EMC Solutions

Abstract

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

November 2013
# Table of contents

**Executive summary** ........................................................................................................................................... 5
  - Business case .................................................................................................................................................. 5
  - Solution overview ........................................................................................................................................... 5

**Introduction** .................................................................................................................................................... 6
  - Purpose .......................................................................................................................................................... 6
  - Scope ............................................................................................................................................................ 6
  - Audience ....................................................................................................................................................... 6
  - Terminology .................................................................................................................................................. 6

**Zero-space reclamation** ................................................................................................................................. 7
  - Overview ....................................................................................................................................................... 7
  - Restrictions ................................................................................................................................................... 7
  - Executing zero-space reclamation with Solutions Enabler ................................................................................. 8
  - EMC Unisphere for VMAX ............................................................................................................................ 9

**Space reclamation from previously used devices** ......................................................................................... 13
  - Executing online reclamation of previously used space ................................................................................ 13
  - VMware ESXi 5.0 U1 ..................................................................................................................................... 14
  - Windows Server 2012 ..................................................................................................................................... 14
  - Linux ............................................................................................................................................................... 14
  - Oracle ASM Reclamation Utility (ASRU) ........................................................................................................ 15
  - Veritas Storage Foundation ............................................................................................................................ 15
  - Zero reclamation of previously used space .................................................................................................... 15

**Application considerations** .......................................................................................................................... 17
  - Overview ....................................................................................................................................................... 17
  - VMware vSphere .......................................................................................................................................... 17
  - Eagerzeroedthick with ESXi 4.1+ and Enginuity 5875 ................................................................................... 18
  - Considerations for zero space reclamation in VMware environments .......................................................... 19
  - Reclaiming dead space with SCSI UNMAP .................................................................................................... 19
  - Reclaiming dead space with UNMAP on ESXi 5.0 U1 through ESXi 5.1.x .................................................. 20
  - Reclaiming dead space with UNMAP on ESXi 5.5+ ..................................................................................... 21
  - Reclaiming dead space on VMFS volumes without dead-space reclamation support .................................. 22
    - Synchronize VSI .......................................................................................................................................... 22
    - Find reclaimable space ............................................................................................................................... 23
    - Deploy the temporary virtual disk and reclaim the space .......................................................................... 24
  - Oracle ............................................................................................................................................................ 26
  - REDO log files ................................................................................................................................................ 26
  - Data files ....................................................................................................................................................... 26
ASM considerations .................................................................................................................. 27
Reclaiming space from deleted Oracle objects ........................................................................ 27
Summary of Oracle considerations .......................................................................................... 27
Microsoft SQL Server .............................................................................................................. 27
  Transaction log files .............................................................................................................. 27
  Database files ....................................................................................................................... 28
Microsoft Exchange Server .................................................................................................... 29
  Microsoft Exchange log files ................................................................................................. 29
  Microsoft Exchange database files ....................................................................................... 30
Windows Server 2012 .............................................................................................................. 31
Microsoft Hyper-V .................................................................................................................. 33

Migration and replication ........................................................................................................ 36
  File system considerations ..................................................................................................... 36
  EMC Open Migrator/LM ....................................................................................................... 36
  EMC Open Replicator ........................................................................................................... 37
  TimeFinder/Clone thick to thin replication .......................................................................... 38
  EMC Symmetrix Remote Data Facility (SRDF) .................................................................... 38
  EMC PowerPath Migration Enabler ...................................................................................... 38
  Veritas Storage Foundation SmartMove .............................................................................. 39

Conclusion ............................................................................................................................... 40
  Summary ............................................................................................................................... 40
  Findings ............................................................................................................................... 40

References ............................................................................................................................... 41
  White papers ......................................................................................................................... 41
  Product documentation ......................................................................................................... 41
  Other documentation ............................................................................................................ 41
Executive summary

Business case

Organizations continually search for ways to both simplify storage management processes and improve storage capacity utilization. When provisioning storage, administrators must consider and 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 over allocate in order to avoid the disruptions and management overhead of exhausting available capacity. Over allocation results in unused storage space; this is inefficient and generates unnecessary costs.

Solution overview

EMC® Symmetrix® 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. Virtual Provisioning can improve storage capacity utilization and simplify storage management by presenting an application with sufficient future capacity, which reduces the need to frequently provision new storage and avoid costly allocated but unused storage. Virtual Provisioning also simplifies data layout with automated wide striping that enables organizations to achieve a performance equivalent to standard provisioning but with less planning and labor required.

One of the goals when you deploy Virtual Provisioning is to ensure that 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 can 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 over-allocated and a data file 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, Virtual Provisioning now enables EMC 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 to reduce capacity requirements and total cost of ownership (TCO).

In addition to zero-space reclamation, VMAX enables host-based applications to request 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.
**Introduction**

**Purpose**

Many applications have the potential to write zeros to free space as part of the standard initialization, allocation, or migration processes. Depending on the way zeros are written, the potential exists to reclaim the storage space allocated as a result of these processes. This white paper discusses some of the most common situations that cause zeros to be written to storage devices.

**Scope**

This white paper outlines how to use EMC Solutions Enabler or the EMC Symmetrix Management Console to reclaim all-zero space and how to reclaim previously used non-zero space with host-based applications.

**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 EMC Symmetrix VMAX storage arrays.

**Terminology**

This white paper includes the terminology listed in Table 1.

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device</td>
<td>A logical unit of storage defined within a Symmetrix array.</td>
</tr>
<tr>
<td>Thin device</td>
<td>A host-accessible device that has no storage directly associated with it.</td>
</tr>
<tr>
<td>Data device</td>
<td>An internal device that provides storage capacity to be used by thin devices.</td>
</tr>
<tr>
<td>Thin device extent</td>
<td>The minimum amount of storage that can be mapped at a time to a thin device. Extent is also known as a “chunk.”</td>
</tr>
<tr>
<td>Data device extent</td>
<td>The minimum amount of storage that is allocated at a time when dedicating storage from a thin pool for use with a specific thin device.</td>
</tr>
<tr>
<td>Thin pool</td>
<td>A collection of data devices that provide storage capacity for thin devices.</td>
</tr>
<tr>
<td>Bind</td>
<td>The process by which one or more thin devices are associated to a thin pool.</td>
</tr>
<tr>
<td>Unbind</td>
<td>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.</td>
</tr>
<tr>
<td>Pre-allocating or pre-provisioning</td>
<td>A user-specified operation performed against a thin device for the purpose of reducing the operational impact of allocating extents or for guaranteeing a specified amount of storage for a thin device in a thin pool.</td>
</tr>
</tbody>
</table>
Zero-space reclamation

Overview

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 (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 EMC Open Replicator and EMC Open Migrator) copied all-zero, unused space to the target thin volume. Enginuity 5875 further extends the Virtual Provisioning space reclamation capabilities of the VMAX platform to include instant zero detection during Open Replicator and Symmetrix Remote Data Facility (SRDF®) migration operations.

Zero-space reclamation is an extension of the existing Virtual Provisioning space de-allocation mechanism. Previous versions of Enginuity and Solutions Enabler enabled 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 (or 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 while the targeted thin device is fully available to operating systems (OSs) and applications.

Starting the space-reclamation process produces a back-end disk adapter task that examines 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 are brought into the Symmetrix cache and examined to see if they contain all-zero data. If the entire extent contains all-zero data, the extent is de-allocated and added back into the pool, which makes it available for a new extent allocation operation. Should any portion of the extent contain non-zero data, none of the extent is reclaimed.

Restrictions

Some restrictions exist for zero-space reclamation with Symmetrix-based local and remote replication. Prior to Enginuity version 5876.159.102, zero-space reclamation was not supported on actively replicating SRDF volumes. The link had to be suspended prior to running the reclamation operation. With 5876.159.102 and later, the SRDF relationship remains active and the reclaim operations can proceed.

For fixed block architecture (FBA) environments, when executed against an R1 device, the zero-space reclamation only occurs on the R1 side. To achieve zero-space reclamation on the R2, you can run the command against the remote device with replication active on the pair. All Symmetrix arrays in the SRDF relationship must be running Enginuity version 5876.159.102. Additionally, zero-space reclamation will not be performed on tracks that participate in a local replication session, including EMC TimeFinder®/Clone and EMC TimeFinder/Snap.

Note: Prior to 5876.159.102, where you must suspend the SRDF relationship for zero-space reclamation, any tracks that are reclaimed against a source R1 device are also marked as invalid. When the SRDF links are resumed with an incremental establish, these tracks are copied to the R2 side where a zero-reclaim automatically occurs against the target device. With 5876.159.102 and later, where zero-reclaim can execute with SRDF active, the tracks reclaimed on the source R1 are not invalidated. This means that if the SRDF relationship is suspended during a reclaim operation against the source device, an incremental establish
Virtual Provisioning space reclamation can be executed using the Solutions Enabler version 7.1 or later `symconfigure` command line interface (CLI).

With older versions of Solutions Enabler, you could use the `symconfigure` command `free tdev` with the thin device as the target for reclamation. As a part of the `free tdev` syntax, you specified either a range of cylinders to be reclaimed that was representative of the thin device based on a beginning and end cylinder value or a total size to be reclaimed in cylinders, megabytes, or gigabytes. Typically, you targeted an entire thin device for reclamation by specifying a starting cylinder 0 and an ending cylinder `last_cyl`.

The `free tdev` command offered these reclamation options:

- **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.

- **Reclaim** performs the reclamation of pre-allocated but unused (unwritten) space and performs the reclamation of space where a thin device extent contains all zero data.

In more recent versions of Solutions Enabler, the `symconfigure` syntax has changed and the `start reclaim` command is recommended instead. Start reclaim performs the function of reclaiming both zero space and pre-allocated but unwritten space. A range of cylinders for the `start reclaim` syntax is not required. 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 starts a background process to perform the specified reclamation. Once initiated, the status of the thin device changes from a `bound` state to a `reclaiming` state. To view the status of the reclamation process and validate the reclaiming state, run the `symcfg` show command against a thin pool with the `-detail` option to view the detail of the thin pool where the thin device resides. This is the syntax of the command with the resulting output shown in Figure 1:

```
symcfg show -thin -pool FC -detail -sid 769 -gb
```
EMC Symmetrix VMAX Virtual Provisioning
Space Reclamation and Application Considerations

9

EMC Unisphere for VMAX

You can use EMC Unisphere® for VMAX to initiate and monitor the Virtual Provisioning space reclamation process.

To start space reclamation:

1. Select the Symmetrix system and go to Storage > Volumes to open the Volume Dashboard.

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

Note: The symcfg show output may differ depending on the Solutions Enabler version and may not show the Status column. As an alternative, you can use the symcfg list –tdev –pool <poolname> command to query the Status column or Status flag, which may also differ depending on the version.

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 purpose of verifying whether a thin device has entered a reclaiming state. The verify command is useful when scripting operations to verify that a given thin device has begun reclaiming. An example of the symcfg verify command is:

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

Once a thin device has completed the reclaim background task the status changes from reclaiming back to bound. You can use the symcfg show and symcfg verify commands to verify that a thin device has completed reclamation and returned to a bound state.

EMC Unisphere for VMAX
2. From within the dashboard expand **Virtual Volume**, select **TDEV**, and then click **View**, as shown in Figure 2.

![Unisphere for VMAX Volumes Dashboard](image)

**Figure 2.** Unisphere for VMAX Volumes Dashboard

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

5. To begin the reclaim background process, from the list box select Add to Job List or Run Now, as shown in Figure 4.

If you select Add to Job List, you must schedule or start the job from System > Job List.
After the reclaim job is started, the reclamation process operates as a background task on Symmetrix.

To verify that reclamation has started for a given thin device:

1. Under Storage, navigate to the Thin Pools window.
2. Select the appropriate pool and click View Details.
3. From the details window, look for related objects and click Bound Volumes.

In the Bound Volumes window, shown in Figure 5, Status shows the reclaiming state of the thin device.

Once the reclamation process is complete, the status of the device returns to Bound. Refresh the window to reflect the change in the status upon completion.
Space reclamation from previously used devices

**Executing online reclamation of previously used space**

Enginuity 5875 introduced support for the T10 industry-standard SCSI command WRITE SAME that includes an UNMAP bit. The WRITE SAME command with the UNMAP bit enables operating systems and applications to communicate a range of logical block addresses (LBAs) to the VMAX that are no longer needed. If the LBA range covers an entire Virtual Provisioning extent, then the extent can be reclaimed.

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

Some device types or configurations can cause the UNMAP command to be blocked or invalid for the entire device:

- UNMAP can be partially blocked by thin devices that have persistent allocations. VMAX accepts the UNMAP command but respects persistent allocations. If the specified UNMAP range has persistent and non-persistent allocations, VMAX de-allocates the non-persistent allocations and leaves the persistent allocations.

- UNMAP is not supported on:
  - Devices in use by Open Replicator for Symmetrix (ORS)
  - Devices in use by EMC RecoverPoint®
  - Encapsulated Federated Tiered Storage (FTS) devices
  - Virtual devices (TimeFinder/Snap targets)
  - Devices currently involved in duplicate write sessions (this is a temporary state; these sessions are used during meta reconfiguration and expansion)
  - TimeFinder/VP Snap targets

**Note:** 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 with UNMAP capabilities of Symmetrix VMAX help to enable the reclamation of areas of a thin device that were previously used and deleted without the need to zero out areas of a file system or otherwise unbind the thin device.

In some instances, you might need to pre-allocate storage to thin devices. Applications or operating systems that use UNMAP or WRITE SAME with UNMAP can reclaim pre-allocated areas of thin devices by default. In order to prevent pre-allocations from being reclaimed unexpectedly, you can mark the allocations as **persistent** for thin devices within the Symmetrix array. Figure 6 gives an example of how a thin device can be pre-allocated persistently using Unisphere.
Figure 6. Persistently pre-allocating thin device capacity

VMware ESXi 5.0 Update 1 introduced support for the SCSI UNMAP command, which issues requests to de-allocate extents on a thin device, called “dead space reclamation.” Dead space reclamation offers the ability to reclaim blocks of a thin-provisioned logical unit number (LUN) on the array using a VMware command line utility. VMware vSphere provides additional details.

Windows Server 2012 provides identification for thin-provisioned storage and file-system-aware reclamation capabilities, called TRIM in Microsoft documentation. When a file is deleted from a file system, Windows Server 2012 can issue reclaim requests to the underlying storage. Windows also issues real-time reclaim operations when formatting a file system on a thin LUN. Windows Server 2012 uses UNMAP when performing reclaim operations against a Symmetrix array. EMC recommends that you use Enginuity version 5876.163.105 or later with Windows Server 2012 when the storage reclamation feature is used. Windows Server 2012 and Microsoft Hyper-V provide additional details.

Linux

Starting with Linux kernel 2.6.33 the ability to reclaim storage was added natively to Linux distributions. Linux terminology refers to reclaim operations as “discard.” Linux discard operations can occur automatically as files are removed from file systems or manually as initiated by a user or script.

Storage reclamation is disabled by default in Red Hat Enterprise Linux (RHEL) and SUSE Linux distributions. You can enable the automatic discard feature by mounting the file system with the discard option in /etc/fstab or when you mount a file system manually, using the following command:

```
mount -t ext4 -o discard /dev/emcpowera /my_filesystem
```
Manual discard operations can be called by issuing the `fstrim` command against the mounted file system.

Discard for SCSI devices can use UNMAP if it is supported by the array. If UNMAP is not supported, WRITE_SAME with UNMAP will be used.

Symmetrix currently supports reclaim operations from Linux with RHEL 6.2 with hotfix BZ805519 and RHEL 6.3 or later. RHEL requires ext4 file systems. SUSE Linux Enterprise Server (SLES) 11 SP2 or later is also supported with the XFS file system.

Symmetrix support requires EMC VMAX 10K V2 (Systems with SN xxx987xxxx,) EMC VMAX SP or EMC VMAX 40K hardware that have SPC-3 advertised support for Linux file system reclamation. EMC recommends that you install Enginuity version 5876.163.105 or later prior to enabling discard functionality on Linux distributions.

**Note:** If you are using earlier versions of Enginuity, ensure that 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 Enginuity 5876.

---

**Oracle ASM Reclamation Utility (ASRU)**

Automatic Storage Management Reclamation Utility (ASRU) is a free tool developed and distributed by Oracle to help users reclaim previously allocated but unused storage space. The EMC Oracle ASRU Integration Module is a plugin for the Oracle ASRU tool that enhances ASRU’s support and performance on EMC storage arrays.

The latest version of the Integration Module uses the SCSI WRITE SAME with UNMAP capabilities of storage arrays to achieve a more efficient way for thin device space reclamation with Oracle Automatic Storage Management (ASM). The use of WRITE SAME with UNMAP eliminates specified blocks of the thin device immediately instead of writing zeroes to be freed later by zero space reclamation.

More information regarding ASRU can be found in the appropriate EMC integration module release notes and in *Implementing FAST VP and Storage Tiering for Oracle Database 11g and EMC Symmetrix VMAX*, both available on [EMC Online Support](http://www.emc.com).

---

**Veritas Storage Foundation**

Veritas Storage Foundation offers a reclamation facility that uses the WRITE SAME with UNMAP specification inherent in the Symmetrix VMAX with Enginuity 5875 or later. 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 Storage Foundation, see the appropriate hardware compatibility lists on [http://www.symantec.com](http://www.symantec.com).

---

**Zero reclamation of previously used space**

Sometimes devices are no longer needed and can be reused for other applications. Such devices continue to retain stale non-zero data from prior use and do not allow reclamation of unused storage space. Better capacity utilization and cost savings can be realized if the devices are zeroed out using operating system tools prior to re-provisioning.

**Note:** Prior to using such tools, you must ensure that the devices do not contain any active application data.
On UNIX and Linux systems, you can use a command like `dd` to zero out the disk blocks. If an entire partition is re-provisioned then you can use `dd` to zero out the whole partition. Once allocated space on the device has zeros in contiguous blocks, you can use space reclamation to reclaim the zeroed (and unused) storage in the thin pool. For example,

```
        dd if=/dev/zero of=/dev/emcpowerdf1 bs=1024 count=1024000
```

zeros out 1 GB of space on the first partition on device `/dev/emcpowerdf`.

On Windows 2003 and 2008 systems, you can use the `diskpart clean all` command to zero out all areas of a specified disk. With Windows 2008 only, a non-quick or full format also writes zeros to the targeted volume.

Additionally, as discussed in File system considerations, 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 useful for future replication from thick to thin volumes, such as with Open Replicator or TimeFinder/Clone. The target thin devices in these use cases 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 is to unbind and rebind the device 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 steps can be used to reclaim space:

1. **Change device state to `not_ready` to prevent host access:**
   
   ```
   symdev -sid 191 -RANGE 0297:0297 not_ready:
   ```

2. **Unbind the device from the thin pool:**
   
   ```
   symconfigure -sid 191 -cmd "unbind tdev 0297 from pool Oracle1 ;" commit
   ```

3. **Rebind the device to the thin pool:**
   
   ```
   symconfigure -sid 191 -cmd "bind tdev 0297 to pool Oracle1 ;" commit
   ```
Application considerations

Overview

At times of initialization or as a part of cleanup operations, applications might write contiguous areas of zeros to data files or file systems. The following sections describe situations when commonly used applications 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 detrimental to the space-saving benefit of Virtual Provisioning. The ability to perform space reclamation on thin devices hosting these zeroed-out virtual disks alleviates this issue. In VMware vSphere, several situations exist where performing space reclamation on thin devices is helpful.

vSphere offers various ways of formatting virtual disks and has integrated these options into VMware vCenter. The allocation options that the VMware vSphere Client provides for creating virtual disks are listed in Table 2.

Table 2. Allocation options when creating new virtual disks on a VMware datastore

<table>
<thead>
<tr>
<th>Allocation mechanism (Virtual Disk format)</th>
<th>VMware kernel behavior</th>
<th>Effect of zero reclaim</th>
</tr>
</thead>
<tbody>
<tr>
<td>zeroedthick</td>
<td>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.</td>
<td>No effect unless guest OS has written contiguous zeros</td>
</tr>
<tr>
<td>eagerzeroedthick</td>
<td>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 results in equivalent storage use in the thin pool.</td>
<td>Reclaims 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 are also reclaimed.</td>
</tr>
<tr>
<td>thin</td>
<td>This allocation mechanism does not reserve any space on the VMware virtual machine file system (VMFS) on creation of the virtual disk. The space is allocated and zeroed on demand.</td>
<td>No effect unless guest OS itself has written contiguous zeros</td>
</tr>
</tbody>
</table>

In vSphere, you can choose these virtual-disk allocation options, enabling them to deploy virtual disks in a manner compatible with the space-saving benefit of Virtual Provisioning.

Figure 7 shows that if a new virtual machine is created with **Thick Provision Eager Zeroed** selected, the virtual machine is created using the “eagerzeroedthick” allocation mechanism; if placed on a thin device, it results in a complete allocation...
because of the zeroing of the virtual disk. Space reclamation can be used in this instance to alleviate the consumed space on the thin pool.

Figure 7. Virtual disk allocation mechanism options in vSphere

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) Virtual Provisioning extent, reclaiming those zeros is not recommended. In this case, the virtual disks of the virtual machines are purposely pre-allocated and those zeros (and more importantly the Virtual Provisioning allocations) should not be removed.

vSphere offers a variety of VMware vStorage application program interfaces (APIs) for Array Integration (VAAI) that enable you to offload specific storage operations to VMAX to increase overall system performance and efficiency. VMAX with Enginuity 5875 supports the following new VMware vStorage APIs:

- Full Copy
- Block Zero
- Hardware-Assisted Locking

The Block Zero primitive delivers hardware-accelerated zero initialization, greatly reducing common input/output (I/O) tasks such as creating new virtual machines. This feature is especially beneficial when you create fault-tolerant-enabled virtual machines. Having the array instead of the host perform bulk zeroing of a virtual disk 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 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 finishes the job of zeroing out the blocks in the background on the array.

This primitive has a profound effect on Virtual Provisioning when deploying eagerzeroedthick virtual disks. When this feature is enabled and combined with
Virtual Provisioning, the typical host zeroing that occurs when deploying disks using this allocation mechanism is offloaded to the array.

Furthermore, the Symmetrix array does not write the zeros to disk, but simply sets the track as *Never Written By Host* and allocates 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, you can de-allocate these allocated but unwritten tracks using the unwritten reclamation process. The same recommendations apply in this situation as ordinary eagerzeroedthick disks. Since the Virtual Provisioning extents are still allocated with Block Zero, the first write latency for the guest OS does not occur. If 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 Enginuity version 5875 and later and on a properly licensed ESXi host.

### Considerations for zero space reclamation in VMware environments

For vSphere, the following important considerations should be evaluated before performing reclams:

- 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, 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 policy reduces the risk of reclaiming the zeros from these virtual machines.

- Zeroing out by guest OSs must be taken into account before performing a zero reclaim. If the virtual machines’ OS or its hosted applications zero out files for a particular reason, you should consider any implications of removing those zeros before reclaiming them.

- EMC recommends using the EMC Virtual Storage Integrator (VSI) Storage Viewer feature to easily map virtual machines and their corresponding VMFS volumes to the correct underlying Symmetrix thin device. Double-checking storage mapping information with the VSI Storage Viewer eliminates the possibility of performing zero reclamation on the incorrect thin device.

### Reclaiming dead space with SCSI UNMAP

Dead-space reclamation offers the ability to reclaim blocks of a thin-provisioned LUN on the array via a VMware-provided command line utility.

The Symmetrix array only supports this functionality with Enginuity 5876.159.102\(^1\) and later. If you are running an earlier version of Enginuity, you can upgrade or else...

\(^1\) Enginuity 5876.159 and 5876.163 require an ePack for VAAI functionality. Refer to emc319797 for more information.
use the workaround process described in Reclaiming dead space on VMFS volumes without dead-space reclamation support.

You should execute a dead-space reclamation after a virtual disk is deleted or is migrated to a different datastore. 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 the VMware Storage API for Array Integration primitive, the VMAX array is informed that the blocks are no longer in use, which results in better reporting of disk space consumption and a much more efficient use of resources.

To confirm that the Enginuity version is correct using the ESXi host and that UNMAP is supported on a LUN, use the following command 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—in other words, the Symmetrix array is running a version of Enginuity that supports UNMAP from an ESXi host.

In order to reclaim space, ESXi 5.0 U1 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. For example:

```
cd /vmfs/volumes/<volume-name>
vmkfstools -y <percentage of deleted block to reclaim>
```

The percentage of deleted blocks can be configured between 1 and 99 percent.

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 shown in Figure 8. If the reclaim operation is interrupted, this temporary file might not be deleted automatically and you might have to delete it manually.
SSH session to ESXi server to reclaim dead space with vmkfstools

You should not use too large a percentage in the vmkfstools reclaim command, because the resulting balloon file temporarily consumes and reserves space on the datastore. It is possible that a large temporary balloon file will block deploying a new virtual disk or growing an existing disk. Therefore, EMC recommends that you use a reclaim percentage close to 60 percent and avoid using the maximum of 99 percent.

The time a vmkfstools -y operation can take will vary, 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 array is a low-priority task (so that it does not negatively affect host I/O performance), so a heavy workload on the device, pool, or the array itself can throttle the process and slow the reclamation.

With the release of ESXi 5.5, the process to reclaim dead space has changed. The process no longer relies on the use of vmkfstools; instead, it uses esxcli. This change allows for remote execution of space reclamation using remote ESX CLI tools. To reclaim space, issue this command against a particular VMFS volume:

```bash
esxcli --server=<server_name> storage vmfs unmap --volume-label=<volume_label>
```

Input the ESXi domain name system (DNS) name in the server parameter and the name of the VMFS volume in the volume label parameter. Unlike previous ESXi versions, esxcli does not create one large balloon file and then issue UNMAP to all of the blocks encompassed by it. Instead esxcli runs through multiple, smaller iterations of this process. This behavior eliminates the possibility of exhausting available space on the VMFS volume due to the size of the balloon file.

---

2 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.
Note: You can add the parameter --reclaim-unit to the command to control the number of VMFS blocks to UNMAP per iteration but EMC recommends that you leave it at the default by omitting this parameter.

If it is not specified, the command uses the default value of 200 blocks per iteration. Therefore, a 200 block balloon file (typically 200 MB due to the 1 MB VMFS 5 block size) is created during space reclamation. This process issues UNMAP to all free space regardless of whether it was previously used to ensure all possible space was reclaimed. Because of the nature of this method, VMware has relaxed the requirement to run this only during maintenance periods and advises that it can now be executed during normal operation. Regardless, EMC still recommends running reclaim procedures during off-peak workload periods.

Note: The ESXCLI UNMAP command does not return any output. The cursor moves to the next line and is not returned until the UNMAP process on the device is complete.

In versions of vSphere earlier than 5.0 U1 and Symmetrix arrays running versions of Enginuity earlier than 5876.159.102, thin device allocations cannot be reclaimed with the vmkfstools or esxcli command. If an ESX and Enginuity upgrade is not possible, the following process is a workaround\(^3\) to remove these allocations quickly and simply:

**Synchronize VSI**

1. Log into the vSphere Client and ensure that the VSI Storage Viewer plugin is installed.

2. To make sure VSI has the most up-to-date information concerning thin pool allocations, synchronize the local database supporting Storage Viewer:
   a. In the vSphere Client, navigate to Home > Solutions and Applications > EMC VSI.
   b. Under Features, select Symmetrix Arrays. Click Sync All Arrays, as shown in Figure 9.

\(^3\) Because this process requires support for zero-space reclamation, it can only be performed on Symmetrix VMAX arrays running Enginuity 5874.207 or later.
**Find reclaimable space**

Before space can be reclaimed, the previously used allocations must be overwritten with contiguous zeroes to enable the VMAX to reclaim them. The simplest method of achieving this is by deploying an eagerzerodthick virtual disk. When possible, the ESXi kernel will re-use previously allocated blocks before writing to new ones, which makes this the most efficient method of zeroing out previously written segments. While you could simply create a virtual disk as large as possible, it is a quicker and far more efficient process if you know approximately how much space can be reclaimed so you can limit the size of the temporary virtual disk and the duration of the process. The maximum reclaimable space can be approximated by using Storage Viewer.

To find the reclaimable space:

1. From the vSphere Client, select a host with access to the VMFS volume to be reclaimed and select **EMC VSI** to open the VSI Storage Viewer.

2. In **Datastores**, select the VMFS volume, and then click **Storage Pools** to see thin pool allocations of the thin device.

3. In the **Free** column of the selected VMFS volume in the datastores list, find the value.
   
   This shows the total free capacity of the VMFS datastore as vCenter observes it (space is freed when virtual machines are deleted or moved).

4. In the storage pool **Storage Details** window of the thin device, identify the value of the **Free** capacity.
   
   This shows the free space of the thin device as the Symmetrix array sees it (space is not freed as virtual machines are deleted or moved).

---

**Note:** Although there might be multiple storage pools listed, only the summary at the top contains the relevant information.

The difference in these numbers is the approximate amount of space that can be reclaimed and therefore should be the size of the eagerzerodthick 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.
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 (that is, 32.06 GB of dead space). Therefore, up to 32.06 GB of space can be freed on the thin device.

**Note:** Because this is merely a workaround, this process is not always 100 percent efficient 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 like VMFS fragmentation. The best practice 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 is deployed to zero the VMFS volume is rounded up in size from 32.06 GB to 40 GB. This increases the likelihood of reclaiming the maximum amount of space. In some situations the virtual disk might need to be increased by 100 percent or more to maximize reclaimed space.

**Deploy the temporary virtual disk and reclaim the space**

At this point, the temporary virtual disk can be deployed. When choosing the method of deployment, ensure that the following requirements are met:

- The size of the virtual disk is adequately large.
- The virtual disk is of type **eagerzeroedthick**.
- The virtual disk is deployed on the correct VMFS volume.
You have the following options:

- Since the vSphere Client does not offer a mechanism to create virtual disks that are not associated with a virtual machine, you must add the virtual disk to an existing virtual machine or to a new virtual machine.
- You can use VMware CLI tools such as vmkfstools to create a virtual disk that is not attached to a virtual machine, which has the added benefit of being scriptable.
- The temporary virtual disk does not have to be associated with a virtual machine and should be deleted immediately after completion.

These options are shown in Figure 11.

![Virtual disk creation options](image)

Figure 11. Virtual disk creation options

Complete these steps to reclaim the space:

1. Create the eagerzeroedthick virtual disk using one of the methods described. This process zeros out most of the dead space on the thin device.
2. After completion, delete the virtual disk (or virtual machine if a new one was created).
3. Newly-written contiguous zeroes can be detected by the Symmetrix and reclaimed using either Unisphere for VMAX or Solutions Enabler CLI. Unisphere is shown in Figure 12.
**Figure 12.** Reclaiming zeroes with Unisphere for VMAX

**IMPORTANT:** If you have purposely allocated space on this TDEV, this process will de-allocate it unless they are 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. It is important to understand what other activity is present on that datastore before executing a reclaim. Therefore, before running a reclaim, use Storage vMotion to move any eagerzeroedthick virtual machines whose space you do not want reclaimed off the VMFS, and then move them back after the reclamation process has completed.

The reclamation process is not instantaneous because the entire device must first be scanned for thin extents that are filled with contiguous zeroes. For smaller devices, this process might take minutes, for large devices, perhaps hours. The reclaim process is a low-priority task on the VMAX directors, so it may slow down if the directors 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 files are pre-allocated files that store all the database change records before they are applied to data files. Oracle writes redo records to REDO log files in a circular fashion, so there is no need for space reclamiation on these devices.

**Data files**

When an Oracle data file is created, Oracle initializes the space in the file. Each initialized data file 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 reclamiation when targeting storage allocated by an initialized Oracle data file.
In order to improve capacity utilization benefits from a Virtual Provisioning infrastructure, you should develop a strategy for sizing data files, pools, and thin devices in accordance with EMC best practices.

For more information, refer to Implementing Virtual Provisioning on EMC Symmetrix VMAX with Oracle 10g and Oracle 11g.

**ASM considerations**

When an ASM disk group is created, Oracle writes just a small amount of metadata but does not initialize with zeros or write to the whole newly created disk group space. ASM is thin-provisioning friendly and enables better storage utilization when used with thin LUNs (as described in EMC Virtual Provisioning best practices for Oracle and ASM). Therefore, there is little potential for zero-space reclamation for devices provisioned to Oracle.

**Reclaiming space from deleted Oracle objects**

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

**Summary of Oracle considerations**

Virtual Provisioning space reclamation improves storage utilization by releasing space consumed by contiguous zero blocks on 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 EMC best practices for Oracle in a Virtual Provisioning environment
- Incorporating external tools to zero out previously used devices before they are given back to Oracle
- Using the TimeFinder/Clone thick-to-thin feature as described in Migration and replication

**Transaction log files**

Active log files are formatted when a Microsoft SQL Server database is created. At the time of creation, every single page of the log files is written to so that the log files become fully provisioned when they are initialized. Because the log file contains mostly 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 no longer contains all zero data. Therefore, a log file allocated to a proper size and used for a given period of time will not contain enough contiguous zeros to be reclaimed.

If a transaction log file is large enough that areas of the log were never used by SQL Server, these areas on disk can potentially be reclaimed. While it is possible to reclaim unused portions of the SQL Server log file, EMC does not recommend reclaiming zero space dedicated to a log device. Pre-written space on thin devices...
provides optimum write performance, which is important for the SQL Server log file. Additionally, pre-written space will guarantee thin pool allocations are dedicated and available for the SQL Server log file.

**Database files**

Prior to SQL Server 2005, SQL Server fully initialized all data file 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, the areas on disk could potentially be reclaimed.

As table and index information is written to a fully initialized data file, areas of the database become allocated and used by non-zero user data. Assuming a database file was over-allocated so that its initialized size is larger than the amount of user data in tables and indexes, it is possible to reclaim the remaining zero space, if 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**

SQL Server 2005 introduced functionality that altered its behavior during the database creation phase. If the SQL Server environment could use Instant File Initialization, the database creation phase was not required to initialize all data files.

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

SQL Server automatically uses Instant File Initialization, provided the service account under which the SQL Server service is running has **Perform volume maintenance tasks** permission under the local security policy. By default, only 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
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 depends on the history of the LUN and file system. Figure 14 depicts this scenario.

Filesyste

![Filesystem Diagram](image)

**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 are 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` procedure zeros previously used areas of the database file and areas of the file that have never been used. This procedure initiates writes and thin pool allocations on areas of the database file that were not allocated when Instant File Initialization was used. As a part of the cleaning process, each page touched by the procedure leaves 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, using `sp_clean_db_free_space` as a means to reclaim space within a SQL Server database file is not recommended.

*Implementing Virtual Provisioning on EMC Symmetrix with Microsoft SQL Server – Applied Technology White Paper*, available on [EMC Online Support](https://www.emc.com/), provides additional information regarding Microsoft SQL Server and Symmetrix Virtual Provisioning.

**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 are deleted. This is normally done following a successful backup operation. Log files are
also deleted automatically if circular logging is enabled. As log files are deleted, space is freed within the new technology file system (NTFS) volume.

For Windows operating systems earlier than Windows Server 2012, the space is not freed within the thin data pool. For the thin pool space to be utilized efficiently prior to Windows Server 2012, NTFS must reuse areas of the volume that were previously written to and subsequently freed through the deletion process.

NTFS reuse rates may vary depending on operating system and service pack levels. Testing with Windows 2003 SP2, Windows 2008, and Windows 2012 has shown more efficient use of newly freed log volume space on NTFS volumes when compared with earlier versions of Windows.

Starting with Windows Server 2012, Microsoft added the ability to reclaim space for deleted items. Given the cyclical nature of log deletion and creation for Microsoft Exchange, it might be beneficial to disable automatic storage reclamation on Microsoft Exchange database servers. For example, a day's worth of logs may be deleted as a part of a nightly backup process. This space will automatically be reclaimed, only to be consumed again the next day thanks to the efficient space reuse of NTFS. This behavior leads to wasted SCSI commands and CPU cycles on the server and storage array.

Conversely, if large volumes of logs are created occasionally but not consistently—because of mailbox move operations, for example—it might be beneficial for you to leave automatic reclamation enabled. Windows Server 2012 provides more details on how to disable automatic reclamation or issue manual or scheduled reclaim operations.

**Microsoft Exchange database files**

By default, a new Exchange database file is between 2 MB and 4 MB in size and incrementally grows by 1 MB or 2 MB (depending on the version of Exchange) as additional space is needed. This auto-extend behavior is efficient from a thin pool perspective because only space needed immediately by the database file is allocated. Based on this behavior, contiguous free or zero value space will generally only exist in the file system and not in the Exchange database itself. How much of the file system and underlying LUN contains zeros is dependent on the size of the database files and whether any other file system objects existed previously and were subsequently deleted.

An Exchange database file can contain zero data if page zeroing functionality is enabled. In this case, once a deleted database object (mailbox message, user mailbox, and so forth) has surpassed the date specified by the deleted item retention policy, the page on which it resided is zeroed. Deleting data in this way 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 occurs during a streaming backup operation or as a part of the online maintenance window.

Testing with Exchange 2007 using 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. You cannot reclaim any data within an Exchange database file following page zeroing operations, even when large amounts of data are deleted.
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 uses the UNMAP specification when performing reclaim operations against a VMAX array. Support for thin awareness and reclamation is available starting with the Enginuity 5876 Q4 2012 Service Release. EMC recommends using the most current Enginuity release (Q2 2013 SR or later) prior to using Windows Server 2012 reclamation support.

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

- When a volume residing on a thin provisioned drive is formatted with the **quick** option. The quick option requests that the entire size of the volume be reclaimed in real-time.

- When the **optimize** option is selected for a volume as a part of a regularly scheduled operation or is manually selected from the **Defragment and Optimize Drives** GUI. By default, drives are automatically optimized on a weekly basis. Clustered Shared Volumes (CSV) cannot be optimized unless they are in redirected mode. An example of the optimize drives GUI is shown in Figure 15.

![Optimize Drives GUI](image)

**Figure 15.** Windows Server 2012 Optimize Drives interface

- When the **optimize-volume** PowerShell command is used with the **-retrim** option.

- When a file or group of files is deleted from a file system, Windows automatically issues reclaim commands for the area of the file system that was freed based on the file deletion. This also holds true for CSV volumes, even if they are not in redirected mode. This automated method of reclamation
reduces the need for running optimize operations; however, to achieve full efficiency, you might still need to run an optimize drive operation.

Windows Server 2012 supports reclaim operations against both NTFS and Resilient File System (ReFS) formatted volumes. Additionally, the Hyper-V virtual hard disk (VHDX) virtual disk format, native to Windows Server 2012, supports reclaim operations from within a Microsoft 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-through disk presented to a Hyper-V based virtual machine. Microsoft Hyper-V provides additional details.

The default behavior of issuing reclaim operations can be globally disabled on a Windows 2012 server. Specifically, you can modify the DisableDeleteNotify parameter to prevent reclaim operations from being issued against all volumes on the server. You can modify this setting 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, run:

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 equals 1. No reboot is required and the change goes into effect immediately.
Microsoft Hyper-V

You can configure these primary representations of storage devices for a virtual machine in Hyper-V environments:

- Provision the storage device as a Virtual Hard Disk (VHD) residing within a file system on the Hyper-V parent.
- Connect the device as a physical hard disk (a pass-through storage device or virtual Fibre Channel (FC) device)

The efficiency of pass-through storage or virtual FC storage in regard to Virtual Provisioning and space reclamation is 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, is dependent on the kind of VHD format used. Several different types of VHD disks are available when configuring new or additional storage devices, and are represented in Figure 16.

The choice between **Fixed size** and **Dynamically expanding** formats is based on the storage utilization requirements because of the difference in how storage is allocated for these types. The selection affects storage provisioning functionality. A fixed-size VHD device is fully written to during creation. This means all storage equal to the size of the VHD file is consumed within the targeted thin pool. The writes that occur to a fixed size VHD during creation are contiguous zeros and can be reclaimed using Virtual Provisioning space reclamation. The creation of the fixed device can also take a considerable amount of time given the requirement to write the full size of the file to the storage array.

The Offloaded Data Transfer (ODX) feature of Windows Server 2012 addresses the time it takes to create a fixed-size VHD by offloading the writing of repeating patterns to a storage device. If ODX is supported by the target storage array, as it is with VMAX systems running Enginuity 5876.228.145 or later, the creation of fixed-size VHD files (either VHD or VHDX) offloads the series of contiguous writes to the storage array. This increases the speed at which the VHD is created.

Another benefit of the ODX write offload capability in virtual provisioning environments is that the zeros that represent the fixed-size VHD are not allocated within the thin pool. This makes a fixed-size VHD file thin-friendly where ODX is available. The fixed-size VHD continues to show its full size within the file system—for example, if a 100 GB fixed-size VHD is created, a 100 GB is consumed within the file system—but that space is not consumed within the thin pool.
Dynamically expanding VHD devices do 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 those needed to acquire additional storage from the parent partition for the VHD device.

In addition to the format of the VHD files, the amount of space reclamation possible in a Hyper-V environment also depends on the volume that supports these files and the historical use of the space. File system considerations provides additional details.

The Windows Server 2012 Hyper-V VHDX format offers the same allocation methods as offered with the VHD format. Fixed and dynamically expanding hard disks function the same as in the VHD format.

What is unique about the VHDX format 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 (assuming ODX is not enabled in the environment). If the VHDX is then presented to a Windows Server 2012-based virtual machine and formatted with the quick option, the format process issues reclaim commands to the VHDX file, which is passed to the physical LUN and to the storage array. The result is that the full volume size is reclaimed from within the fixed VHDX and from within the VMAX thin pool.
If this same 20 GB VHDX was presented to a Windows 2008 R2 virtual machine, the full 20 GB would remain allocated because Windows 2008 R2 does not issue UNMAP commands natively that could take advantage of the VHDX reclaim awareness.
Migration and replication

File system considerations

The efficiency of space reclamation following a migration process is highly dependent on the nature of the data written to the original source file system. Most file systems are compatible with Virtual Provisioning because they do not write non-zero data to the entire contents of a volume during creation. Most allocations that occur are a result of 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 a file system, it is removed from being referenced and is unavailable for access; however, the data on the disk is still non-zero and represents the original contents of the file. Over time, as files are repeatedly added and deleted, it is possible for the entire file system to contain non-zero data, even though the operating system reports that a large amount of free space is 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 leads to lower space savings on thin devices targeted for migration and also prevents 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 are 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 ensure that a given area of a file system that was previously used and subsequently freed contains zero data that can be reclaimed.

ECM Open Migrator/LM

EMC Open Migrator/LM 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—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. If the target device is virtually provisioned, the target thin device becomes fully allocated within the thin pool. The ability to reclaim the allocations done during the migration is dependent on the amount of contiguous zero blocks that originally existed on the source device. File system considerations provides more information.

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 Open Migrator/LM for Microsoft Windows enables 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 are
not 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 are replicated with Open Migrator/LM using sparse copy are those that exist within the files and areas of the file system marked as used by NTFS. In the SQL Server example in Figure 13 on page 28, this means the entire SQL Server data file, including the initialized and zeroed regions of the file, is migrated. The area of free space in the file system is not copied. In this case, the zeroed region of the data file can be reclaimed following the migration using Virtual Provisioning space reclamation.

EMC Open Replicator for Symmetrix enables remote point-in-time copies to be used for data mobility, remote vaulting, and migration between VMAX arrays and qualified storage arrays with full or incremental copy capabilities. Open Replicator uses a SAN environment for copying data between the 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 array with minimal disruption to host applications

You can use Open Replicator 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.

You can use thin devices as control devices for “hot-and-cold pull” and “cold push” Open Replicator copy operations. If a push operation is performed using a thin device as the source, zeros are sent for any regions of the thin device that have not been allocated or that have been allocated but not written to.

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 version 5874 or earlier, a portion of the target thin device equal in size to the reported size of the source volume becomes allocated.

The ability to reclaim portions of a thin device following a migration with Open Replicator is 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.

In Enginuity version 5875, Open Replicator pull operations have been enhanced to detect contiguous ranges of zero data. The Symmetrix VMAX front-end adapter scans incoming tracks for all zero data. If the 12 tracks that comprise a thin provisioning extent all contain zero data, that extent is 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. Performing instant detection reduces the need to run zero space reclamtion as a post-migration activity.
**Note**: Open Replicator instant zero detection is not enabled by default with pull operations. The feature must be explicitly specified during the Open Replicator create operation.

---

**TimeFinder/Clone thick to thin replication**

Previous versions of Solutions Enabler and Enginuity enabled thin-to-thin replication. For TimeFinder/Clone, the replication from a source thin device to a target thin device within a Symmetrix array was allowed.

With Enginuity 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 are not 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.

**EMC Symmetrix Remote Data Facility (SRDF)**

Symmetrix Remote Data Facility (SRDF) is a frequently used option for migrating data between Symmetrix storage arrays. Beginning with Enginuity 5875, you can use SRDF to migrate data from standard devices in a Symmetrix array running Enginuity 5671, 5773, or 5875 to thin devices in a Symmetrix VMAX running Enginuity 5875. When the source array is running 5773 or later, the SRDF replication adapters on the target Symmetrix VMAX 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 conditions:

- **If the never written by host (NWBH) indicator is set for the track**: Beginning with Symmetrix VMAX and Enginuity 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 Enginuity 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 is set on the corresponding track of the target device.

- **If the track contains all zero data**: If the replication adapter scans an incoming track and determines that the track contains all zero data, it provides this information in the track’s metadata. If the 12 tracks that comprise a thin provisioning extent all contain zero data, the extent is reclaimed immediately, thus maintaining the thin nature of the target device.

**Note**: Zero data detection for SRDF is supported only in adaptive copy mode or during synchronization operations. If the SRDF state is “synchronized” or “consistent,” zero data detection does not occur.

**EMC PowerPath Migration Enabler**

EMC PowerPath® Migration Enabler (Migration Enabler) is a migration tool that enables non-disruptive or minimally disruptive data migration between storage systems or between logical units within a single storage system.
Migration Enabler resides on the host and enables applications continuous data access throughout the migration process. When pseudo device names are used, the migration can be non-disruptive. 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. Migration Enabler integrates with other technologies to minimize or eliminate application downtime. Supported technology types are:

- EMC Open Replicator for 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 contains a VMAX or EMC CLARiiON® source logical unit.
- TimeFinder/Clone, 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 thin devices. Host Copy automatically detects virtually provisioned CLARiiON and VMAX devices. When a thin target is detected, Migration Enabler scans the source disk for zeros and does not replicate these areas of the disk as a part of the migration process.

Using the SQL Server example from Figure 13 on page 28, Migration Enabler only copies the non-zero regions of the database and might 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-compatible migration using Migration Enabler Host Copy.

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

Veritas Storage Foundation from Symantec offers a “SmartMove” capability, which migrates only used space from the file system perspective during migrations from thick to thin storage. Previously deleted objects and other areas of the file system considered empty are not moved. SmartMove enables efficient, thick-to-thin migrations with the reclamation of unused and otherwise free space.

The *EMC Symmetrix VMAX with Veritas Storage Foundation White Paper*, available on EMC Online Support and [http://www.symantec.com](http://www.symantec.com), provides more information regarding the use of Veritas Storage Foundation SmartMove with EMC Symmetrix arrays.
Conclusion

Summary
EMC Solutions Enabler 7.1 and Enginuity 5874 with space reclamation enhance the benefits of Virtual Provisioning in EMC Symmetrix VMAX environments. The ability to reclaim unnecessary ranges of zero data from a thin pool enhances storage efficiency and provides potential cost savings.

Enginuity 5875 with instant zero reclamation with EMC Open Replicator and SRDF and the implementation of the WRITE SAME with UNMAP T10 standard enable additional efficiencies for virtually provisioned environments.

The advances provided in the Enginuity 5876 Q4 2012 service release with support for the T10 UNMAP standard enables integrated reclaim capabilities with operating environments like vSphere 5 and Windows Server 2012.

Findings
The ability to use native application functionality and common migration tools with less wasted storage space 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, leads to an overall higher return on investment for critical data center resources.
## References

### White papers
For additional information, see the white papers listed below.

- *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*

### Product documentation
For additional information, see the product documents listed below.

- *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*

### Other documentation
For additional information, see the documents listed below.

- *Best Practices for Fast, Simple Capacity Allocation with EMC Symmetrix Virtual Provisioning Technical Notes*
- *Symmetrix Virtual Provisioning Feature Specification Feature Sheet*