has more details and considerations.

Open Migrator/LM for Microsoft Windows can also perform full volume copies but has an additional feature referred to as “Sparse” copy. The sparse copy functionality of OM for Microsoft Windows allows for the replication of only known file system objects, specifically only the clusters of data marked as in use by NTFS. Previously deleted objects and other areas of the file system considered empty will not be migrated. This makes sparse migrations a compelling option when moving data from a thick environment to a thin environment. The only contiguous areas of zero data that will be replicated with OM using sparse copy will be those that exist within the files and areas of the file system marked as used by NTFS. To use our SQL Server example in Figure 13, this would mean the entire SQL datafile, including the initialized and zeroed regions of the file, would be migrated. The area of free space in the file system would not be copied. In this case, the zeroed region of the datafile could be reclaimed following the migration using Virtual Provisioning space reclamation

EMC Open Replicator

EMC Open Replicator for Symmetrix (ORS) enables remote point-in-time copies to be used for data mobility, remote vaulting, and migration between EMC Symmetrix and qualified storage arrays with full or incremental copy capabilities. Open Replicator leverages a SAN environment for copying data between Symmetrix and qualified storage arrays. Open Replicator can:

- Pull from source volumes on qualified remote arrays to a Symmetrix volume
- Push any live source Symmetrix volume to a target volume on a qualified array with incremental updates
- Perform online data migration from qualified storage to a Symmetrix with minimal disruption to host applications

Open Replicator can be used to perform remote replication between thin devices or between thin and regular devices. Managing thin device replication with Open Replicator is exactly the same as managing the replication of regular devices.

Thin devices can be used as control devices for hot and cold pull and cold push Open Replicator copy operations. If a push operation is done using a thin device as the source, zeros will be sent for any regions of the thin device that have not been allocated, or that have been allocated but to which have not been written.

Open Replicator can also be used to copy data from a regular device to a thin device. If a pull or push operation is initiated from a regular device that targets a thin device, with Enginuity 5874 or earlier, a portion of the target thin device, equal in size to the reported size of the source volume, will become allocated.

As has been previously discussed, the ability to reclaim portions of a thin device, following a migration with ORS, will be dependent on the ranges of contiguous zeros that existed on the source device prior to migration. If the source device contained contiguous ranges of zeros, they can be reclaimed using Virtual Provisioning zero space reclamation.

With Engenuity 5875, Open Replicator pull operations have been enhanced to detect contiguous ranges of zero data. The Symmetrix VMAX front-end adapter will scan incoming tracks for all zero data. If the 12 tracks that comprise a thin provisioning extent all contain zero data, that extent will be reclaimed immediately, thus maintaining the thin nature of the target device. This enables unused space on thick devices to be reclaimed while being migrated to a thin device on a Symmetrix VMAX. The feature works for both hot and cold pull operations and is supported for all systems that are currently supported by Open Replicator. Further, performing on-the-fly zero detection mitigates the need to run zero space reclamation as a post-migration activity.

TimeFinder/Clone thick to thin replication

Previous versions of Solutions Enabler and Engenuity have allowed for thin to thin replication. In the context of TimeFinder/Clone, what has been allowed is the replication from a source thin device to a target thin device within a Symmetrix. Starting with Engenuity 5874, TimeFinder/Clone can replicate standard volumes to thin volumes "sparsely" to ensure only written tracks are copied, reducing capacity requirements and TCO. Thin volumes also can be replicated to standard volumes, further improving mobility into and out of thin pools.

When a clone copy is made between a regular source device and a thin target device, device extents that have never been written to by a host will not be copied to the target volume. Following the clone copy, any thin device extents that were allocated on the clone target that contain all zeros can be reclaimed and added back to the target thin device's pool.

Symmetrix Remote Data Facility (SRDF)

SRDF is a frequently used option for migrating data between Symmetrix storage arrays. Beginning with Engenuity 5875 it is possible to use SRDF to migrate data from standard devices in a Symmetrix array running Engenuity 5671, 5773, or 5875 to thin devices in a Symmetrix VMAX running Engenuity 5875. In the case where the source array is running 5773 or later, the SRDF adapters (RAs) on the target Symmetrix VMAX will perform zero data detection on the fly on a per-track basis. Zero data detection is accomplished by looking at each track of data as it arrives to determine if it meets one of these two conditions:

- **If the never written by host (NWBH) indicator is set for the track:** Beginning with Symmetrix VMAX and Engenuity 5874 the special NWBH indicator is maintained in the metadata stored in cache for every track on every drive in the array. When this indicator is set for a track, it signifies that a host has never written any data to the track. If Engenuity receives a read request for a track with the NWBH indicator set, it can skip performing a read I/O to the disk and simply return all zeros to the host. If the track is part of a device participating in a local or remote replication session, the NWBH indicator will be set on the corresponding track of the target device.

- **If the track contains all zero data:** If the RA determines that an incoming track contains all zero data, by scanning that track's worth of data, it will indicate this in the track's metadata. If the 12 tracks that comprise a thin provisioning extent all contain zero data, that extent will be reclaimed immediately, thus maintaining the thin nature of the target device.

Zero data detection for SRDF is supported only in adaptive copy mode.

PowerPath Migration Enabler

EMC PowerPath Migration Enabler is a migration tool that enables nondisruptive or minimally disruptive data migration between storage systems or between logical units within a single storage system. Migration Enabler resides on the host and allows applications continued data access throughout the migration process. When pseudo device names are used, the migration can be nondisruptive. When native device names are used, a migration is minimally disruptive because applications must be reconfigured to use the new target device name that contains the migrated data. Migration Enabler works independently of PowerPath multipathing and can be used whether or not PowerPath is used for multipathing. If PowerPath is not already installed on the host, on some platforms a disruption may be required to install it. PowerPath Migration Enabler integrates with other technologies to minimize or eliminate application downtime. Supported technology types are:

- EMC Open Replicator for EMC Symmetrix. The online pull method is supported, in which data is copied through the SAN from a remote logical unit to a Symmetrix storage system.
- EMC Invista® in which a Virtual Volume encapsulates a Symmetrix or EMC CLARiiON source logical unit.
- EMC TimeFinder/Clone (Symclone), which migrates data between devices within a single Symmetrix system.
- Host-based copy (Host Copy), which works in conjunction with the operating system to migrate data from the source logical unit to the target logical unit.

The Host Copy technology type supports migrating virtually provisioned, or *thin*, devices. Host Copy automatically detects virtually provisioned CLARiiON and Symmetrix devices. When a thin target is detected PPME will scan the source disk for zeros and will not replicate these areas of the disk as a part of the migration process.

To again use our SQL example from Figure 13, PPME would only copy the non-zero regions of the database and may potentially replicate the free space region of the LUN depending on whether it contains non-zero data based on previous use. Due to this functionality, Virtual Provisioning reclamation should not be used following a thin-friendly migration using PPME Host Copy.

With technology types other than Host Copy, Migration Enabler support for thin devices is determined by the underlying technology's support. Please see the previous sections to understand TimeFinder/Clone and Open Replicator in the context of Virtual Provisioning and space reclamation.

Veritas Storage Foundation SmartMove

Veritas Storage Foundation from Symantec offers a “SmartMove” capability where, during migrations from thick to thinly provisioned storage, only used space from the file system perspective is migrated. Previously deleted objects and other areas of the file system considered empty will not be moved. SmartMove therefore helps to enable efficient, thick-to-thin migrations with the reclamation of unused and otherwise free space. More information regarding the use of Veritas Storage Foundation SmartMove with EMC Symmetrix can be found in the white paper *EMC Symmetrix VMAX with Veritas Storage Foundation* available at <http://powerlink.emc.com> and <http://www.symantec.com>.

Conclusion

Solutions Enabler 7.1 and Enginuity 5874, with the addition of space reclamation, enhanced the benefits of Virtual Provisioning in Symmetrix VMAX environments. The ability to reclaim unnecessary ranges of zero data from a thin pool enhanced storage efficiency and potential cost savings. Enginuity 5875, with the addition of on-the-fly zero reclamation with Open Replicator and SRDF, as well as the implementation of the WRITE_SAME unmap T10 standard, enabled even greater efficiencies for virtually provisioned environments. The Enginuity 5876 Q4 2012 service release advanced further, with support for the T10 UNMAP standard which enables integrated reclaim capabilities with operating environments like vSphere 5 and Windows Server 2012. The ability to utilize native application functionality as well as common migration tools, with less worry of wasted storage space, further enhances administrative efficiencies, including the time it takes to provision or migrate storage resources. Increased efficiency and savings, from both a storage and administrative perspective, will lead to an overall higher return on investment for critical data center resources.

References

White papers

New Features in EMC Enginuity 5874 for Symmetrix Open Systems Environments

New Features in EMC Enginuity 5875 for Symmetrix Open Systems Environments

Implementing EMC Symmetrix Virtual Provisioning with VMware vSphere

Technical notes

Best Practices for Fast, Simple Capacity Allocation with EMC Symmetrix Virtual Provisioning

Feature sheet

Symmetrix Virtual Provisioning Feature Specification

Product documentation

EMC Solutions Enabler Symmetrix Array Management CLI Product Guide

EMC Solutions Enabler Symmetrix Array Controls CLI Product Guide

EMC Solutions Enabler Symmetrix CLI Command Reference