This technical note contains information on these topics:

Executive summary ................................................................. 4
Introduction .................................................................................. 4
  Audience .................................................................................. 5
  Conventions used in this document........................................ 5
Virtual Provisioning on Symmetrix: Implementation overview ...... 6
  Use cases.................................................................................. 8
Other notes about appropriate environments......................... 9
Performance ............................................................................. 10
Implementation considerations .............................................. 10
Benefits of Virtual Provisioning ............................................. 11
Oversubscribed thin pools ...................................................... 14
Thin-hostile environments ..................................................... 15
Data availability considerations of wide striping.................... 17
Virtual Provisioning configuration guidelines ......................... 17
Adding capacity to a thin pool that is running out of space....... 20
Automated pool rebalancing ................................................... 21
Ensuring that DATA devices are spread wide across the back end 21
Migrating to thin devices ......................................................... 22
Dynamic Cache Partitioning ............................................................. 23
Local and remote replication between thin devices ...................... 23
Local replication between thin and regular devices ..................... 24
Local replication between thin and larger thin devices ............... 24
Virtual Provisioning space reclamation ......................................... 24
Local replication with SRDF/A devices ........................................ 25
Data device protection ................................................................. 26
Data and thin device restrictions ................................................... 26
Thin pool restrictions ................................................................. 26
Thin and data device limits within a pool .................................... 27
Virtual Provisioning with Solutions Enabler ................................. 27
Configuring and viewing data devices and pools ...................... 27
Configuring, binding, and viewing thin devices ......................... 32
Adding additional data devices to a pool ................................. 35
Binding additional thin devices to a pool ................................. 38
Preallocating pool space for thin devices ................................. 39
Allocating pool space for thin devices ............................... 41
Freeing allocated pool space from thin devices ....................... 44
Setting a maximum subscription percent .................................. 45
Unbinding thin devices from a pool ........................................... 48
Disabling and removing data devices from a pool ................... 49
Deleting thin pools ................................................................. 51
Configuring thin metadevices .................................................. 52
Monitoring thin pool space ...................................................... 60
Local replication using TimeFinder/Snap ................................. 63
Local replication using TimeFinder/Clone ................................. 66
Remote replication using SRDF ................................................. 69
Remote replication using Open Replicator ............................. 74
Automated pool rebalancing ..................................................... 74
Space reclamation ................................................................. 78
Virtual Provisioning with Symmetrix Management Console.................. 84
Configuring and viewing data devices and pools.............................. 84
Adding additional data devices to a pool........................................ 88
Configuring, binding, and viewing thin devices............................... 90
Preallocating pool space for thin devices........................................ 92
Allocating pool space for thin devices.......................................... 94
Freeing allocated pool space from thin devices............................. 97
Setting oversubscription limit......................................................... 98
Unbinding thin devices from a pool............................................... 100
Disabling and removing data devices from a pool ......................... 101
Deleting thin device pools............................................................. 104
Configuring thin metadevices......................................................... 104
Monitoring thin pool space............................................................ 108
Local replication with TimeFinder/Snap......................................... 115
Local replication using TimeFinder/Clone...................................... 118
Remote replication using SRDF....................................................... 123
Automated pool rebalancing........................................................... 126
Space reclamation........................................................................ 131
Conclusion ................................................................................. 137
Appendix A: Terminology............................................................... 138
Basic Symmetrix terms................................................................. 138
Thin provisioning terms............................................................... 138
Appendix B: Virtual Provisioning and host file systems................. 140
Executive summary

EMC® Symmetrix® Virtual Provisioning™, which marks a significant advancement over technologies commonly known in the industry as “thin provisioning,” adds a new dimension to tiered storage in the array by improving capacity utilization and simplifying storage management. Symmetrix Virtual Provisioning integrates with existing device management, replication, and management tools, enabling customers to easily build Virtual Provisioning into their existing storage management processes.

Introduction

Organizations continually search for ways to both simplify storage management processes and improve storage capacity utilization. When provisioning storage for a new application, administrators must consider that application’s future capacity requirements rather than simply its current requirements. In order to reduce the risk that storage capacity will be exhausted, disrupting application and business processes, and to reduce the number of times new storage must be provisioned for that application, organizations often have allocated more physical storage to an application than is needed for a significant amount of time. This allocated but unused storage introduces acquisition and operational costs. Even with the most careful planning, it often is necessary to provision additional storage in the future, which could potentially require an application outage.

EMC Virtual Provisioning addresses these challenges. It builds on the base “thin provisioning” functionality, which is the ability to have a large “thin” device (that is, volume) configured and presented to the 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 the application with sufficient capacity for an extended period of time, reducing the need to provision new storage frequently and avoiding costly allocated but unused storage. Virtual Provisioning also simplifies data layout, with automated wide striping that enables organizations to achieve equivalent or potentially better performance than standard provisioning, with less planning and labor required. This technology is appropriate for all storage types in a tiered storage environment. It supports both local and remote replication and all RAID types. Symmetrix Management Console and the command line interface (CLI) are the primary management and monitoring tools.
Virtual Provisioning

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

Figure 1. Symmetrix Virtual Provisioning

Audience

This technical note is intended for the technology professional who works in an environment with Symmetrix DMX-3 and DMX-4, and EMC Symmetrix VMAX™ with EMC Enginuity™ arrays. It is specifically targeted at EMC field technical staff and EMC customers who need to understand how to implement Virtual Provisioning.

Conventions used in this document

An ellipsis (...) appearing on a line by itself indicates that unnecessary command output has been removed.

Command syntax, output, and examples appear in the Courier New font.
Virtual Provisioning on Symmetrix: Implementation overview

Symmetrix Virtual Provisioning introduces a new type of host accessible device called a thin device that can be used in many of the same ways that regular, host accessible Symmetrix devices have traditionally been used. Unlike regular Symmetrix devices, thin devices do not need to have physical storage completely allocated at the time the devices are created and presented to a host. The physical storage that is used to supply drive space for a thin device comes from a shared thin storage pool that has been associated with the thin device. A thin storage pool is comprised of a new type of internal Symmetrix device called a data device that is dedicated to the purpose of providing the actual physical storage used by thin devices. When they are first created, thin devices are not associated with any particular thin pool. An operation referred to as “binding” must be performed to associate a thin device with a thin pool.

When a write is performed to a portion of the thin device, the Symmetrix allocates a minimum allotment of physical storage from the pool and maps that storage to a region of the thin device including the area targeted by the write. The storage allocation operations are performed in small units of storage called “thin device extents.” Extents may also be called “chunks.” A round-robin mechanism is used to balance the allocation of data device extents across all of the data devices in the pool that are enabled and that have remaining unused capacity. The thin device extent size is 12 tracks (768 KB). That means that the initial bind of a thin device to a pool causes one thin device extent, or 12 tracks, to be allocated per thin device. So a four-member thin meta would cause 48 tracks (3078 KB) to be allocated when the device is bound to a thin pool.

When a read is performed on a thin device, the data being read is retrieved from the appropriate data device in the storage pool to which the thin device is bound. Reads directed to an area of a thin device that has not been mapped do not trigger allocation operations. The result of reading an unmapped block is that a block in which each byte is equal to zero will be returned. When more storage is required to service existing or future thin devices, data devices can be added to existing thin storage pools. New thin devices can also be created and associated with existing thin pools.

As noted, it is possible for a thin device to be presented for host-use before all of the reported capacity of the device has been mapped. It is also possible for the sum of the reported capacities of the thin devices using a given pool to exceed the available storage capacity of the pool. Such a thin device configuration is said to be oversubscribed.
In Figure 2, as host writes to a thin device are serviced by the Symmetrix array, storage is allocated to the thin device from the data devices in the associated storage pool. The storage is allocated from the pool using a round-robin approach that tends to stripe the data devices in the pool.

In Figure 3, as the free space in the thin storage pool becomes exhausted,
new data devices can be added to the thin storage pool. The new data devices are immediately added to the set of data devices over which the round-robin selection operates.

**Figure 4. Addition of a thin device**

Figure 4 shows that new thin devices can also be added to the thin pool.

**Use cases**

- To improve ease of use

Organizations using Virtual Provisioning should have well-understood application capacity growth requirements in order to avoid unexpected consumption of pool capacity. Organizations can improve ease of use in these environments. Data layout becomes simpler as automated wide striping provides equivalent or potentially better performance, with less planning and labor than is required with standard provisioning. In addition, users can accommodate growth more easily. Initial setup of regular devices and thin devices and pools involves many similar steps and processes. After this initial setup, fewer steps are required with Virtual Provisioning to make additional capacity available to the host.

While regular provisioning requires the same setup steps to be repeated—that is, creating a new device and presenting it to the host (mapping, masking, and so on). Virtual Provisioning requires that data devices are created and added to the pool. The thin device and its relationship to the application remains the same (provided it was made large enough at the outset).
Users can further simplify the growth process by creating command line scripts that automatically add data devices to a thin pool when pool capacity exceeds a percentage threshold. In either case users should closely monitor the space consumption of the thin device and thin pool.

- To improve capacity utilization

Organizations are able to reduce the amount of allocated but unused space in storage systems by adopting Virtual Provisioning, because multiple applications can share a common pool from which actual storage is allocated only as needed. The degree of utilization benefits will depend on the allocation methods of file systems and databases, which often allocate space before using it. Many file systems do not reuse the space associated with deleted files. In addition, many databases preallocate space and write zeros to it. This allocated but unused space cannot be shared in a thin pool.

Organizations should take these allocation methods into account and wherever possible leverage space-efficiency features such as Oracle Auto Extend and SQL Server Instant File Initialization.

Independent of these factors, Virtual Provisioning still can improve ease of use and improve performance via wide striping depending on the workload.

Virtual Provisioning can help ensure that premium Flash drive resources are highly utilized. Flash drives can offer both unparalleled response times and significant IOs/GB. Unlike traditional drives, throughput and response time in Flash drives do not steadily decline as utilization of the drive increases.

Other notes about appropriate environments

- General purpose performance requirements

Virtual Provisioning is appropriate for applications for which some performance variability can be tolerated. Some workloads will see performance improvements from wide striping with Virtual Provisioning, as outlined in the “Performance” section. However, when multiple thin devices contend for shared spindle resources in a given pool, and when utilization reaches higher levels, the performance for a given application and thin device might become more variable.

- Document and media repositories

Document repositories can be a strong fit for Virtual Provisioning, provided the environments meet the performance, growth, and other criteria outlined previously, as rapidly rising capacity requirements for these repositories create the opportunity for improved capacity.
Virtual Provisioning

utilization.

- Test/Development and QA
  Many organizations will see an opportunity to lower TCO by improving ease of use and capacity utilization for storage associated with test, development, and QA activities, which often can tolerate some level of performance variability.

- Control of data placement
  When full control of data placement is required, regular provisioning rather than Virtual Provisioning is appropriate. Users also have the option to fully preallocate space to a thin device if they prefer.

Performance

The performance implication of using Virtual Provisioning depends on the nature of the workload and the state of the thin device. In any thin device implementation there are response time and throughput overheads that are incurred the first time a write is performed on an unallocated region of a thin device. In the Symmetrix implementation these overheads are quite modest. The overhead applies primarily to the first write to a thin device extent and will tend to disappear altogether once the “working set” of a thin device has been written to.

The most important aspect to thin device performance to understand has to do with how thin devices are spread across the back end. As already discussed, the back-end layout of the storage underlying each thin device will tend to be widely striped, typically spanning a much greater number of drives than a regular device. In many cases, this aspect of thin devices can make it easier to avoid skews in the back-end workload. The result can be improved performance for certain random I/O workloads.

Care should be used when migrating an application whose back-end layout has already been carefully tuned, especially if the tuning effort included isolating the back-end resources used by certain portions of the workload and the application has stringent throughput or response time requirements. In such a case, simply migrating all devices to a single, large thin pool could cause performance degradation.

Implementation considerations

When implementing Virtual Provisioning it is important that realistic utilization objectives are set. Generally, organizations should target no higher than 60 percent to 80 percent capacity utilization per pool. A buffer should be provided for unexpected growth or a “runaway” application that consumes more physical capacity than was originally
planned for. There should be sufficient free space in the storage pool equal to the capacity of the largest unallocated thin device.

Organizations also should balance growth against storage acquisition and installation timeframes. It is recommended that the storage pool is expanded before the last 20 percent of the storage pool is utilized to allow for adequate striping across the existing data devices and the newly added data devices in the storage pool.

Thin devices can be deleted once they are unbound from the thin storage pool. When thin devices are unbound, the extents that have been allocated to them from the thin pool are freed, causing all data from the thin device to be discarded.

Data devices may also be disabled and removed from a storage pool. Disabling data devices with allocated extents will cause the data devices to drain and copy any allocated extents to the remaining, enabled devices in the pool. Therefore, data devices can be removed from a pool and repurposed without having to unbind thin devices that may have allocated extents associated with the data devices.

**Benefits of Virtual Provisioning**

The following describe some of the potential benefits of Virtual Provisioning:

**Less expensive to pre-provision storage**

The process of provisioning storage with regular devices typically requires the following activities:

- Determining how much storage is needed and to which target hosts and applications it must be presented.
- Executing a set of array management operations to create and present new LUNs or to enlarge existing LUNs.
- Executing a set of SAN management operations to make the new LUNs visible to the target hosts.
- Performing operations on target hosts to recognize new or enlarged LUNs.
- Executing management operations in the target applications to recognize new or enlarged LUNs.

The frequency of performing the tasks associated with storage provisioning can be greatly reduced with Virtual Provisioning by allocating thin devices that represent larger amounts of storage than immediately required. This may allow a number of future cycles of the
provisioning process outlined above, and the possible application interruptions involved, to be avoided. This is referred to as pre-provisioning.

The pre-provisioning approach can be applied using regular devices, but in this case the entire amount of storage (including the amount that is needed in the near term and the “extra” storage) must be physically configured and dedicated at the time the pre-provisioning operation is performed.

The use of thin devices makes it possible to pre-provision large devices without having to physically pre-provision drive space. In this case the storage administrator satisfies a provisioning request using thin devices that have reported sizes that exceed the amount of storage needed immediately, but which belong to thin pools configured with physical storage needed only for the near term. The physical storage needed to satisfy storage capacity that will be required later can be added into the thin pools at a later time.

Using this method, the administrator can determine when to add additional physical storage to the pool by monitoring the amount of storage that has been consumed in the pool. When it exceeds a predetermined threshold, the administrator would then add additional data devices to the pool. This has the effect of making storage available to the target applications without the need for host or application management operations normally required to present additional storage to hosts.

This approach entails the use of an oversubscribed configuration, which means that the storage administrator must monitor thin pool utilization carefully.

The use of oversubscribed pools may also allow the organization to defer purchase and installation of additional physical drives.

**Array operations involved in provisioning storage take less time**

The use of thin devices can reduce the time required to perform the array operations required to create the new devices or to expand already existing concatenated thin metadevices.

Because a thin device does not consume any actual drive capacity until it is bound to a pool, creating a thin device in advance will not consume any drive space. Since the bind operation is very fast, thin devices can be created ahead of time so that only the bind operation is needed during the course of a typical storage provisioning operation. This approach may only be manageable in an environment in which there are a fairly
small number of device sizes in use, but in such cases the time required to present new storage can be greatly reduced.

Even when thin pool space is preallocated, the allocation happens in the background. The time required for a bind operation with preallocation should be about the same as a bind with no preallocation. This is because the bind operation is performed first and, following the completion of the bind, the preallocate command is issued and the prompt is immediately returned. While the space is being preallocated, the thin device is available for use.

Pool-based view of back-end layout

In a properly designed thin device configuration, each thin pool contains storage of a particular class that is aligned with a defined business purpose. This, together with the fact that the thin device storage allocation mechanism tends to automatically widely spread thin devices across the thin pool, allows the storage administrator to view storage at a higher level of abstraction when planning the back-end layout of new devices. When using thin devices, the back-end layout planning operation consists largely of deciding what class of storage is required and which business group’s resources should be used.

Easier implementation of device-level wide striping

With regular Symmetrix volumes, wide striping can be achieved at the device level using striped metadevices. In that case it is the storage administrator’s responsibility to ensure that the metadevice is striped evenly across a set of physical drives. Because each properly configured thin data pool ensures that thin devices will be widely striped across the back end of the Symmetrix in 768 KB extents (12 Symmetrix tracks), striping can be implemented without the requirement of metas. In other words, wide striping at the device level, for a single thin device, requires no planning on the part of the administrator.

Thin metadevices

If the use of thin metadevices is desired in order to present larger devices to hosts, metadevices can be used. Concatenated metadevices using regular volumes have the drawback of being limited by the performance of a single metamember. In the case of a concatenated meta comprised of thin devices, each member device is typically spread across the entire underlying thin pool, thus eliminating that drawback.

Concatenated metadevices have two important operational advantages over striped metas:

- Concatenated metadevices can be expanded without destroying
Virtual Provisioning

existing data by adding meta members to an existing metadevice.

- Non-metadevices can be converted to concatenated metadevices without destroying existing data.

In most cases, EMC recommends using concatenated rather than striped metadevices with Virtual Provisioning. However, there may be certain situations where better performance may be achieved using striped metas.

With Synchronous SRDF®, Enginuity allows one outstanding write per thin device per path. With concatenated metadevices, this could cause a performance problem by limiting the concurrency of writes. This limit will not affect striped metadevices in the same way because of the small size of the metavolume stripe (1 cylinder or 1920 blocks).

Enginuity allows eight read requests per path per thin device. This limits the number of read requests that can be passed through to the thin pool regardless of the number of data devices that may be in it. This can cause slower performance in environments with a high read miss rate.

Symmetrix Enginuity has a logical volume write pending limit to prevent one volume from monopolizing writeable cache. Because each meta member gets a small percentage of cache, a striped meta is likely to offer more writable cache to the meta volume.

Before configuring striped metadevices, please consult with an EMC performance specialist.

**Caution: Striped thin metadevices cannot be expanded while preserving data.**

**Oversubscribed thin pools**

As already noted, it is permissible for the amount of storage mapped to a thin device to be less than the reported size of the device. It is also permissible for the sum of the reported sizes of the thin devices using a given thin pool to exceed the total capacity of the data devices comprising the thin pool. In this case the thin pool is said to be “oversubscribed.” Oversubscribing allows the organization to present larger than needed devices to hosts and applications without having to purchase enough physical drives to fully allocate all of the space represented by the thin devices.

The capacity utilization of oversubscribed pools must be monitored to determine when space must be added to the thin pool to avoid out-of-space conditions.
Not all application environments will be appropriate for use with oversubscribed thin pools. If the application or any part of the software stack underlying the application, such as a database, file system, and logical volume manager, has a tendency to produce dense patterns of writes to all available storage, thin devices will tend to become fully allocated quickly. If thin devices belonging to an oversubscribed pool are used in this type of environment out-of-space conditions may be encountered before an administrator can take steps to add storage capacity to the thin data pool.

Caution: Operating systems react in different ways when encountering out-of-space conditions due to a thin pool running out of space. The *EMC Host Connectivity Guide* for the specific host platform that will be accessing thin devices bound to an oversubscribed pool provides more details. Host connectivity guides for all supported platforms are available on the EMC Powerlink® website.

**Thin-hostile environments**

The use of oversubscribed thin device configurations is not viable in all application environments unless the host administrator takes measures to restrict how much of the storage seen by the host is actually made available to the applications. This is because some applications quickly generate sufficiently dense patterns of writes to force large amounts of storage presented to the application to become allocated. Such application environments are called “thin-hostile.”

There are a variety of factors that can contribute to making a given application environment thin-hostile:

- One step or a combination of steps involved in simply preparing storage for use by the application may force all of the storage that is being presented to become fully allocated.

- If the storage space management policies of the application and underlying software components do not tend to reuse storage that was previously used and released, that will increase the speed in which underlying thin devices become fully allocated.

- Whether there are any data copy operations (including drive balancing operations and de-fragmentation operations) that are carried out as part of the administration of the environment.

- If there are administrative operations such as bad block detection operations or file system check commands that perform dense patterns of writes on all reported storage.

If an oversubscribed thin device configuration is used with a thin-hostile
application environment (without a type of host-level restriction of the sort that will be described) then the likely result is that the capacity of the thin pool will become exhausted before the storage administrator can add capacity unless measures are taken at the host level to restrict the amount of capacity that is actually placed in control of the application.

Note: Thin device configurations that are not oversubscribed cannot be thin-hostile.

Pre-provisioning with thin devices in a thin hostile environment

In some cases, many of the benefits of pre-provisioning with thin devices can be exploited in a thin-hostile environment. This requires that the host administrator cooperate with the storage administrator by enforcing restrictions on how much storage is placed under the control of the thin-hostile application. For example:

The storage administrator pre-provisions larger than initially needed thin devices to the hosts, but only configures the thin pools with the storage needed initially. The various steps required to create, map, and mask the devices and make the target host operating systems recognize the devices are performed.

The host administrator uses a host logical volume manager to carve out portions of the devices into logical volumes to be used by the thin-hostile applications.

The host administrator may want to fully preallocate the thin devices underlying these logical volumes before handing them off to the thin-hostile application so that any storage capacity shortfall will be discovered as quickly as possible, and so that discovery is not made by way of a failed host write.

When more storage needs to be made available to the application, the host administrator extends the logical volumes out of the thin devices that have already been presented. Many databases can absorb an additional disk partition non-disruptively, as can most file systems and logical volume managers.

Again the host administrator may want to fully allocate the thin devices underlying these volumes before assigning them to the thin-hostile application.

In this example it is still necessary for the storage administrator to closely monitor the oversubscribed pools. This procedure will not work if the host administrators do not observe restrictions on how much of the storage presented is actually assigned to the application.
Data availability considerations of wide striping

The data availability considerations that apply to a thin device configuration are the same as those that apply to a configuration in which device-level wide striping is achieved using regular devices (such as with striped metadevices). Regardless of which mechanism is used to achieve device-level wide striping, the use of wide striping does not cause the amount of data that may be lost in a data loss event to increase. Device-level wide striping may, however, increase the number of LUNs that will be affected by a data loss event. When designing a configuration involving thin devices, the availability properties of the thin devices can be calculated by considering the following aspects of the drives underlying the thin pool:

- Mean time between drive failure (MTBF) for drives underlying the set of data devices
- The type of RAID protection used
- The number of RAID groups over which the set of data devices are spread
- Mean time to repair (MTTR), including drive rebuild time and service call response time

It is important to note that availability is dependent upon the number of RAID groups underlying the thin pool. Because of the use of wide striping, the availability of any thin device using a pool may be impacted by the failure of any RAID group used by the thin pool to which the thin device is bound. The dependency on the MTTR should also be noted. It is recommended that the drives underlying a thin pool are configured with available permanent spares.

When designing a configuration involving thin devices (or any other approach that results in devices that are widely striped), device level availability implications should be carefully considered.

Virtual Provisioning configuration guidelines

Creating a Virtual Provisioning environment that results in high performance and availability requires adherence to specific configuration standards.

Overview of configuration requirements for thin devices

When planning a configuration using thin devices the first step involves determining how many separate thin pools are needed and the required composition of each thin data pool. Typically, this will involve conceptually organizing drive storage into separate classes, with further
subdivision as needed to allow the back-end resources (drives and DAs) used by the pools to be isolated from one another. Depending on the mix of applications to be placed on thin devices, it will often be necessary to create multiple thin pools but generally, the most efficient use of resources will be achieved by using a minimal number of pools.

Typically, a thin pool should be designed for use by a given application, or set of related applications, aligned with a given business group. The applications sharing a thin pool will compete for back-end resources, including thin pool storage capacity, so applications should not share the same pool if this is not acceptable. The devices comprising a thin pool will have the same performance and protection properties, so the applications sharing a thin pool should have the same performance and protection requirements.

Once a set of thin pools has been designed, the back-end layout of future storage provisioning requests may be planned by simply considering the class of storage that is required and which business group is requesting the storage, if that class has been further sub-divided.

**Sizing resources for thin devices**

The initial storage capacity of a thin pool must be large enough to accommodate the initial storage requirements of the applications using the pools. The initial storage requirements must be well understood and there must always be enough physical space available to the thin devices to accommodate this initial requirement.

If an oversubscribed pool is being used to permit inexpensive pre-provisioning then there should be enough additional capacity to ensure that there is time for the storage administrator to add capacity to the pool when it is determined that the risk of the pool filling has become imminent.

The data devices comprising a thin pool should be spread across a back-end hardware configuration that has enough available performance capacity to handle the I/O workload that will delivered, both in the near and longer term, to regions of thin devices that get mapped to storage allocated from the initial pool. After determining the DA and drive configuration of the initial thin pool layout, adding storage capacity to the pool at a later time can be done straightforwardly. If, however, the initial pool was not spread over enough drives and DAs to accommodate the workload on the initially mapped regions of thin devices, adding data devices to the pool, even if they are on separate physical drives will not automatically alleviate the problem.
Additional cache requirements

Because thin devices are cache devices, the use of thin devices introduces some additional cache requirements. Though minimal, these requirements should be considered when designing a thin device configuration. Each thin device has a cache overhead of approximately 143 KB plus 8 KB for each gigabyte of reported device size.

Thin pool layout considerations

Many of the same considerations that apply to the design of other types of Symmetrix pools also apply to the design of thin pools. The devices comprising a given thin pool should satisfy all of the following requirements:

- Only data devices may be placed in a thin pool.
- The data devices must all have the same emulation.
- The data devices must all have the same protection type.
- It is recommended that data devices in a pool all reside on drives that have the same rotational speed.
- The data devices in the pool should generally be spread across as many DAs and drives of a given speed as possible.
- The devices should be evenly spread across the DAs and drives. The wide striping provided by Virtual Provisioning will spread thin devices evenly across the data devices. The storage administrator must ensure that the data devices are evenly spread across the back end.
- It is recommended that all data devices in a pool are of the same size. Using different size devices could result in uneven data distribution.
- The data device sizes should be as large as possible to minimize the number of devices required to encompass the desired overall pool capacity.

Though all of these requirements should be met by a given pool, a Symmetrix array is likely to contain more than one pool. The configuration of pools within a given system may differ in a variety of ways, including the emulation, protection type, and rotational speed of the physical drives comprising each of the various pools.

Considerations with allocation

Even though space can be allocated for any region of a disk, it may not be advisable to do so. File systems as well as databases and other
applications typically choose the area of the drive on which to write. Usually, this allocation policy is not known to end users. Hence, a partial allocation by an end user may result in sub-optimal allocation.

Because all thin devices bound to a pool consume space from the pool, a “badly behaved” device could consume a large amount of space from the pool, hence penalizing other “well behaved” devices. The “allocate” facility should be used when a storage administrator believes that a specific thin device is critical and should be fully allocated so it never runs out of space. In such instances, the storage administrator can allocate the entire device, thus guaranteeing that other “badly behaved” devices can never starve a critical device.

**Adding capacity to a thin pool that is running out of space**

When an oversubscribed thin pool begins to run out of space, data devices should be added to the pool before the pool completely fills. Otherwise there is a risk that an application that performs a write to a previously unmapped portion of a thin device may encounter an out-of-space error. Data devices can be added to a thin pool non-disruptively. The set of data devices to be added to an existing thin pool should satisfy the following requirements:

- They must have the same protection type as the devices already in the target thin pool.
- They must have the same emulation as the devices already in the target thin pool.
- They should all reside on drives that have the same rotational speed.
- When considered on their own, the data devices should be evenly spread across a set of DAs and drives.
- It is recommended that all data devices in a pool are of the same size. Using different size devices could result in uneven data distribution.
- The sizes of the data devices to be added should be as large as possible to minimize the number of devices required to encompass the desired added pool capacity.

Typically, the DA and drive configuration underlying a set of data devices to be added to a pool should be sized to have enough available performance capacity to handle the workload that will be delivered to any regions of thin devices that get allocated after the pool addition. This is because once the pool addition is made and any remaining capacity in the pre-existing pool becomes completely exhausted, a point
will be reached where all storage allocations made from the pool will be satisfied using just data devices from the pool addition. In most cases it is inadvisable to add a set of data devices that is spread over a small number of drives to a thin pool that is running out of space.

**Automated pool rebalancing**

Beginning in Enginuity 5874, SMC 7.1, and Solutions Enabler 7.1, automated pool rebalancing allows the user to automatically rebalance workloads non-disruptively in order to extend thin pool capacity in small increments as needed, maximizing performance and minimizing TCO. Users can run a command against a thin pool that will rebalance the allocated extents across all enabled data devices in the pool. This allows a small number of data devices to be used to expand a pool without causing wide striping to be compromised.

Up to eight thin pools may be rebalanced at any one time.

**Ensuring that DATA devices are spread wide across the back end**

In order to ensure good initial performance and to reduce the possibility that adding data devices to a pool will negatively impact current performance, data devices put in a pool should be spread across as many DAs and drives as possible. In many cases, contiguously numbered Symmetrix devices will be spread across DA processors, but this must be verified for each specific configuration. A good rule of thumb is to create a pool with at least as many data devices as there are DA slices or processors (four per DA board).

For example, a DMX-4 with four disk directors contains 16 DA processors (four per disk director) with each processor having two Fibre Channel drive loops. That means that a pool should be created with a minimum of 16 devices and expanded in multiples of 16. Using default Symmetrix configuration rules, a group of 16 contiguously numbered, RAID 1 (mirrored) data volumes will have each mirror on a separate DA slice and loop. This means that if the first mirror of device 117E is on DA 01B-1, the next device that will have one of its mirrors on DA 01B-1 will be 118E. The same holds true for the second mirror. This means that adding volumes 117E through 118D (16 devices) to a thin pool will ensure that the devices are spread across all available DA slices. This example should be considered a very basic guideline. Because all applications and environments are different and have different performance requirements, consulting with an EMC performance specialist when designing a virtual provisioning environment is highly recommended.

An exception to this general rule can occur when Symmetrix devices
have been deleted and new devices are created that fill in the “hole” in
device numbers left by the deleted devices. Regular devices that have
been swapped by Symmetrix Optimizer or non-optimal upgrades to the
Symmetrix may also introduce exceptions. Because of this, the physical
location of each data device should always be verified before adding it to
a thin pool.

Migrating to thin devices

Once it has been decided to use thin devices there are choices that may
need to be made about how to accomplish this. In the case of the
deployment of a new application or host the decision is simple; all new
devices presented can be thin.

Transitioning an existing application or host to thin devices is more
difficult if one of the purposes of using thin devices is to use an
oversubscribed thin device configuration.

Currently, there is no array-based copy mechanism that can be used to
copy data from a regular device to a thin device that will recognize
unused space on the source device and mark it as unallocated on the
target thin device. The result is that the amount of storage that will be
allocated for the thin device is equal to the full size of the source device.

There is also no mechanism for recognizing and reclaiming unused space
on a thin device. Once space on a thin device is allocated there is no
mechanism for gathering information from the host or application about
which blocks are actually unused and therefore are eligible to be released
so that the underlying storage may be reclaimed.

If one of the purposes of using thin devices is to exploit the benefits of an
oversubscribed configuration with an existing application, it is possible
to enter into an oversubscribed configuration:

- Data can be migrated from existing devices to larger,
  oversubscribed thin devices using either LVM or array-based
  migration methods. The data that is migrated will be fully
  allocated on the target thin devices, but post-migration host
  writes to the thin devices can take advantage of
  oversubscription. If concatenated thin metadevices are used, the
  size of the thin target devices may be subsequently increased to
  introduce greater degrees of oversubscription.

- A host-level copy mechanism can be used to copy data from
  regular source volumes to thin target volumes. This approach
  has the disadvantage of consuming host cycles and requiring
  that the data traverse the SAN even if the source and target are
Virtual Provisioning

in the same array. It may, however, have the advantage of reclaiming (or failing to claim) unused space in the process.

- In Microsoft Windows environments, EMC Open Migrator/LM, which is a host-based migration product, can be used to migrate a regular device to a thin device using the Sparse Copy feature. This will copy only the used space from the regular device to the target thin device.

- In supported UNIX environments, EMC PowerPath® Migration Enabler (PPME) will allow migrations between regular and thin devices while allowing the thin target device to remain thin. This can be accomplished using either PPME Host Copy or PPME with EMC TimeFinder®/Clone to copy only chunks that contain non-zero data. The E-Lab™ Navigator on the EMC Powerlink website provides information on how to confirm support for a specific operating system.

Dynamic Cache Partitioning

Symmetrix Dynamic Cache Partitioning can be used with thin devices. Thin devices may belong to the default cache partition or they can be assigned to a named cache partition. Logical volume and system write pending limits will be enforced for thin devices just as they would for regular Symmetrix devices.

There is no requirement that all thin devices using a given thin pool be assigned to the same cache partition. If a given thin pool is being shared by multiple applications then it may make sense to divide the thin devices using the pool among multiple cache partitions.

Data devices must remain in the default cache partition.

Local and remote replication between thin devices

Organizations will be able to perform “thin to thin” replication with Symmetrix thin devices by using standard TimeFinder, SRDF, and Open Replicator operations. This includes TimeFinder/Snap, TimeFinder/Clone, SRDF/Synchronous, SRDF/Asynchronous, and SRDF/Data Mobility.

In addition, 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, zeroes will be sent for any regions of the thin device that have not been allocated, or that have been allocated but have not been 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, then a portion of the target thin device, equal in size to the reported size of the source volume, will become allocated.

Notes: Data devices may not be used as the source or target of any type of TimeFinder operation.

Multi-virtual snap is now supported with thin devices and TimeFinder/Snap. This means that as many as 128 snapshots can be created from a single thin volume.

The target of a TimeFinder/Snap operation is always a Virtual Device (VDEV). Though not technically a thin device, VDEVs are cache only devices that use pooled storage in a similar manner.

Local replication between thin and regular devices

With the Enginuity 5874 Q4 Service Release, TimeFinder/Clone enables users to 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, thin device tracks 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. This is accomplished using Virtual Provisioning space reclamation.

Local replication between thin and larger thin devices

TimeFinder/Clone now supports the creation of clone sessions between thin source devices and larger thin target devices in Enginuity 5874.

Note: The size in this case refers to the amount of host-perceived capacity rather than allocated capacity for the thin devices. Allocated capacity following the clone copy will be the same for both the source and target devices.

Virtual Provisioning space reclamation

Solutions Enabler 7.1 and Enginuity 5874 provide the ability to automatically reclaim extents containing all zeros after migrating from standard volumes to thin volumes in order to reduce capacity requirements and TCO.

This feature involves the de-allocation of thin device extents that contain all zeros and is an extension of the existing Virtual Provisioning space.
de-allocation mechanism. Running the space reclamation command will spawn a DA background task that will examine the allocated extents on specified thin device. For each allocated extent, all 12 tracks will be brought into cache and scanned 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.

Space reclamation is not supported on actively replicating SRDF volumes. The SRDF link must be suspended prior to running the reclamation operation.

Reclamation will also not be performed on individual tracks that are participating in a local replication session.

**Local replication with SRDF/A devices**

Support for SRDF/A R1 and R2 devices as clone source devices is available in Enginuity 5874 for Symmetrix VMAX arrays.

In Enginuity 5773, TimeFinder/Snap and TimeFinder/Clone operations are not supported with SRDF devices that are actively replicating in Asynchronous mode. However, the ability to create gold copies of R2 devices is critical to data protection during resynchronization operations. This can still be accomplished despite the current restriction.

When the RDF link is partitioned and the R2 cannot communicate across the link, the mode of operation for the R2 will change from asynchronous to synchronous because synchronous is the SRDF default and the mode of operation is an R1 attribute. Because of this, there is no additional action required before creating local clone or snap copies on the R2 side. At the source, the R1s will remain asynchronous until the mode is changed to synchronous manually (with a force due to the partitioned state). The user would then be able to make a local copy of the R1 devices. If the link has recovered and is in a suspended state, the mode will have to be changed to synchronous first before any snap or clone sessions can be created.

If a device is in a Trans Idle state, Trans Idle must be deactivated before a clone or snap session can be created.

**Notes:** Leaving the devices in synchronous mode while resynchronizing the SRDF pairs is not recommended. In general, the resynchronization should be started in Adaptive Copy mode until there are less than 30,000 tracks remaining to be copied to the R2. At that time, the mode can be set to asynchronous.

With the Enginuity 5874 Q4 SR, TimeFinder/Clone (with a thick or thin clone...
Data device protection

In Enginuity 5773, RAID 1 (mirrored), RAID 5 (3+1), and RAID 6 (6+2 and 14+2) are supported for the protection of data devices. Support for RAID 5 (7+1) has been added in Enginuity 5874.

Because Virtual Provisioning stripes data across many physical drives (and RAID groups) in a thin pool, each RAID group may contain data from many thin devices. In some cases, a particular physical drive might have chunks of data from all the thin devices bound to a particular pool, increasing the need for greater protection against drive failures. A RAID 1 or RAID 6 thin pool provides better protection than RAID 5 would against the rare case of multiple drive failures within a RAID group.

Note: In Enginuity 5773, thin devices bound to a pool comprised of RAID 5 data devices cannot be replicated remotely with SRDF and cannot be source devices for TimeFinder/Snap. They may be replicated locally using TimeFinder/Clone. In Enginuity 5874, these restrictions have been removed.

Data and thin device restrictions

The following are general restrictions that apply to data devices:

- Data devices cannot be swapped using Symmetrix Optimizer. Thin devices, like virtual devices for TimeFinder/Snap, are cache-only devices, so Optimizer swapping is not applicable.
- There is no support for dynamic sparing of data devices. The use of dynamic sparing will be supported for the drives underlying data devices. Because dynamic sparing is done on the logical level, splits on a failing drive that contain data devices will be copied to a dynamic spare, but the data device splits themselves will not be copied. To protect data devices, permanent member sparing must be used.
- Data devices cannot be metadevices.
- Regular devices cannot be converted to thin devices and vice versa.

Thin pool restrictions

The number of pools that can be configured in a Symmetrix array is 512. This is the total number of pools, including Virtual Provisioning thin pools, SRDF/A Delta Set Extension (DSE) pools, or TimeFinder/Snap pools.
Thin and data device limits within a pool

There is no limit to the number of thin devices that can be bound to a thin pool or data devices that can be added to a thin pool.

The limit to the number of thin and data devices that can be configured within a Symmetrix system is 64,000.

Virtual Provisioning with Solutions Enabler

The following are examples of how to configure, update, and monitor a Symmetrix Virtual Provisioning environment using Solutions Enabler (SYMCLI).

The examples used in this technical note have been created for illustrative purposes and do not represent a Virtual Provisioning environment configured for production workloads.

Note: The `symconfigure` command is used to configure and modify thin devices, data devices, and pools. In all of the command examples, the `-cmd` option is used followed by the command syntax in quotes. As with all `symconfigure` commands requiring configuration input, the `-f` option can be specified followed by a path to a command file containing the syntax shown in the examples.

Configuring and viewing data devices and pools

Data devices are created in the same way that all Symmetrix devices are created. In this example, the `symconfigure` command is used to create mirrored devices with the `datadev` attribute. Devices with other protection schemes may also be supported for use in thin pools, depending on the specific release of Enginuity in use. Regardless of their protection, all devices with the `datadev` attribute are used exclusively for populating thin pools:

```
# symconfigure -sid 34 -cmd "create dev count=16, config=2-Way-Mir, attribute=datadev, emulation=FBA, size=4602;" commit -v -nop
A Configuration Change operation is in progress. Please wait...
```

```
Establishing a configuration change session.............Established.
Processing symmetrix 000190103232
{
  create dev count=16, size=4602, emulation=FBA,
    config=2-Way Mir, mvs_ssid=0000, attribute=datadev;
}
Performing Access checks.................................Allowed.
Checking Device Reservations............................Allowed.
Submitting configuration changes.......................Submitted
```
Virtual Provisioning

Validating configuration changes.................Validated.

New symdevs: 10E4:10F3
Initiating PREPARE of configuration changes.........Prepared.
Initiating COMMIT of configuration changes...........Queued.
COMMIT requesting required resources.................Obtained.
Step 004 of 078 steps................................Executing.
Step 015 of 078 steps................................Executing.
Step 020 of 078 steps................................Executing.
Step 027 of 078 steps................................Executing.
Step 029 of 078 steps................................Executing.
Step 040 of 078 steps................................Executing.
Step 056 of 151 steps................................Executing.
Step 067 of 151 steps................................Executing.
Step 071 of 151 steps................................Executing.
Step 071 of 151 steps................................Executing.
Step 071 of 151 steps................................Executing.
Step 071 of 151 steps................................Executing.
Step 082 of 173 steps................................Executing.
Step 090 of 173 steps................................Executing.
Step 097 of 173 steps................................Executing.
Step 098 of 173 steps................................Executing.
Step 098 of 173 steps................................Executing.
Step 099 of 173 steps................................Executing.
Step 104 of 173 steps................................Executing.
Step 104 of 173 steps................................Executing.
Step 107 of 173 steps................................Executing.
Step 110 of 173 steps................................Executing.
Step 117 of 173 steps................................Executing.
Step 123 of 173 steps................................Executing.
Step 125 of 173 steps................................Executing.
Step 125 of 173 steps................................Executing.
Step 125 of 173 steps................................Executing.
Step 130 of 173 steps................................Executing.
Local: COMMIT........................................Done.
Terminating the configuration change session........Done.
The configuration change session has successfully completed.

# symdev list -sid 34 -datadev

Symmetrix ID: 000190103334

<table>
<thead>
<tr>
<th>Device Name</th>
<th>Directors</th>
<th>Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sym</td>
<td>Physical</td>
<td>SA :P</td>
</tr>
<tr>
<td>10E4</td>
<td>Not Visible</td>
<td>????:?</td>
</tr>
<tr>
<td>10E5</td>
<td>Not Visible</td>
<td>????:?</td>
</tr>
<tr>
<td>10E6</td>
<td>Not Visible</td>
<td>????:?</td>
</tr>
<tr>
<td>10E7</td>
<td>Not Visible</td>
<td>????:?</td>
</tr>
<tr>
<td>10E8</td>
<td>Not Visible</td>
<td>????:?</td>
</tr>
<tr>
<td>10E9</td>
<td>Not Visible</td>
<td>????:?</td>
</tr>
<tr>
<td>10EA</td>
<td>Not Visible</td>
<td>????:?</td>
</tr>
<tr>
<td>10EB</td>
<td>Not Visible</td>
<td>????:?</td>
</tr>
</tbody>
</table>
Thin pools are also created using the `symconfigure` command. They can be created without adding data devices so that they can be populated at a later time. In this example, a pool called “HR” is created:

```
# symconfigure -sid 34 -cmd "create pool HR type=thin;" commit -nop
```

After a pool is created, data devices can be added to the pool and enabled:

```
# symconfigure -sid 34 -cmd "add dev 10E4:10E5 to pool HR type=thin, member_state=ENABLE;" commit -nop
```

Devices can exist in a thin pool and be in a disabled state, but disabled devices are not available for extent allocation.

A composite command can also be used that will allow the user to create data devices, add them to an existing pool, and enable them in a single step:

```
# symconfigure -sid 32 -cmd "create dev count=2, config=2-Way-Mir, attribute=datadev, emulation=FBA, size=4602, in pool HR, member_state=ENABLE;" commit -nop
```

The `symcfg list` command is used to display a list of Symmetrix pools. These pools can be Virtual Provisioning thin pools, SRDF/A DSE pools, or TimeFinder/Snap pools. The output can be limited to Virtual Provisioning thin pools by including the `-thin` option:

```
# symcfg list -pools -thin -sid 34
```

Symmetrix ID: 000190103334

<table>
<thead>
<tr>
<th>SYMMETRIX POOLS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Pool Name</strong></td>
</tr>
<tr>
<td>------------------</td>
</tr>
<tr>
<td>HR</td>
</tr>
</tbody>
</table>

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

Total                              -------- -------- -------- -------- ----
Tracks                               138048   138048        0   138048    0

Legend for Pool Types:

SN = Snap,  RS = Rdfa DSE  TH = Thin Data

Data can also be displayed using tracks (default), megabytes (-mb), or gigabytes (-gb). In this example, the same output from the previous command is shown with the output in megabytes:

# symcfg list -pool -thin -sid 34 -mb

Symmetrix ID: 000190103334

SYM MET R I X     PO O LS

<table>
<thead>
<tr>
<th>Pool Name</th>
<th>Type</th>
<th>Dev Emulation</th>
<th>Dev Configuration</th>
<th>Total MBs</th>
<th>Enabled MBs</th>
<th>Used MBs</th>
<th>Free MBs</th>
<th>Full (%)</th>
<th>Sta</th>
</tr>
</thead>
<tbody>
<tr>
<td>HR</td>
<td>TH</td>
<td>FBA</td>
<td>2-Way Mir</td>
<td>8628</td>
<td>8628</td>
<td>0</td>
<td>8628</td>
<td>0 Ena</td>
<td></td>
</tr>
</tbody>
</table>

Total                              -------- -------- -------- -------- ----
MBs                                    8628     8628        0     8628    0

Legend for Pool Types:

SN = Snap,  RS = Rdfa DSE  TH = Thin Data

Information about specific pools can be viewed using the symcfg show command. In this example, the newly created pool is displayed along with details about the pool and the data devices that have been added to it. The -all option shows both enabled and disabled devices. In this case, all devices are enabled. Using the -detail option displays any bound thin devices:

# symcfg show -pool HR -thin -detail -all -sid 34

Symmetrix ID: 000190103334

Symmetrix ID: 000190103334
Pool Name : HR
Pool Type : Thin
Dev Emulation : FBA
Dev Configuration : 2-Way Mir
Pool State : Enabled
# of Devices in Pool : 2
# of Enabled Devices in Pool : 2
Max. Subscription Percent : None

Enabled Devices(2):

{
Virtual Provisioning

---

Sym     Total Alloc Free   Full Device
Dev     Tracks Tracks Tracks (%) State
---            -------- -------- -------- ------- ----
10E4     69024     0       69024     0     Enabled
10E5     69024     0       69024     0     Enabled

Tracks  138048     0      138048     0
}

No Thin Devices Bound to Device Pool HR

The symcfg list command can also be used to list information about devices. In this example, the -datadev option is used to restrict the output to data devices:

```
# symcfg -sid 34 list -datadev

Symmetrix ID: 000190103334

SYM MTRIX DATA DEVICES

<table>
<thead>
<tr>
<th>Sym</th>
<th>Emul</th>
<th>Config</th>
<th>Type</th>
<th>Pool Name</th>
<th>State</th>
<th>Total Tracks</th>
<th>Used Tracks</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>10E4</td>
<td>FBA</td>
<td>2-Way Mir</td>
<td>TH</td>
<td>HR</td>
<td>Ena</td>
<td>69024</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10E5</td>
<td>FBA</td>
<td>2-Way Mir</td>
<td>TH</td>
<td>HR</td>
<td>Ena</td>
<td>69024</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10E6</td>
<td>FBA</td>
<td>2-Way Mir</td>
<td>-</td>
<td>-</td>
<td>Dis</td>
<td>69024</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10E7</td>
<td>FBA</td>
<td>2-Way Mir</td>
<td>-</td>
<td>-</td>
<td>Dis</td>
<td>69024</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10E8</td>
<td>FBA</td>
<td>2-Way Mir</td>
<td>-</td>
<td>-</td>
<td>Dis</td>
<td>69024</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10E9</td>
<td>FBA</td>
<td>2-Way Mir</td>
<td>-</td>
<td>-</td>
<td>Dis</td>
<td>69024</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10EA</td>
<td>FBA</td>
<td>2-Way Mir</td>
<td>-</td>
<td>-</td>
<td>Dis</td>
<td>69024</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10EB</td>
<td>FBA</td>
<td>2-Way Mir</td>
<td>-</td>
<td>-</td>
<td>Dis</td>
<td>69024</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10EC</td>
<td>FBA</td>
<td>2-Way Mir</td>
<td>-</td>
<td>-</td>
<td>Dis</td>
<td>69024</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10ED</td>
<td>FBA</td>
<td>2-Way Mir</td>
<td>-</td>
<td>-</td>
<td>Dis</td>
<td>69024</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10EE</td>
<td>FBA</td>
<td>2-Way Mir</td>
<td>-</td>
<td>-</td>
<td>Dis</td>
<td>69024</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10EF</td>
<td>FBA</td>
<td>2-Way Mir</td>
<td>-</td>
<td>-</td>
<td>Dis</td>
<td>69024</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10F0</td>
<td>FBA</td>
<td>2-Way Mir</td>
<td>-</td>
<td>-</td>
<td>Dis</td>
<td>69024</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10F1</td>
<td>FBA</td>
<td>2-Way Mir</td>
<td>-</td>
<td>-</td>
<td>Dis</td>
<td>69024</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10F2</td>
<td>FBA</td>
<td>2-Way Mir</td>
<td>-</td>
<td>-</td>
<td>Dis</td>
<td>69024</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10F3</td>
<td>FBA</td>
<td>2-Way Mir</td>
<td>-</td>
<td>-</td>
<td>Dis</td>
<td>69024</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1104</td>
<td>FBA</td>
<td>2-Way Mir</td>
<td>-</td>
<td>-</td>
<td>Dis</td>
<td>69045</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1105</td>
<td>FBA</td>
<td>2-Way Mir</td>
<td>-</td>
<td>-</td>
<td>Dis</td>
<td>69045</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1106</td>
<td>FBA</td>
<td>2-Way Mir</td>
<td>-</td>
<td>-</td>
<td>Dis</td>
<td>69045</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1107</td>
<td>FBA</td>
<td>2-Way Mir</td>
<td>-</td>
<td>-</td>
<td>Dis</td>
<td>69045</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1108</td>
<td>FBA</td>
<td>2-Way Mir</td>
<td>-</td>
<td>-</td>
<td>Dis</td>
<td>69045</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1109</td>
<td>FBA</td>
<td>2-Way Mir</td>
<td>-</td>
<td>-</td>
<td>Dis</td>
<td>69045</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>110A</td>
<td>FBA</td>
<td>2-Way Mir</td>
<td>-</td>
<td>-</td>
<td>Dis</td>
<td>69045</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>110B</td>
<td>FBA</td>
<td>2-Way Mir</td>
<td>-</td>
<td>-</td>
<td>Dis</td>
<td>69045</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>110C</td>
<td>FBA</td>
<td>2-Way Mir</td>
<td>-</td>
<td>-</td>
<td>Dis</td>
<td>69045</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>110D</td>
<td>FBA</td>
<td>2-Way Mir</td>
<td>-</td>
<td>-</td>
<td>Dis</td>
<td>69045</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>110E</td>
<td>FBA</td>
<td>2-Way Mir</td>
<td>-</td>
<td>-</td>
<td>Dis</td>
<td>69045</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>110F</td>
<td>FBA</td>
<td>2-Way Mir</td>
<td>-</td>
<td>-</td>
<td>Dis</td>
<td>69045</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Total Tracks  1932924  0  0
Legend for Pool Types:

SN = Snap, RS = Rdfa DSE, TH = Thin Data, - = N/A

Configuring, binding, and viewing thin devices

Thin devices are also created using the symconfigure command. The configuration type must be specified as TDEV:

```
# symconfigure -sid 34 -cmd "create dev count=16, size=4602, emulation=fba, config=TDEV;" commit -nop
```

Thin devices can also be created and bound to an existing pool in a single operation by adding "binding to pool = <poolname>" to the symconfigure command.

Once the thin devices are created, they can then be bound to a thin pool:

```
# symconfigure -sid 34 -cmd "bind tdev 10F4:10F7 to pool HR;" commit -nop
```

Because thin devices can be created with no associated physical storage, thin devices that are not bound to a thin pool will remain Not Ready. The newly created devices can be displayed using symdev list. Adding the -tdev option will restrict the output to only thin devices.

In the following output, the four thin devices bound to the HR pool are Read/Write while the other 12 devices remain Not Ready:

```
# symdev list -sid 34 -tdev
Symmetrix ID: 000190103334

<table>
<thead>
<tr>
<th>Device Name</th>
<th>Directors</th>
<th>Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sym</td>
<td>Physical</td>
<td>SA :P DA :IT</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
<td>--------------</td>
</tr>
<tr>
<td>10F4 Not Visible</td>
<td>***<em>:</em></td>
<td>NA:NA</td>
</tr>
<tr>
<td>10F5 Not Visible</td>
<td>***<em>:</em></td>
<td>NA:NA</td>
</tr>
<tr>
<td>10F6 Not Visible</td>
<td>***<em>:</em></td>
<td>NA:NA</td>
</tr>
<tr>
<td>10F7 Not Visible</td>
<td>***<em>:</em></td>
<td>NA:NA</td>
</tr>
<tr>
<td>10F8 Not Visible</td>
<td>????:?</td>
<td>NA:NA</td>
</tr>
<tr>
<td>10F9 Not Visible</td>
<td>????:?</td>
<td>NA:NA</td>
</tr>
<tr>
<td>10FA Not Visible</td>
<td>????:?</td>
<td>NA:NA</td>
</tr>
<tr>
<td>10FB Not Visible</td>
<td>????:?</td>
<td>NA:NA</td>
</tr>
<tr>
<td>10FC Not Visible</td>
<td>????:?</td>
<td>NA:NA</td>
</tr>
<tr>
<td>10FD Not Visible</td>
<td>????:?</td>
<td>NA:NA</td>
</tr>
<tr>
<td>10FE Not Visible</td>
<td>????:?</td>
<td>NA:NA</td>
</tr>
<tr>
<td>10FF Not Visible</td>
<td>????:?</td>
<td>NA:NA</td>
</tr>
<tr>
<td>1100 Not Visible</td>
<td>????:?</td>
<td>NA:NA</td>
</tr>
<tr>
<td>1101 Not Visible</td>
<td>????:?</td>
<td>NA:NA</td>
</tr>
</tbody>
</table>
```
The `symcfg list` command, used here with the `-tdev` option, can also display information about thin devices. Note that binding thin devices to a pool will cause one extent from the thin pool to be allocated for each thin device. In this case, the data devices are mirrored devices (RAID 1), so the amount allocated is 12 tracks (768 KB):

```
# symcfg -sid 34 list -tdev
```

When the pool is examined using `symcfg show` with the `-detail` option,
Virtual Provisioning

the thin devices bound to the pool are displayed:

```bash
# symcfg show -pool HR -thin -detail -all -mb -sid 34
```

Symmetrix ID: 000190103334

Symmetrix ID : 000190103334
Pool Name : HR
Pool Type : Thin
Dev Emulation : FBA
Dev Configuration : 2-Way Mir
Pool State : Enabled
# of Devices in Pool : 2
# of Enabled Devices in Pool : 2
Max. Subscription Percent : None

Enabled Devices(2):
{
   Sym  Total  Alloc  Free  Full  Device
    Dev  MBs   MBs    MBs  (%)  State
        -------------------------------
    10E4  4314   2    4311  0   Enabled
    10E5  4314   0    4313  0   Enabled

        MBs  8628   3    8625  0
}

Thin Devices(4):
{
   Sym  Total  Subs  Alloc  Alloc  Wrtn  Wrtn
    Dev  MBs (%) MBs (%) MBs (%)
         -------------------------------
    10F4  4314  50    0    0    0    0
    10F5  4314  50    0    0    0    0
    10F6  4314  50    0    0    0    0
    10F7  4314  50    0    0    0    0

        MBs  17257  200    3    0    0    0
}

In the above output, the Subs (%) column indicates the subscription of the thin device as compared to the pool. It represents the percentage of pool space that this thin device would occupy if it was fully allocated. The total percentage of this column represents the amount of oversubscription against this pool. In this example, if all of the thin devices were fully allocated, they would need twice the amount of pool space than is currently available. This number gives the storage administrator an indication of how much extra space would be needed if all the thin devices bound to the pool were to be fully allocated.

Thin devices can be mapped and masked to hosts at any time after they are created. The host will be able to discover newly assigned thin devices even if they have not been bound to a pool and are in a Not Ready state.
The four thin devices created in the previous examples have been mapped to a host and are now available for use by the operating system.

The following output was gathered after a file system and a 250 MB file were written to each of the four thin devices:

```
# symcfg show -sid 34 -pool HR -detail -thin -all -mb

Symmetrix ID: 000190103334

Symmetrix ID                  : 000190103334
Pool Name                     : HR
Pool Type                     : Thin
Dev Emulation                 : FBA
Dev Configuration             : 2-Way Mir
Pool State                    : Enabled
# of Devices in Pool          : 2
# of Enabled Devices in Pool  : 2
Max. Subscription Percent     : None

Enabled Devices(2):
{
  --------------- --------------- --------------- ---------------
  Sym   Total   Alloc   Free  Full  Device
  Dev   MBs    MBs     MBs (%) State
  --------------- --------------- --------------- ---------------
  10B4     4314    775   3538   17  Enabled
  10B5     4314    769   3544   17  Enabled

  --------------- --------------- --------------- ---------------
  MBs          8628       1545       7083   17
}

Thin Devices(4):
{
  --------------- --------------- --------------- ---------------
  Sym   Total   Subs  Alloc  Alloc  Wrtn  Wrtn
  Dev   MBs    (%)   MBs (%) MBs (%)
  --------------- --------------- --------------- ---------------
  10F4     4314    50    386    4    324    4
  10F5     4314    50    386    4    324    4
  10F6     4314    50    386    4    324    4
  10F7     4314    50    386    4    324    4

  --------------- --------------- --------------- ---------------
  MBs           17257     200     1545    18    1297    15
}
```

**Adding additional data devices to a pool**

Data devices can be added to a pool when more space in the pool is required. Because thin device extents are taken from all enabled devices in a pool on which free space exists, care must be taken to ensure that performance is not compromised by adding too few data devices. This can happen because storage is allocated more heavily from the new data devices that have more free space, potentially creating disk contention. The Virtual Provisioning best practices documentation provides more information.
In this example, two more data devices are added to the HR thin pool:

```
# symconfigure -sid 34 -cmd "add dev 10E6:10E7 to pool HR type=thin;" commit -nop
```

In this case, the devices were added to the pool, but they were not enabled at the time (the `member_state=ENABLE` option was not included in the syntax):

```
# symcfg show -sid 34 -pool HR -detail -thin -all -mb
Symmetrix ID: 000190103334
Symmetrix ID                  : 000190103334
Pool Name                     : HR
Pool Type                     : Thin
Dev Emulation : FBA
Dev Configuration : 2-Way Mir
Pool State : Enabled
# of Devices in Pool          :   4
# of Enabled Devices in Pool  :   2
Max. Subscription Percent : None

Enabled Devices(2):
{
------------------------------------------------------
<p>| Sym         | Total      | Alloc       | Free Full  | Device |</p>
<table>
<thead>
<tr>
<th>Dev</th>
<th>MBs</th>
<th>MBs</th>
<th>MBs (%)</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>10E4</td>
<td>4314</td>
<td>775</td>
<td>3538   17</td>
<td>Enabled</td>
</tr>
<tr>
<td>10E5</td>
<td>4314</td>
<td>769</td>
<td>3544   17</td>
<td>Enabled</td>
</tr>
<tr>
<td>-------------</td>
<td>------------</td>
<td>------------</td>
<td>-----------</td>
<td>--------</td>
</tr>
<tr>
<td>MBs</td>
<td>8628</td>
<td>1545</td>
<td>7083   17</td>
<td></td>
</tr>
</tbody>
</table>
}

Disabled Devices(2):
{
------------------------------------------------------
<p>| Sym         | Total      | Alloc       | Free Full  | Device |</p>
<table>
<thead>
<tr>
<th>Dev</th>
<th>MBs</th>
<th>MBs</th>
<th>MBs (%)</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>10E6</td>
<td>4314</td>
<td>0</td>
<td>4314    0</td>
<td>Disabled</td>
</tr>
<tr>
<td>10E7</td>
<td>4314</td>
<td>0</td>
<td>4314    0</td>
<td>Disabled</td>
</tr>
<tr>
<td>-------------</td>
<td>------------</td>
<td>------------</td>
<td>-----------</td>
<td>--------</td>
</tr>
<tr>
<td>MBs</td>
<td>8628</td>
<td>0</td>
<td>8628    0</td>
<td></td>
</tr>
</tbody>
</table>
}

Thin Devices(4):
{
------------------------------------------------------
<p>| Sym         | Total | Subs | Alloc | Alloc | Wrttn | Wrttn |</p>
<table>
<thead>
<tr>
<th>Dev</th>
<th>MBs</th>
<th>(%)</th>
<th>MBs</th>
<th>(%)</th>
<th>MBs</th>
<th>(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10F4</td>
<td>4314</td>
<td>50</td>
<td>386</td>
<td>4</td>
<td>324</td>
<td>4</td>
</tr>
<tr>
<td>10F5</td>
<td>4314</td>
<td>50</td>
<td>386</td>
<td>4</td>
<td>324</td>
<td>4</td>
</tr>
<tr>
<td>10F6</td>
<td>4314</td>
<td>50</td>
<td>386</td>
<td>4</td>
<td>324</td>
<td>4</td>
</tr>
<tr>
<td>-------------</td>
<td>-------</td>
<td>------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
</tr>
</tbody>
</table>
Virtual Provisioning

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

They can be enabled with a separate operation:

```bash
# symconfigure -sid 34 -cmd "enable dev 10E6:10E7 in pool HR type=thin;" commit -nop
```

```bash
# symcfg show -sid 34 -pool HR -detail -thin -all -mb
```

Symmetrix ID: 000190103334

Symmetrix ID                  : 000190103334
Pool Name                     : HR
Pool Type                     : Thin
Dev Emulation                 : FBA
Dev Configuration             : 2-Way Mir
Pool State                    : Enabled
# of Devices in Pool          : 4
# of Enabled Devices in Pool  : 4
Max. Subscription Percent     : None

Enabled Devices(4):

<table>
<thead>
<tr>
<th>Sym</th>
<th>Total MBs</th>
<th>Alloc MBs</th>
<th>Free MBs</th>
<th>(%)</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>10F4</td>
<td>4314</td>
<td>775</td>
<td>3538</td>
<td>17</td>
<td>Enabled</td>
</tr>
<tr>
<td>10F5</td>
<td>4314</td>
<td>769</td>
<td>3544</td>
<td>17</td>
<td>Enabled</td>
</tr>
<tr>
<td>10F6</td>
<td>4314</td>
<td>0</td>
<td>4314</td>
<td>0</td>
<td>Enabled</td>
</tr>
<tr>
<td>10F7</td>
<td>4314</td>
<td>0</td>
<td>4314</td>
<td>0</td>
<td>Enabled</td>
</tr>
</tbody>
</table>

| MBs | 17256 | 1545 | 15711 | 8   |

Thin Devices(4):

<table>
<thead>
<tr>
<th>Sym</th>
<th>Total MBs</th>
<th>Subs (%)</th>
<th>Alloc MBs</th>
<th>Alloc (%)</th>
<th>Wrtn MBs</th>
<th>Wrtn (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10F4</td>
<td>4314</td>
<td>25</td>
<td>386</td>
<td>2</td>
<td>324</td>
<td>2</td>
</tr>
<tr>
<td>10F5</td>
<td>4314</td>
<td>25</td>
<td>386</td>
<td>2</td>
<td>324</td>
<td>2</td>
</tr>
<tr>
<td>10F6</td>
<td>4314</td>
<td>25</td>
<td>386</td>
<td>2</td>
<td>324</td>
<td>2</td>
</tr>
<tr>
<td>10F7</td>
<td>4314</td>
<td>25</td>
<td>386</td>
<td>2</td>
<td>324</td>
<td>2</td>
</tr>
</tbody>
</table>

| MBs | 17257 | 100 | 1545 | 9 | 1297 | 8 |

New writes to the thin devices will now be spread equally across all four enabled data devices:

```bash
# symcfg show -sid 34 -pool HR -detail -thin -all -mb
```

Symmetrix ID: 000190103334
### Virtual Provisioning

<table>
<thead>
<tr>
<th>Symmetrix ID</th>
<th>: 000190103334</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pool Name</td>
<td>: HR</td>
</tr>
<tr>
<td>Pool Type</td>
<td>: Thin</td>
</tr>
<tr>
<td>Dev Emulation</td>
<td>: FBA</td>
</tr>
<tr>
<td>Dev Configuration</td>
<td>: 2-Way Mir</td>
</tr>
<tr>
<td>Pool State</td>
<td>: Enabled</td>
</tr>
<tr>
<td># of Devices in Pool</td>
<td>: 4</td>
</tr>
<tr>
<td># of Enabled Devices in Pool</td>
<td>: 4</td>
</tr>
<tr>
<td>Max. Subscription Percent</td>
<td>: None</td>
</tr>
</tbody>
</table>

**Enabled Devices (4):**

<table>
<thead>
<tr>
<th>Sym</th>
<th>Total</th>
<th>Alloc</th>
<th>Free</th>
<th>Full</th>
<th>Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>10B4</td>
<td>4314</td>
<td>1023</td>
<td>3291</td>
<td>23</td>
<td>Enabled</td>
</tr>
<tr>
<td>10B5</td>
<td>4314</td>
<td>1020</td>
<td>3293</td>
<td>23</td>
<td>Enabled</td>
</tr>
<tr>
<td>10B6</td>
<td>4314</td>
<td>248</td>
<td>4065</td>
<td>5</td>
<td>Enabled</td>
</tr>
<tr>
<td>10B7</td>
<td>4314</td>
<td>249</td>
<td>4065</td>
<td>5</td>
<td>Enabled</td>
</tr>
</tbody>
</table>

**Thin Devices (4):**

<table>
<thead>
<tr>
<th>Sym</th>
<th>Total Subs</th>
<th>Alloc</th>
<th>Alloc</th>
<th>Wrtn</th>
<th>Wrtn</th>
</tr>
</thead>
<tbody>
<tr>
<td>10F4</td>
<td>4314</td>
<td>25</td>
<td>635</td>
<td>4</td>
<td>574</td>
</tr>
<tr>
<td>10F5</td>
<td>4314</td>
<td>25</td>
<td>635</td>
<td>4</td>
<td>574</td>
</tr>
<tr>
<td>10F6</td>
<td>4314</td>
<td>25</td>
<td>635</td>
<td>4</td>
<td>574</td>
</tr>
<tr>
<td>10F7</td>
<td>4314</td>
<td>25</td>
<td>635</td>
<td>4</td>
<td>574</td>
</tr>
</tbody>
</table>

**Binding additional thin devices to a pool**

Additional thin devices can also be bound to a pre-existing pool:

```
# symconfigure -sid 34 -cmd "bind tdev 10F8:10F9 to pool HR;" commit -nop

# symcfg show -sid 34 -pool HR -detail -thin -all -mb
```

<table>
<thead>
<tr>
<th>Symmetrix ID</th>
<th>: 000190103334</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pool Name</td>
<td>: HR</td>
</tr>
<tr>
<td>Pool Type</td>
<td>: Thin</td>
</tr>
<tr>
<td>Dev Emulation</td>
<td>: FBA</td>
</tr>
<tr>
<td>Dev Configuration</td>
<td>: 2-Way Mir</td>
</tr>
<tr>
<td>Pool State</td>
<td>: Enabled</td>
</tr>
<tr>
<td># of Devices in Pool</td>
<td>: 4</td>
</tr>
<tr>
<td># of Enabled Devices in Pool</td>
<td>: 4</td>
</tr>
<tr>
<td>Max. Subscription Percent</td>
<td>: None</td>
</tr>
</tbody>
</table>
Virtual Provisioning

Enabled Devices(4):

<table>
<thead>
<tr>
<th>Sym Dev</th>
<th>Total MBs</th>
<th>Alloc MBs</th>
<th>Free MBs</th>
<th>(%)</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>10E4</td>
<td>4314</td>
<td>1023</td>
<td>3290</td>
<td>23</td>
<td>Enabled</td>
</tr>
<tr>
<td>10E5</td>
<td>4314</td>
<td>1020</td>
<td>3293</td>
<td>23</td>
<td>Enabled</td>
</tr>
<tr>
<td>10E6</td>
<td>4314</td>
<td>248</td>
<td>4065</td>
<td>5</td>
<td>Enabled</td>
</tr>
<tr>
<td>10E7</td>
<td>4314</td>
<td>249</td>
<td>4064</td>
<td>5</td>
<td>Enabled</td>
</tr>
</tbody>
</table>

MBs 17256 2542 14713 14

Thin Devices(6):

<table>
<thead>
<tr>
<th>Sym Dev</th>
<th>Total MBs</th>
<th>Subs (%)</th>
<th>Alloc MBs</th>
<th>Alloc (%)</th>
<th>Wrtn MBs</th>
<th>Wrtn (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10F4</td>
<td>4314</td>
<td>25</td>
<td>635</td>
<td>4</td>
<td>574</td>
<td>3</td>
</tr>
<tr>
<td>10F5</td>
<td>4314</td>
<td>25</td>
<td>635</td>
<td>4</td>
<td>574</td>
<td>3</td>
</tr>
<tr>
<td>10F6</td>
<td>4314</td>
<td>25</td>
<td>635</td>
<td>4</td>
<td>574</td>
<td>3</td>
</tr>
<tr>
<td>10F7</td>
<td>4314</td>
<td>25</td>
<td>635</td>
<td>4</td>
<td>574</td>
<td>3</td>
</tr>
<tr>
<td>10F8</td>
<td>4314</td>
<td>25</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10F9</td>
<td>4314</td>
<td>25</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

MBs 25886 150 2542 15 2297 13

Preallocating pool space for thin devices

Contiguous extents on data devices can be preallocated when the thin devices are bound to a thin pool. Users may wish to preallocate space to certain thin volumes or they may choose to preallocate all thin volumes in their entirety. When space is preallocated that space will be reserved for the thin device starting at the beginning of the thin device. So if 100 MB is preallocated when a thin device is bound to a pool, the space for the first 100 MB of the thin device is allocated in the thin pool. Any writes to this area of the thin device will not result in additional thin pool allocation.

# symconfigure -sid 34 -cmd "create pool SALES type=thin;" commit -nop

# symconfigure -sid 34 -cmd "add dev 10E8:10EB to pool SALES type=thin, member_state=ENABLE;" commit -nop

Each of the data devices in the SALES pool is 4314 MB in size:

# symcfg show -sid 34 -pool SALES -thin -mb

Symmetrix ID: 000190103334

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

Symmetrix ID : 000190103334
Pool Name    : SALES
Pool Type    : Thin
Dev Emulation: FBA
Dev Configuration: 2-Way Mir
Pool State   : Enabled
# of Devices in Pool : 4
# of Enabled Devices in Pool : 4

Enabled Devices(4):

<table>
<thead>
<tr>
<th>Sym Device</th>
<th>Total MBs</th>
<th>Used MBs</th>
<th>Free MBs</th>
<th>Used (%)</th>
<th>Device State</th>
<th>Session Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>10E8</td>
<td>4314</td>
<td>0</td>
<td>4314</td>
<td>0</td>
<td>Enabled</td>
<td>N/A</td>
</tr>
<tr>
<td>10E9</td>
<td>4314</td>
<td>0</td>
<td>4314</td>
<td>0</td>
<td>Enabled</td>
<td>N/A</td>
</tr>
<tr>
<td>10EA</td>
<td>4314</td>
<td>0</td>
<td>4314</td>
<td>0</td>
<td>Enabled</td>
<td>N/A</td>
</tr>
<tr>
<td>10EB</td>
<td>4314</td>
<td>0</td>
<td>4314</td>
<td>0</td>
<td>Enabled</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Note: The Session Status field appears in this output because a uniform display is required for both TimeFinder/Snap save pools, which use sessions, and thin pools, which do not. This command produces the uniform view. Session Status has no meaning with thin pools. If run with the -detail option, the view is tailored to thin pools only and will not contain the Session Status column.

The amount of preallocation can be specified in megabytes, gigabytes, or cylinders, with cylinders as the default. A single command can be run that will bind both preallocated and nonpreallocated devices:

```
# symconfigure -sid 34 -cmd "bind tdev 10FA:10FB to pool SALES preallocate size=4314MB; bind tdev 10FC:1101 to pool SALES;" commit -nop
```

```
# symcfg show -sid 34 -pool SALES -thin -details -all -mb
```

Symmetrix ID: 000190103334
Pool Name    : SALES
Pool Type    : Thin
Dev Emulation: FBA
Dev Configuration: 2-Way Mir
Pool State   : Enabled
# of Devices in Pool : 4
# of Enabled Devices in Pool : 4
Max. Subscription Percent : None

Enabled Devices(4):

<table>
<thead>
<tr>
<th>Sym Device</th>
<th>Total MBs</th>
<th>Used MBs</th>
<th>Free MBs</th>
<th>Used (%)</th>
<th>Device State</th>
<th>Session Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>10E8</td>
<td>17256</td>
<td>0</td>
<td>17256</td>
<td>0</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>10E9</td>
<td>17256</td>
<td>0</td>
<td>17256</td>
<td>0</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>10EA</td>
<td>17256</td>
<td>0</td>
<td>17256</td>
<td>0</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>10EB</td>
<td>17256</td>
<td>0</td>
<td>17256</td>
<td>0</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Dev</td>
<td>Total MBs</td>
<td>Allocated MBs</td>
<td>Free Full MBs</td>
<td>(%)</td>
<td>State</td>
<td></td>
</tr>
<tr>
<td>-----</td>
<td>-----------</td>
<td>---------------</td>
<td>---------------</td>
<td>------</td>
<td>--------</td>
<td></td>
</tr>
<tr>
<td>10E8</td>
<td>4314</td>
<td>2202</td>
<td>2111</td>
<td>51</td>
<td>Enabled</td>
<td></td>
</tr>
<tr>
<td>10E9</td>
<td>4314</td>
<td>2173</td>
<td>2140</td>
<td>50</td>
<td>Enabled</td>
<td></td>
</tr>
<tr>
<td>10EA</td>
<td>4314</td>
<td>2165</td>
<td>2148</td>
<td>50</td>
<td>Enabled</td>
<td></td>
</tr>
<tr>
<td>10EB</td>
<td>4314</td>
<td>2092</td>
<td>2221</td>
<td>48</td>
<td>Enabled</td>
<td></td>
</tr>
<tr>
<td></td>
<td>17256</td>
<td>8634</td>
<td>8622</td>
<td>50</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Thin Devices(8):

<table>
<thead>
<tr>
<th>Dev</th>
<th>Total MBs</th>
<th>Subs (%)</th>
<th>Allocated MBs</th>
<th>Alloc (%)</th>
<th>Wrtn MBs</th>
<th>Wrtn (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10FA</td>
<td>4314</td>
<td>25</td>
<td>4314</td>
<td>25</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10FB</td>
<td>4314</td>
<td>25</td>
<td>4314</td>
<td>25</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10FC</td>
<td>4314</td>
<td>25</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10FD</td>
<td>4314</td>
<td>25</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10FE</td>
<td>4314</td>
<td>25</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10FF</td>
<td>4314</td>
<td>25</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1100</td>
<td>4314</td>
<td>25</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1101</td>
<td>4314</td>
<td>25</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>34515</td>
<td>200</td>
<td>8634</td>
<td>50</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

The SALES pool now has eight devices bound to it with two devices (10FA and 10FB) that are fully preallocated. The preallocated space (8628 MB, in total) is spread across all of the data devices and comprises half of the thin pool.

Allocating pool space for thin devices

Space can also be allocated for thin devices that are already bound to a pool. This might be required if it is determined that allocation is warranted for a thin device at some point after the device has been bound to a pool.

The PRODDB pool was created with four data devices and four thin devices, two of which were fully preallocated (1112:1113) and the other two (1110:1111) that were not:

```
# symcfg show -sid 34 -pool PRODDB -thin -details -all -mb

Symmtrix ID: 000190103334

Symmtrix ID: 000190103334
Pool Name: PRODDB
Pool Type: TP Data
Dev Emulation: FBA
Dev Configuration: 2-Way Mir
Pool State: Enabled
# of Devices in Pool: 4
# of Enabled Devices in Pool: 4
```
Virtual Provisioning

Max. Subscription Percent : None

Enabled Devices(4):
{
  ------------------------------------------------------
  Sym    Total      Alloc       Free Full  Device
  Dev    MBs        MBs        MBs  (%)  State
  ------------------------------------------------------
  1108   4315       2148       2167   49  Enabled
  1109   4315       2126       2189   49  Enabled
  110A   4315       1930       2386   44  Enabled
  110B   4315       2429       1887   56  Enabled
  MBs    17261       8633       8629   50
}

Thin Devices(4):
{
  -----------------------------------------------
  Pool          Pool              Pool
  Sym          Total  Subs        Allocated         Written
  Dev            MBs   (%)        MBs   (%)        MBs   (%)  Status
  -----------------------------------------------
  1110          4315    25          1     0          0     0  Bound
  1111          4315    25          1     0          0     0  Bound
  1112          4315    25       4316    25          0     0  Bound
  1113          4315    25       4316    25          0     0  Bound
  MBs          17261   100       8633    50          0     0

Along with a specific amount of allocated space, when allocation is performed on existing devices, the user must specify a region of the thin device for which to allocate space.

Device 1110 can be fully allocated using the following:

# symconfigure -sid 34 -cmd "allocate tdev 1110 start_cyl=0, end_cyl=last_cyl;"
commit -nop

# symcfg show -sid 34 -pool PRODDB -thin -details -all -mb

Symmetrix ID: 000190103334

Symmetrix ID : 000190103334
Pool Name : PRODDB
Pool Type : TP Data
Dev Emulation : FBA
Dev Configuration : 2-Way Mir
Pool State : Enabled
# of Devices in Pool : 4
# of Enabled Devices in Pool : 4
Max. Subscription Percent : None

Enabled Devices(4):
{
  ------------------------------------------------------
  Sym    Total      Alloc       Free Full  Device
  Dev    MBs        MBs        MBs  (%)  State
  ------------------------------------------------------
Virtual Provisioning

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

### Virtual Provisioning Details

<table>
<thead>
<tr>
<th>Dev</th>
<th>MBs</th>
<th>MBs</th>
<th>MBs (%)</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>1108</td>
<td>4315</td>
<td>3205</td>
<td>1111</td>
<td>74</td>
</tr>
<tr>
<td>1109</td>
<td>4315</td>
<td>3134</td>
<td>1182</td>
<td>72</td>
</tr>
<tr>
<td>110A</td>
<td>4315</td>
<td>2957</td>
<td>1358</td>
<td>68</td>
</tr>
<tr>
<td>110B</td>
<td>4315</td>
<td>3652</td>
<td>664</td>
<td>84</td>
</tr>
</tbody>
</table>

### Thin Devices (4):

<table>
<thead>
<tr>
<th>Sym Dev</th>
<th>Pool Total MBs (%)</th>
<th>Pool Allocated MBs (%)</th>
<th>Pool Written MBs (%)</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>1110</td>
<td>4315 25</td>
<td>4316 25</td>
<td>0</td>
<td>0 Bound</td>
</tr>
<tr>
<td>1111</td>
<td>4315 25</td>
<td>1 0</td>
<td>0</td>
<td>0 Bound</td>
</tr>
<tr>
<td>1112</td>
<td>4315 25</td>
<td>4316 25</td>
<td>0</td>
<td>0 Bound</td>
</tr>
<tr>
<td>1113</td>
<td>4315 25</td>
<td>4316 25</td>
<td>0</td>
<td>0 Bound</td>
</tr>
</tbody>
</table>

### Allocation Example

If allocation is desired for only a portion of device 1111 that can be accomplished by starting the command with a cylinder number that represents a specific point on the thin device. A size (MB, GB, or CYL) can also be specified that will determine how much space is allocated.

Device 1111 is 4603 cylinders in size:

```bash
# symdev show 1111 -sid 34 | grep Cylinders
Cylinders : 4603
```

To allocate 1,000 cylinders starting at cylinder 2303:

```bash
# symconfigure -sid 34 -cmd "allocate tdev 1111 start_cyl=2303 size=1000;" commit -nop
# symcfg show -sid 34 -pool PRODDB -thin -details -all -mb
```

---

---
Virtual Provisioning

<table>
<thead>
<tr>
<th>Sym</th>
<th>Total MBs</th>
<th>Alloc MBs</th>
<th>Free MBs</th>
<th>Full (%)</th>
<th>Device State</th>
</tr>
</thead>
<tbody>
<tr>
<td>1108</td>
<td>4315</td>
<td>3422</td>
<td>894</td>
<td>79</td>
<td>Enabled</td>
</tr>
<tr>
<td>1109</td>
<td>4315</td>
<td>3364</td>
<td>952</td>
<td>77</td>
<td>Enabled</td>
</tr>
<tr>
<td>110A</td>
<td>4315</td>
<td>3191</td>
<td>1125</td>
<td>73</td>
<td>Enabled</td>
</tr>
<tr>
<td>110B</td>
<td>4315</td>
<td>3910</td>
<td>406</td>
<td>90</td>
<td>Enabled</td>
</tr>
</tbody>
</table>

---

**Thin Devices (4):**

<table>
<thead>
<tr>
<th>Sym</th>
<th>Total MBs</th>
<th>Subs</th>
<th>Alloc MBs</th>
<th>(%)</th>
<th>Written MBs</th>
<th>(%)</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>1110</td>
<td>4315</td>
<td>25</td>
<td>4316</td>
<td>25</td>
<td>0</td>
<td>0</td>
<td>Bound</td>
</tr>
<tr>
<td>1111</td>
<td>4315</td>
<td>25</td>
<td>939</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>Bound</td>
</tr>
<tr>
<td>1112</td>
<td>4315</td>
<td>25</td>
<td>4316</td>
<td>25</td>
<td>0</td>
<td>0</td>
<td>Bound</td>
</tr>
<tr>
<td>1113</td>
<td>4315</td>
<td>25</td>
<td>4316</td>
<td>25</td>
<td>0</td>
<td>0</td>
<td>Bound</td>
</tr>
</tbody>
</table>

---

**Freeing allocated pool space from thin devices**

Space that has been allocated or preallocated for a thin device but not yet written to can be freed and made available to the other devices in the thin pool.

The ability to free up space should typically be used by a storage administrator when there is a greater amount of preallocated or allocated space than required on a thin device. If a device has been fully allocated, it is consuming a significant amount of space in the pool. If the device is not being fully written to, this may be wasted space in the pool that could be better utilized by other thin devices. In this instance, the storage administrator can use the **free** command to free all the space that has been allocated but not written to.

Here, the allocated space on the second half of device 1110 is freed:

```
# symconfigure -sid 34 -cmd "free tdev 1110 start_cyl=2303 end_cyl=last_cyl;"
commit -nop

# symcfg show -sid 34 -pool PRODDB -thin -details -all -mb
```

Symmetrix ID: 000190103334

Symmetrix ID : 000190103334
Pool Name : PRODDB
Pool Type : TP Data
Dev Emulation : FBA
Virtual Provisioning

Dev Configuration: 2-Way Mir
Pool State: Enabled
# of Devices in Pool: 4
# of Enabled Devices in Pool: 4
Max. Subscription Percent: None

Enabled Devices (4):

<table>
<thead>
<tr>
<th>Sym</th>
<th>Total</th>
<th>Alloc</th>
<th>Free</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MBs</td>
<td>MBs</td>
<td>MBs</td>
<td>(%)</td>
</tr>
<tr>
<td>1108</td>
<td>4315</td>
<td>2897</td>
<td>1418</td>
<td>67</td>
</tr>
<tr>
<td>1109</td>
<td>4315</td>
<td>2892</td>
<td>1423</td>
<td>67</td>
</tr>
<tr>
<td>110A</td>
<td>4315</td>
<td>2652</td>
<td>1663</td>
<td>61</td>
</tr>
<tr>
<td>110B</td>
<td>4315</td>
<td>3288</td>
<td>1027</td>
<td>76</td>
</tr>
</tbody>
</table>

Thin Devices (4):

<table>
<thead>
<tr>
<th>Sym</th>
<th>Total</th>
<th>Subs</th>
<th>Allocated</th>
<th>Written</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MBs</td>
<td>(%)</td>
<td>MBs (%)</td>
<td>MBs (%)</td>
<td></td>
</tr>
<tr>
<td>1110</td>
<td>4315</td>
<td>25</td>
<td>2159</td>
<td>0</td>
<td>Bound</td>
</tr>
<tr>
<td>1111</td>
<td>4315</td>
<td>25</td>
<td>939</td>
<td>0</td>
<td>Bound</td>
</tr>
<tr>
<td>1112</td>
<td>4315</td>
<td>25</td>
<td>4316</td>
<td>0</td>
<td>Bound</td>
</tr>
<tr>
<td>1113</td>
<td>4315</td>
<td>25</td>
<td>4316</td>
<td>0</td>
<td>Bound</td>
</tr>
</tbody>
</table>

Setting a maximum subscription percent

In all previous examples, the thin pools were oversubscribed, meaning that there was less physical storage space in the thin pool than total space of all the thin devices bound to the pool. Pools can also be undersubscribed. A limit can be set to prevent the subscription percent for a specific pool from going beyond a level set by the user.

The maximum subscription percent (or max subs percent) can be set to "none" or to any value between 0 and 65534. The default value is “none”, meaning that no limit is set and the amount of oversubscription is not restricted. Setting a max subs percent of 0 will prevent any thin devices from being bound to a pool. Setting the value to 100 means that the pool cannot be oversubscribed.

This limit can be set by specifying the max_subs_percent value when creating a thin pool. In this example, the limit is set at 150 percent for a pool called DOCS, meaning that the total amount of configured space for bound thin devices cannot exceed 150 percent of the space provided by
Virtual Provisioning

the data devices:

```shell
# symconfigure -sid 34 -cmd "create pool DOCS type=thin max_subs_percent=150;"
commit -nop -nop
```

Existing thin pools can be modified to set a maximum subscription percent.

The SALES pool, which has no limit set, is currently oversubscribed at 200 percent because it has twice as much thin device space as data device space. This is indicated by the total in the “Subscription %” column (Subs %).

```shell
# symcfg show -sid 34 -pool SALES -thin -details -all -mb
```

Symmetrix ID: 000190103334

<table>
<thead>
<tr>
<th>Symmetrix ID</th>
<th>: 000190103334</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pool Name</td>
<td>: SALES</td>
</tr>
<tr>
<td>Pool Type</td>
<td>: Thin</td>
</tr>
<tr>
<td>Dev Emulation</td>
<td>: FBA</td>
</tr>
<tr>
<td>Dev Configuration</td>
<td>: 2-Way Mir</td>
</tr>
<tr>
<td>Pool State</td>
<td>: Enabled</td>
</tr>
<tr>
<td># of Devices in Pool</td>
<td>: 4</td>
</tr>
<tr>
<td># of Enabled Devices in Pool</td>
<td>: 4</td>
</tr>
<tr>
<td>Max. Subscription Percent</td>
<td>: None</td>
</tr>
</tbody>
</table>

Enabled Devices(4):

```plaintext
{  
<table>
<thead>
<tr>
<th>Sym</th>
<th>Total</th>
<th>Alloc</th>
<th>Free</th>
<th>Full</th>
<th>Device</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MBs</td>
<td>MBs</td>
<td>MBs (%)</td>
<td>(%)</td>
<td>State</td>
</tr>
<tr>
<td>-------------</td>
<td>----------</td>
<td>---------</td>
<td>----------</td>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td>10E8</td>
<td>4314</td>
<td>2202</td>
<td>2111</td>
<td>51</td>
<td>Enabled</td>
</tr>
<tr>
<td>10E9</td>
<td>4314</td>
<td>2173</td>
<td>2140</td>
<td>50</td>
<td>Enabled</td>
</tr>
<tr>
<td>10EA</td>
<td>4314</td>
<td>2165</td>
<td>2148</td>
<td>50</td>
<td>Enabled</td>
</tr>
<tr>
<td>10EB</td>
<td>4314</td>
<td>2092</td>
<td>2221</td>
<td>48</td>
<td>Enabled</td>
</tr>
</tbody>
</table>

---------- ---------- ---------- -----
MBs         17256     8634     8622     50
}
```

Thin Devices(8):

```plaintext
{  
<table>
<thead>
<tr>
<th>Sym</th>
<th>Total</th>
<th>Subs</th>
<th>Alloc</th>
<th>Alloc</th>
<th>Wrtn</th>
<th>Wrtn</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MBs</td>
<td>(%)</td>
<td>MBs</td>
<td>(%)</td>
<td>MBs</td>
<td>(%)</td>
</tr>
<tr>
<td>-------------</td>
<td>----------</td>
<td>--------</td>
<td>-------</td>
<td>-------</td>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>10FA</td>
<td>4314</td>
<td>25</td>
<td>4314</td>
<td>25</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10FB</td>
<td>4314</td>
<td>25</td>
<td>4314</td>
<td>25</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10FC</td>
<td>4314</td>
<td>25</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10FD</td>
<td>4314</td>
<td>25</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10FE</td>
<td>4314</td>
<td>25</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10FF</td>
<td>4314</td>
<td>25</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1100</td>
<td>4314</td>
<td>25</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1101</td>
<td>4314</td>
<td>25</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

---------- ---------- ---------- ------
```
Thin pool attributes can be modified by using `set pool` and choosing the desired `max_subs_percent` value:

```
# symconfigure -sid 34 -cmd "set pool SALES, type=thin, max_subs_percent=225;"
commit -nop
```

The `symcfg` show output now displays the maximum subscription percent:

```
# symcfg show -sid 34 -pool SALES -thin -details -all -mb

Symmetrix ID: 000190103334

Symmetrix ID : 000190103334
Pool Name : SALES
Pool Type : Thin
Dev Emulation : FBA
Dev Configuration : 2-Way Mir
Pool State : Enabled
# of Devices in Pool : 4
# of Enabled Devices in Pool : 4
Max. Subscription Percent : 225
```

The maximum subscription percent will prevent the user from adding thin devices that push the limit past 225 percent. In this case, each thin device represents 25 percent of the space available on the data devices, so attempting to add two more thin devices will fail:

```
# symconfigure -sid 34 -cmd "bind tdev 1102:1103 to pool SALES;" commit -nop
A Configuration Change operation is in progress. Please wait...
Establishing a configuration change session.................Established.
Error occurred while Defining change number 1:
The requested operation will exceed the pool over-subscription limit
Device 1103 generated the failure
Terminating the configuration change session..............Done.
```

Setting the maximum subscription percent back to none will mean that the amount of subscription is no longer restricted on the SALES pool.

```
# symconfigure -sid 34 -cmd "set pool SALES, type=thin, max_subs_percent=None;"
commit -nop -nop
```

The setting will now show as “None” in the query output:

```
# symcfg show -sid 34 -pool SALES -thin -details -all -mb
```
Unbinding thin devices from a pool

In order to unbind thin devices from a pool they must be unmapped or Not Ready. In this case, neither of the two devices being unbound is mapped to an FA:

```bash
# symconfigure -sid 34 -cmd "unbind tdev 1100:1101 from pool SALES;" commit -nop
```

They have now been unbound and are free to bind to another pool or delete:

```bash
# symcfg show -pool SALES -thin -detail -all -sid 34
```

```bash
Symmetrix ID: 000190103334
Symmetrix ID : 000190103334
Pool Name    : SALES
Pool Type    : Thin
Dev Emulation: FBA
Dev Configuration : 2-Way Mir
Pool State   : Enabled
# of Devices in Pool : 4
# of Enabled Devices in Pool : 4
Max. Subscription Percent : None

Enabled Devices(4):
{
  -------------------------------
  Sym   Total  Alloc  Free Full  Device
  Dev   Tracks Tracks Tracks (%)  State
  -------------------------------
  10E8   69024  35244  33780   51  Enabled
  10E9   69024  34752  34272   50  Enabled
  10EA   69024  34644  34380   50  Enabled
  10EB   69024  33480  35544   48  Enabled
  -------------------------------
  Tracks 276096 138120 137976  50
}

Thin Devices(6):
{
  -------------------------------
  Sym  Total  Alloc  Alloc  Wrtn  Wrtn
  Tracks 276096 138120 137976  50
}
```
Disabling and removing data devices from a pool

Support for the draining of data devices has been added to Enginuity 5874 for Symmetrix VMAX arrays. This allows data devices to be removed from thin pools non-disruptively.

Before removing a data device from a pool, the device must be disabled. Disabling a data device changes its state from Enabled to Draining. The concept of device draining is common to all pool types (data, DSE, and snap pools).

When a device enters a draining state, used tracks on that device are copied to the remaining, enabled devices in the pool. Therefore, data devices can be removed from a pool and repurposed without having to unbind thin devices that may have allocated extents associated with the data device:

```bash
# symconfigure -sid 34 -cmd "disable dev 10EB in pool SALES, type=thin;" commit -nop
```

If the `symcfg show` command is run with the `–all` option, the disabled device will be displayed. In this case, the command was run while the device was still draining:

```bash
# symcfg show -pool SALES -thin -all -sid 34
```

Symmetrix ID: 000190103334

Symmetrix ID : 000190103334
Pool Name : SALES
Pool Type : Thin
Dev Emulation : FBA
Dev Configuration : 2-Way Mir
Pool State : Enabled
# of Devices in Pool : 4
# of Enabled Devices in Pool : 3

Enabled Devices(3):

<table>
<thead>
<tr>
<th>Sym</th>
<th>Total Tracks</th>
<th>Used Tracks</th>
<th>Free Tracks</th>
<th>(%) State</th>
<th>Device Status</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Best Practices for Fast, Simple Capacity Allocation with Symmetrix Virtual Provisioning
After it has drained, the state will be “Disabled”. The used tracks on device 10EB have been copied to the three enabled devices in the pool:

```
# symcfg show -pool SALES -thin -all -sid 34

Symmetrix ID: 000190103334
Symmetrix ID: 000190103334
Pool Name: SALES
Pool Type: Thin
Dev Emulation: FBA
Dev Configuration: 2-Way Mir
Pool State: Enabled
# of Devices in Pool: 4
# of Enabled Devices in Pool: 3

Enabled Devices(3):

<table>
<thead>
<tr>
<th>Sym</th>
<th>Total Tracks</th>
<th>Used Tracks</th>
<th>Free Tracks</th>
<th>Full (%)</th>
<th>Device State</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>10E8</td>
<td>69024</td>
<td>46404</td>
<td>22520</td>
<td>67</td>
<td>Enabled</td>
<td>N/A</td>
</tr>
<tr>
<td>10E9</td>
<td>69024</td>
<td>45924</td>
<td>21100</td>
<td>66</td>
<td>Enabled</td>
<td>N/A</td>
</tr>
<tr>
<td>10EA</td>
<td>69024</td>
<td>45792</td>
<td>23232</td>
<td>66</td>
<td>Enabled</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Tracks: 207072

Disabled Devices(1):

<table>
<thead>
<tr>
<th>Sym</th>
<th>Total Tracks</th>
<th>Used Tracks</th>
<th>Free Tracks</th>
<th>Full (%)</th>
<th>Device State</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>10EB</td>
<td>69024</td>
<td>9864</td>
<td>59160</td>
<td>14</td>
<td>Draining</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Tracks: 69024
```
Device 10EB can now be removed:

```bash
# symconfigure -sid 34 -cmd "remove dev 10EB from pool SALES, type=thin;" commit -nop

# symcfg show -pool SALES -thin -all -sid 34
```

Symmetrix ID: 000190103334

<table>
<thead>
<tr>
<th>Symmetrix ID</th>
<th>000190103334</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pool Name</td>
<td>SALES</td>
</tr>
<tr>
<td>Pool Type</td>
<td>Thin</td>
</tr>
<tr>
<td>Dev Emulation</td>
<td>FBA</td>
</tr>
<tr>
<td>Dev Configuration</td>
<td>2-Way Mir</td>
</tr>
<tr>
<td>Pool State</td>
<td>Enabled</td>
</tr>
<tr>
<td># of Devices in Pool</td>
<td>3</td>
</tr>
<tr>
<td># of Enabled Devices in Pool</td>
<td>3</td>
</tr>
</tbody>
</table>

Enabled Devices(3):

```
{
   Sym         Total      Used      Free  Full  Device    Session
   Dev        Tracks    Tracks    Tracks   (%)  State     Status
   -----------------------------------------------
    10E8       69024     46404     22620    67  Enabled   N/A
    10E9       69024     45924     23100    66  Enabled   N/A
    10EA       69024     45792     23232    66  Enabled   N/A
```

Deleting thin pools

Before deleting a thin pool, any bound thin devices must be unbound, causing all data to be discarded. All data devices must also be disabled and removed.

```bash
# symconfigure -sid 34 -cmd "unbind tdev 10FA:10FF from pool SALES;" commit -nop

# symconfigure -sid 34 -cmd "disable dev 10E8:10EA in pool SALES, type=thin;" commit -nop

# symconfigure -sid 34 -cmd "remove dev 10E8:10EA from pool SALES, type=thin;" commit -nop
```

The pool is now empty and can be deleted.

```
# symcfg show -pool SALES -thin -details -all -sid 34
```

Symmetrix ID: 000190103334

No devices found in Device Pool SALES.

```bash
# symconfigure -sid 34 -cmd "delete pool SALES, type=thin;" commit -nop

# symcfg list -pools -thin -sid 34
```

Symmetrix ID: 000190103334
Virtual Provisioning

### Symmetrix Pools

<table>
<thead>
<tr>
<th>Pool Name</th>
<th>Type</th>
<th>Dev</th>
<th>Emul</th>
<th>Config</th>
<th>Total Tracks</th>
<th>Enabled Tracks</th>
<th>Used Tracks</th>
<th>Free Tracks</th>
<th>Total (%)</th>
<th>Full Tracks</th>
</tr>
</thead>
<tbody>
<tr>
<td>HR</td>
<td>TH</td>
<td>FBA</td>
<td>2-Way Mir</td>
<td></td>
<td>276096</td>
<td>276096</td>
<td>40680</td>
<td>235416</td>
<td>14 Ena</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>276096</td>
<td>276096</td>
<td>40680</td>
<td>235416</td>
<td>14 Ena</td>
<td></td>
</tr>
</tbody>
</table>

Legend for Pool Types:

SN = Snap, RS = Rdfa DSE TH = Thin Data

### Configuring thin metadevices

Thin devices can also be created and formed into metadevices.

Metadevices made from thin volumes are created using the same command syntax as metadevices created from regular volumes.

Thin metadevices must be created before the thin devices are bound to a thin pool. If an attempt is made to form a metadevice from bound devices, the command will fail.

In the following example, a four-member, concatenated metadevice is created from thin devices:

```
# symconfigure -sid 34 -cmd "form meta from dev 1110, config=concatenated; add dev 1111:1113 to meta 1110;" commit -nop
```

A Configuration Change operation is in progress. Please wait...

Establishing a configuration change session............Established.
Processing symmetrix 000190103334
Performing Access checks.................................Allowed.
Checking Device Reservations............................Allowed.
Submitting configuration changes.........................Submitted
Locking devices...........................................Locked.
Validating configuration changes.........................Validated.
Initiating PREPARE of configuration changes...............Prepared.
Initiating COMMIT -NOP of configuration changes...........Queued.
COMMIT -NOP requesting required resources.................Obtained.
Step 007 of 009 steps.................................Executing.
Local: COMMIT -NOP.....................................Done.
Terminating the configuration change session.............Done.

The configuration change session has successfully completed.

Note: When forming thin metadevices, the symconfigure form meta parameter count=<member_count> is not supported. The automatic metadevice creation feature, however, can be used. The EMC Solutions Enabler Symmetrix Array Controls CLI Version 7.1 Product Guide has more details.
The following is the output of `symdev show` run against the newly created thin metadevice:

```
# symdev show 1110 -sid 34

Device Physical Name     : Not Visible
Device Symmetrix Name    : 1110
Device Serial ID         : N/A
Symmetrix ID             : 000190103334
Attached BCV Device      : N/A
Attached VDEV TGT Device : N/A
Vendor ID                : EMC
Product ID               : SYMMETRIX
Product Revision         : 5773
Device WWN               : 600604800019010334533031313130
Device Emulation Type    : FBA
Device Defined Label Type: N/A
Device Defined Label     : N/A
Device Sub System Id     : 0x0009
Cache Partition Name     : DEFAULT_PARTITION
DATA Device Pool Name    : N/A
Device Block Size        : 512

Device Capacity
{
  Cylinders            :      18412
  Tracks               :     276180
  512-byte Blocks      :   35351040
  MegaBytes            :      17261
  KiloBytes            :   17675520
}

Device Configuration     : TDEV            (Meta Head,
                          Non-Exclusive Access)

Device is WORM Enabled   : No
Device is WORM Protected : No
SCSI-3 Persistent Reserve: Disabled
Dynamic Spare Invoked   : No
Dynamic RDF Capability  : None
STAR Mode               : No
STAR Recovery Capability: None
STAR Recovery State     : NA
Device Service State    : Normal
Device Status           : Not Ready        (NR)
Device SA Status        : N/A              (N/A)
```
Virtual Provisioning

Meta Configuration : Concatenated
Meta Device Members (4) :

```
----------------------------  --------------------------
| Sym    | Cap (MB) | Std Inv Tracks | BCV Inv Tracks | State         |
| Dev    |          |               |               |               |
------------------------------  --------------------------
| 1110   | 4315     | -             | -             | N/A           |
| 1111   | 4315     | -             | -             | N/A           |
| 1112   | 4315     | -             | -             | N/A           |
| 1113   | 4315     | -             | -             | N/A           |
------------------------------  --------------------------
| 17261  |          |               |               |               |
```

Mirror Set Type : [Thin,N/A,N/A,N/A]
Mirror Set DA Status : [N/A,N/A,N/A,N/A]
Mirror Set Inv. Tracks : [0,0,0,0]

Back End Disk Director Information

```
Hyper Type : Thin
Hyper Status : Ready (RW)
Disk [Director, Interface, TID] : [N/A,N/A,N/A]
Disk Director Volume Number : N/A
Hyper Number : N/A
```

The `symdev list` command, when run with the `-tdev` option, will return a list of only thin devices. In this case, meta head 1110 and members 1111, 1112, and 1113 are shown in the output.

Of the thin devices that have been created in the DMX-4, only devices 10F4 through 10F7 have been bound to a thin pool:

```
# symdev list -sid 34 -tdev
Symmetrix ID: 000190103334

Device Name          Directors                   Device
--------------------------- ------------- -------------------------------------
Cap  Sym  Physical               SA :P DA :IT  Config        Attribute    Sts   (MB)
--------------------------- ------------- -------------------------------------
10F4 Not Visible        ***:*  NA:NA TDEV N/Grp’d     RW    4314
10F5 Not Visible        ***:*  NA:NA TDEV N/Grp’d     RW    4314
10F6 Not Visible        ***:*  NA:NA TDEV N/Grp’d     RW    4314
10F7 Not Visible        ***:*  NA:NA TDEV N/Grp’d     RW    4314
10F8 Not Visible        ????:? NA:NA TDEV N/Grp’d     NR    4314
10F9 Not Visible        ????:? NA:NA TDEV N/Grp’d     NR    4314
10FA Not Visible         ????:? NA:NA TDEV N/Grp’d     NR    4314
10FB Not Visible         ????:? NA:NA TDEV N/Grp’d     NR    4314
```
The meta can then be bound to a pool. In this case, the pool is called “meta” and has already been created. As with other SYMCLI commands acting on metadevices, only the meta head is specified in the command syntax.

```
# symconfigure -sid 34 -cmd "bind tdev 1110 to pool meta;" commit -nop
```

A Configuration Change operation is in progress. Please wait...

Establishing a configuration change session.................Established.
Processing symmetrix 000190103334
Performing Access checks.....................................Allowed.
Checking Device Reservations.................................Allowed.
Locking devices...............................................Locked.
Validating configuration changes..............................Validated.
Preparing configuration changes...............................Prepared.
Commit -nopting configuration changes.........................Commit -nopted.
Terminating the configuration change session..................Done.

The configuration change session has successfully completed.

Once the metadevice is bound to the pool, it is read/write enabled.

Note that devices 10F4 through 10F7 have been configured by the Solaris operating system and symcfg discover has been run, so the physical device name field is now populated for those devices:

```
# symdev list -sid 34 -tdev
```

Symmetrix ID: 000190103334

<table>
<thead>
<tr>
<th>Device Name</th>
<th>Directors</th>
<th>Device</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SA :P DA :IT Config Attribute Sts (MB)</td>
<td></td>
</tr>
<tr>
<td>Sym</td>
<td>Physical</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Best Practices for Fast, Simple Capacity Allocation with Symmetrix Virtual Provisioning
Like metadevices made up of regular Symmetrix volumes, concatenated thin metadevices can also be expanded. The command syntax to perform this operation with thin volumes is identical.

The following example shows the expansion of the existing meta. No further action is required to read/write enable the new metamember. Because it is now part of a metadevice that is bound to a thin pool, it is bound to the same pool:

```
symconfigure -sid 34 -cmd "add dev 1114 to meta 1110;" commit -nop
```

Device 1110 is now a five-member meta:

```
# symdev show 1110 -sid 34

  Device Physical Name    : Not Visible
  Device Symmetrix Name   : 1110
  Device Serial ID        : N/A
  Symmetrix ID            : 000190103334
  Attached BCV Device     : N/A
  Attached VDEV TGT Device: N/A
  Vendor ID               : EMC
  Product ID              : SYMMETRIX
```
Virtual Provisioning Best Practices for Fast, Simple Capacity Allocation with Symmetrix Virtual Provisioning

- **Product Revision**: 5773
- **Device WWN**: 60060480000190103334533031313130
- **Device Emulation Type**: FBA
- **Device Defined Label Type**: N/A
- **Device Defined Label**: N/A
- **Device Sub System Id**: 0x0009
- **Cache Partition Name**: DEFAULT_PARTITION
- **DATA Device Pool Name**: meta
- **Device Block Size**: 512
- **Device Capacity**:
  - Cylinders: 23015
  - Tracks: 345225
  - 512-byte Blocks: 44188800
  - MegaBytes: 21577
  - KiloBytes: 22094400
- **Device Configuration**: TDEV (Meta Head, Non-Exclusive Access)
- **Device is WORM Enabled**: No
- **Device is WORM Protected**: No
- **SCSI-3 Persistent Reserve**: Disabled
- **Dynamic Spare Invoked**: No
- **Dynamic RDF Capability**: None
- **STAR Mode**: No
- **STAR Recovery Capability**: None
- **STAR Recovery State**: NA
- **Device Service State**: Normal
- **Device Status**: Not Ready (NR)
- **Device SA Status**: Ready (RW)

Front Director Paths (2):

<table>
<thead>
<tr>
<th>PdevName</th>
<th>Type</th>
<th>Type Num</th>
<th>Sts</th>
<th>VBUS</th>
<th>TID</th>
<th>SYMM</th>
<th>Host</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not Visible</td>
<td>N/A</td>
<td>FA</td>
<td>08A:0</td>
<td>RW</td>
<td>000</td>
<td>009</td>
<td>N/A</td>
</tr>
<tr>
<td>Not Visible</td>
<td>N/A</td>
<td>FA</td>
<td>09A:1</td>
<td>RW</td>
<td>000</td>
<td>009</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Meta Configuration: Concatenated

Meta Device Members (5):

<table>
<thead>
<tr>
<th>Sym</th>
<th>Cap (MB)</th>
<th>Std Inv</th>
<th>BCV Inv</th>
<th>Pair</th>
<th>R1 Inv</th>
<th>R2 Inv</th>
<th>Pair</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Tracks</td>
<td>Tracks</td>
<td>State</td>
<td>Tracks</td>
<td>Tracks</td>
<td>State</td>
</tr>
</tbody>
</table>
**Virtual Provisioning**

<table>
<thead>
<tr>
<th>Device</th>
<th>Total Tracks</th>
<th>Alloc Tracks</th>
<th>Free Tracks</th>
<th>Full (%)</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>1110</td>
<td>4315</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>1111</td>
<td>4315</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>1112</td>
<td>4315</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>1113</td>
<td>4315</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>1114</td>
<td>4315</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Total Tracks: 21577

Mirror Set Type: [Thin, Thin, N/A, N/A]
Mirror Set DA Status: [N/A, N/A, N/A, N/A]
Mirror Set Inv. Tracks: [0, 0, 0, 0]

---

Back End Disk Director Information

```
Hyper Type                : Thin
Hyper Status              : Ready (RW)
Disk [Director, Interface, TID] : [N/A, N/A, N/A]
Disk Director Volume Number : N/A
Hyper Number              : N/A
```

# symcfg show -sid 34 -pool meta -detail -thin -all

Symmetrix ID: 000190103334
Symmetrix ID: 000190103334
Pool Name: meta
Pool Type: Thin
Dev Emulation: FBA
Dev Configuration: 2-Way Mir
Pool State: Enabled
# of Devices in Pool: 2
# of Enabled Devices in Pool: 2
Max. Subscription Percent: None

Enabled Devices(2):

```
------------------------------------------------------
Sym         Total      Alloc       Free Full  Device
Dev        Tracks     Tracks     Tracks  (%)  State
------------------------------------------------------
1104        69045         24      69021    0  Enabled
1105        69045         36      69009    0  Enabled
------------------------------------------------------
Tracks      138090         60     138030    0
```

Thin Devices(1):

```
Sym          Total  Subs      Alloc Alloc       Wrtn  Wrtn
Dev         Tracks   (%)     Tracks   (%)     Tracks   (%)
----------------------------------------------------------
1110        345210   250         12     0          1     0
----------------------------------------------------------
Tracks      345210   250         12     0          1     0
```
Members can also be removed from concatenated metadevices. In order to perform this change, the meta must be unmasked or unmapped.

```bash
# symconfigure -sid 34 -cmd "unmap dev 1110 from dir 8a:0;unmap dev 1110 from dir 9a:1;" commit -nop

# symconfigure -sid 34 -cmd "remove dev 1114 from meta 1110;" commit -nop
```

If a metamember is removed from a metadevice it will remain bound to the thin pool.

```bash
# symcfg -sid 34 show -pool meta -details -all -thin
```

If the device does not need to be bound to the pool any longer, it can be removed.

```bash
# symconfigure -sid 34 -cmd "unbind tdev 1114 from pool meta;" commit -nop
```
Virtual Provisioning

# symcfg -sid 34 show -pool meta -details -all -thin

Symmetrix ID: 000190103334

Symmetrix ID                  : 000190103334
Pool Name                     : meta
Pool Type                     : Thin
Dev Emulation                 : FBA
Dev Configuration             : 2-Way Mir
Pool State                    : Enabled
# of Devices in Pool          : 2
# of Enabled Devices in Pool  : 2
Max. Subscription Percent     : None

Enabled Devices(2):
{
   Sym    Total     Alloc     Free    Full  Device
   Dev    Tracks    Tracks    Tracks  (%)  State
   ------------------------------------------------------
   1104    69045     24       69021   0  Enabled
   1105    69045     24       69021   0  Enabled
   ------------------------------------------------------
   Tracks  138090    48       138042  0
}

Thin Devices(1):
{
   Sym    Total     Subs     Alloc     Alloc     Wrtn   Wrtn
   Dev    Tracks    (%)      Tracks    (%)      Tracks   (%)
   ----------------------------------------------------------
   1110    276180   200       48        0       4        0
   ----------------------------------------------------------
   Tracks  276180   200       48        0       4        0
}

Monitoring thin pool space

The symcfg monitor command can be used to monitor thin pool consumption. It can be run with the -i <interval> and -c <count> options so that the user can see how much space is consumed in the pool over a given period of time. If symcfg monitor is run with the -i 120 and -c 2 parameters, the command will run once when it is entered and again 2 minutes later and then the prompt will return:

# symcfg monitor -sid 34 -thin -pool HR -i 120 -c 2 -mb

Symmetrix ID: 000190103334

SYM METRIX DATA DEVICES

<table>
<thead>
<tr>
<th>Dev</th>
<th>Emul</th>
<th>Config</th>
<th>Pool</th>
<th>Type</th>
<th>Pool Name</th>
<th>State</th>
<th>Total</th>
<th>Used Full</th>
</tr>
</thead>
<tbody>
<tr>
<td>----</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
During the 2-minute interval that was being monitored, 996 megabytes were used from the HR thin pool.

The symcfg monitor command also allows the user to specify a script to call when the save pool reaches a specified percent. The script could perform any function that the user can script, such as sending an e-mail or writing to a log file. In this example, symcfg monitor will be run every 5 minutes until the pool reaches 60 percent.

At the first interval after the threshold has been reached, a script called vp_action_script will be called that writes a dated entry to a log file:

```bash
# symcfg monitor -sid 34 -percent 60 -action /opt/scripts/vp_action_script.sh -pool HR -thin -i 300
# scripts/vp_action_script.sh -pool HR -thin -i 300
```
### Virtual Provisioning

<table>
<thead>
<tr>
<th>Sym</th>
<th>Emul</th>
<th>Config</th>
<th>Dev</th>
<th>Pool</th>
<th>Type</th>
<th>Pool Name</th>
<th>State</th>
<th>Total Tracks</th>
<th>Used Full Tracks</th>
<th>(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10E4 FBA</td>
<td>2-Way Mir</td>
<td>TH HR</td>
<td>Ena</td>
<td>69024</td>
<td>2-Way Mir</td>
<td>TH HR</td>
<td>Ena</td>
<td>69024</td>
<td>20328</td>
<td>29</td>
</tr>
<tr>
<td>10E5 FBA</td>
<td>2-Way Mir</td>
<td>TH HR</td>
<td>Ena</td>
<td>69024</td>
<td>2-Way Mir</td>
<td>TH HR</td>
<td>Ena</td>
<td>69024</td>
<td>20280</td>
<td>29</td>
</tr>
<tr>
<td>10E6 FBA</td>
<td>2-Way Mir</td>
<td>TH HR</td>
<td>Ena</td>
<td>69024</td>
<td>2-Way Mir</td>
<td>TH HR</td>
<td>Ena</td>
<td>69024</td>
<td>8004</td>
<td>11</td>
</tr>
<tr>
<td>10E7 FBA</td>
<td>2-Way Mir</td>
<td>TH HR</td>
<td>Ena</td>
<td>69024</td>
<td>2-Way Mir</td>
<td>TH HR</td>
<td>Ena</td>
<td>69024</td>
<td>8004</td>
<td>11</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>276096</td>
<td>56616</td>
<td>20</td>
</tr>
</tbody>
</table>

Legend for Pool Types:

- SN = Snap
- RS = Rdfa DSE
- TH = Thin Data
- = N/A

---

Script `/opt/scripts/vp_action_script.sh` executed SUCCESSFULLY

# cat /var/adm/vp_messages/vp_logfile

Mon Feb  4 20:55:32 EST 2008

The set threshold of 60% for thin pool HR has been exceeded.
Note: The log file shown here is user created. Virtual Provisioning related errors and events are logged to the symapi logs like all Solutions Enabler commands. There is no default log for output written by user scripts that are called by symcfg monitor.

Local replication using TimeFinder/Snap

Thin devices can be used with TimeFinder/Snap in the same way that other types of Symmetrix devices can be used. Point-in-time snap copies of thin devices can be created on VDEVs, which can subsequently be presented to a host. The VDEVs can also be updated and the changes on the VDEVs can be restored back to the source thin devices or to different thin devices.

Thin devices are classified as standard devices and can be added to Symmetrix device groups or put in device files.

A device group called “thinsnapdg” has been created that contains four thin devices along with four virtual devices:

```
# symdev list -sid 34 -tdev | grep c2t50060482D52FC
10F4 rdmp/c2t50060482D52FC* 09A:1 NA:NA TDEV Grp’d RW 4314
10F5 rdmp/c2t50060482D52FC* 09A:1 NA:NA TDEV Grp’d RW 4314
10F6 rdmp/c2t50060482D52FC* 09A:1 NA:NA TDEV Grp’d RW 4314
10F7 rdmp/c2t50060482D52FC* 09A:1 NA:NA TDEV Grp’d RW 4314
```

```
# symdg show thinsnapdg
Group Name: thinsnapdg
Group Type : REGULAR
Device Group in GNS : No
Valid : Yes
Symmetrix ID : 000190103334
Group Creation Time : Thu Feb 28 11:06:35 2008
Vendor ID : EMC Corp
Application ID : SYMCLI

Number of STD Devices in Group : 4
Number of Associated GK’s : 0
Number of Locally-associated BCV’s : 0
Number of Locally-associated VDEV’s : 4
Number of Locally-associated TGT’s : 0
Number of Remotely-associated VDEV’s(STD RDF) : 0
Number of Remotely-associated BCV’s (STD RDF) : 0
Number of Remotely-associated TGT’s(TGT RDF) : 0
Number of Remotely-associated BCV’s (BCV RDF) : 0
Number of Remotely-assoc’d BCV’s (RBCV RDF) : 0
Number of Remote-assoc’d VDEV’s(Hop-2 BCV) : 0
Number of Remote-assoc’d BCV’s (BCV RDF) : 0
Number of Remote-assoc’d VDEV’s(Hop-2 VDEV) : 0
Number of Remote-assoc’d TGT’s (Hop-2 TGT) : 0
```
Virtual Provisioning

Standard (STD) Devices (4):

<table>
<thead>
<tr>
<th>LdevName</th>
<th>PdevName</th>
<th>Sym</th>
<th>Dev</th>
<th>Att. Sts</th>
<th>Cap (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEV001</td>
<td>rdmp/c2t50060482D52FC9*</td>
<td>10F4</td>
<td>RW</td>
<td></td>
<td>4314</td>
</tr>
<tr>
<td>DEV002</td>
<td>rdmp/c2t50060482D52FC9*</td>
<td>10F5</td>
<td>RW</td>
<td></td>
<td>4314</td>
</tr>
<tr>
<td>DEV003</td>
<td>rdmp/c2t50060482D52FC9*</td>
<td>10F6</td>
<td>RW</td>
<td></td>
<td>4314</td>
</tr>
<tr>
<td>DEV004</td>
<td>rdmp/c2t50060482D52FC9*</td>
<td>10F7</td>
<td>RW</td>
<td></td>
<td>4314</td>
</tr>
</tbody>
</table>

VDEV Devices Locally-associated (4):

<table>
<thead>
<tr>
<th>LdevName</th>
<th>PdevName</th>
<th>Sym</th>
<th>Dev</th>
<th>Att. Sts</th>
<th>Cap (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDEV001</td>
<td>N/A</td>
<td>1130</td>
<td>NR</td>
<td></td>
<td>4314</td>
</tr>
<tr>
<td>VDEV002</td>
<td>N/A</td>
<td>1131</td>
<td>NR</td>
<td></td>
<td>4314</td>
</tr>
<tr>
<td>VDEV003</td>
<td>N/A</td>
<td>1132</td>
<td>NR</td>
<td></td>
<td>4314</td>
</tr>
<tr>
<td>VDEV004</td>
<td>N/A</td>
<td>1133</td>
<td>NR</td>
<td></td>
<td>4314</td>
</tr>
</tbody>
</table>

A save pool called “thindev_snap” has been created for the VDEVs in the “thinsnapdg” device group. Save pools for TimeFinder/Snap can be listed by using the –snap option with the symcfg list -pool command:

```
# symcfg list -pool -snap -sid 34
```

Symmetrix ID: 000190103334

<table>
<thead>
<tr>
<th>Pool Name</th>
<th>Ty</th>
<th>p Dev</th>
<th>Dev Emul</th>
<th>Dev Config</th>
<th>Total Tracks</th>
<th>Enabled Tracks</th>
<th>Used Tracks</th>
<th>Free Tracks</th>
<th>Full (%)</th>
<th>Sta</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFAULT_POOL</td>
<td>SN FBA</td>
<td>Unknown</td>
<td>2-Way Mir</td>
<td>5659605</td>
<td>5659605</td>
<td>0</td>
<td>5659605</td>
<td>0</td>
<td>Ena</td>
<td></td>
</tr>
<tr>
<td>DEFAULT_POOL</td>
<td>SN 3390</td>
<td>Unknown</td>
<td>2-Way Mir</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Dis</td>
<td></td>
</tr>
<tr>
<td>DEFAULT_POOL</td>
<td>SN 3380</td>
<td>Unknown</td>
<td>2-Way Mir</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Dis</td>
<td></td>
</tr>
<tr>
<td>DEFAULT_POOL</td>
<td>SN AS400</td>
<td>Unknown</td>
<td>2-Way Mir</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Dis</td>
<td></td>
</tr>
<tr>
<td>thindev_snap</td>
<td>SN FBA</td>
<td>2-Way Mir</td>
<td>276120</td>
<td>276120</td>
<td>0</td>
<td>276120</td>
<td>0</td>
<td>Ena</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td></td>
<td>5935725</td>
<td>5935725</td>
<td>0</td>
<td>5935725</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Legend for Pool Types:

- SN = Snap, RS = Rdfa DSE, TD = Thin Data

TimeFinder/Snap operations can be performed against the thin devices in the same way they can be performed against any regular device. Here a snap session is created, activated, and terminated:


# symsnap -g thinsnapdg create -svp thindev_snap -nop

'Create' operation execution is in progress for
device group 'thinsnapdg'. Please wait...

'Create' operation successfully executed for device group
'thinsnapdg'.

# symsnap -g thinsnapdg query

Device Group (DG) Name: thinsnapdg
DG's Type : REGULAR
DG's Symmetrix ID : 000190103334

<table>
<thead>
<tr>
<th>Source Device</th>
<th>Target Device</th>
<th>State</th>
<th>Copy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logical</td>
<td>Sym</td>
<td>Tracks</td>
<td>Logical</td>
</tr>
<tr>
<td>DEV001</td>
<td>10F4</td>
<td>69030</td>
<td>VDEV001</td>
</tr>
<tr>
<td>DEV002</td>
<td>10F5</td>
<td>69030</td>
<td>VDEV002</td>
</tr>
<tr>
<td>DEV003</td>
<td>10F6</td>
<td>69030</td>
<td>VDEV003</td>
</tr>
<tr>
<td>DEV004</td>
<td>10F7</td>
<td>69030</td>
<td>VDEV004</td>
</tr>
</tbody>
</table>

Total

Track(s) 276120
MB(s) 17257.5

Legend:
(G): X = The Target device is associated with this group.
. = The Target device is not associated with this group.

# symsnap -g thinsnapdg activate -nop

'Activate' operation execution is in progress for
device group 'thinsnapdg'. Please wait...

'Activate' operation successfully executed for device group
'thinsnapdg'.

# symsnap -g thinsnapdg query

Device Group (DG) Name: thinsnapdg
DG's Type : REGULAR
DG's Symmetrix ID : 000190103334

<table>
<thead>
<tr>
<th>Source Device</th>
<th>Target Device</th>
<th>State</th>
<th>Copy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logical</td>
<td>Sym</td>
<td>Tracks</td>
<td>Logical</td>
</tr>
<tr>
<td>DEV001</td>
<td>10F4</td>
<td>69030</td>
<td>VDEV001</td>
</tr>
<tr>
<td>Device</td>
<td>ID</td>
<td>Size</td>
<td>VDEV</td>
</tr>
<tr>
<td>--------</td>
<td>----</td>
<td>--------</td>
<td>------</td>
</tr>
<tr>
<td>DEV002</td>
<td>10F5</td>
<td>69030</td>
<td>VDEV002</td>
</tr>
<tr>
<td>DEV003</td>
<td>10F6</td>
<td>69030</td>
<td>VDEV003</td>
</tr>
<tr>
<td>DEV004</td>
<td>10F7</td>
<td>69030</td>
<td>VDEV004</td>
</tr>
</tbody>
</table>

Total  
Track(s): 276120  
MB(s): 17257.5  

Legend:

(G): X = The Target device is associated with this group,  
. = The Target device is not associated with this group.

# symsnap -g thinsnapdg terminate -nop

'Terminate' operation execution is in progress for  
device group 'thinsnapdg'. Please wait...

'Terminate' operation successfully executed for device group  
'thinsnapdg'.

# symsnap -g thinsnapdg query

The Source device and the Target device do not form a Copy session

Device group 'thinsnapdg' does not have any devices that are Snap source devices

**Local replication using TimeFinder/Clone**

TimeFinder/Clone can be used to create point-in-time copies of thin devices on other thin devices. Starting in Enginuity 5874, clone sessions can also be created between regular and thin devices with the thin device being the clone source, the clone target, or both.

Performing clone operations using thin devices is exactly the same as creating clones of regular devices. Thin clone targets can be presented to a host and accessed just like regular devices or clone source devices. Thin clone targets can also be updated and the changes on the targets can be restored back to the source thin devices, to different thin devices, or (in Enginuity 5874) to a regular device.

Thin devices are classified as standard devices and can be added to Symmetrix device groups or put in device files. When using device groups, clone operations can be performed against devices by specifying the clone source and target when the clone pairs are created or clone targets can instead be added as targets by specifying the -tgt option when they are added to the group. The -tgt option will also be used with the other commands:

```
# symdev list -sid 34 -tdev | grep c2t50060482D52FC
```

---

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

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

TimeFinder/Clone operations can now performed against the thin devices in the group. In this example, a clone session is created,
Virtual Provisioning

activated, and terminated:

```
# symclone -g thinclonedg create -differential -precopy -tgt -nop

'Create' operation execution is in progress for device group 'thinclonedg'. Please wait...

'Create' operation successfully executed for device group 'thinclonedg'.

# symclone -g thinclonedg query

Device Group (DG) Name: thinclonedg
DG's Type : REGULAR
DG's Symmetrix ID : 000190103334

<table>
<thead>
<tr>
<th>Source Device</th>
<th>Target Device</th>
<th>State</th>
<th>Copy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logical</td>
<td>Logical</td>
<td>Sym</td>
<td>Sym</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tracks</td>
<td>Tracks</td>
</tr>
<tr>
<td>DEV001</td>
<td>TGT001</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>DEV002</td>
<td>TGT002</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Legend:

(C): X = The background copy setting is active for this pair.
(G): X = The Target device is associated with this group.
(D): X = The Clone session is a differential copy session.
(P): X = The pre-copy operation has completed one cycle

Total
Track(s) 0 0
MB(s) 0.0 0.0

# symclone -g thinclonedg activate -tgt -nop

'Activate' operation execution is in progress for device group 'thinclonedg'. Please wait...

'Activate' operation successfully executed for device group 'thinclonedg'.

# symclone -g thinclonedg query

Device Group (DG) Name: thinclonedg
DG's Type : REGULAR
Virtual Provisioning

DG's Symmetrix ID : 000190103334

<table>
<thead>
<tr>
<th>Source Device</th>
<th>Target Device</th>
<th>State</th>
<th>Copy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protected</td>
<td>Modified</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Logical</td>
<td>Sym Tracks</td>
<td>Logical Sym Tracks</td>
<td>CGDP SRC &lt;=&gt; TGT (%)</td>
</tr>
<tr>
<td>DEV001 1160</td>
<td>0 TGT001 1162</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>DEV002 1161</td>
<td>0 TGT002 1163</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>Total</td>
<td>--------</td>
<td>-------</td>
<td></td>
</tr>
<tr>
<td>Track(s)</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>MB(s)</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Legend:

(C): X = The background copy setting is active for this pair.
      . = The background copy setting is not active for this pair.
(G): X = The Target device is associated with this group.
      . = The Target device is not associated with this group.
(D): X = The Clone session is a differential copy session.
      . = The Clone session is not a differential copy session.
(P): X = The pre-copy operation has completed one cycle
      . = The pre-copy operation has not completed one cycle

# symclone -g thinclonedg terminate -tgt -nop

'Terminate' operation execution is in progress for
device group 'thinclonedg'. Please wait...

'Terminate' operation successfully executed for device group
'thinclonedg'.

# symclone -g thinclonedg query

The Source device and the Target device do not form a Copy session

Device group 'thinclonedg' does not have any devices that are Clone source devices

Remote replication using SRDF

SRDF/Synchronous, SRDF/Asynchronous, or SRDF/Adaptive Copy can be used to perform replication of thin devices to thin devices on a remote Symmetrix array. Other than the requirement that both the source and target devices are thin, creating and managing thin SRDF devices is exactly the same as creating and managing SRDF using regular devices.

In this example, four bound thin devices in a local Symmetrix (devices 1164:1167 in Symm 3334) and four bound thin devices in a remote Symmetrix (0DF2:0DF5 in Symm 3232) are configured for Synchronous SRDF.
Virtual Provisioning

```bash
# symcfg list

```

<table>
<thead>
<tr>
<th>SymmID</th>
<th>Attachment</th>
<th>Model</th>
<th>Mcode</th>
<th>Cache</th>
<th>Num Phys</th>
<th>Num Symm</th>
</tr>
</thead>
<tbody>
<tr>
<td>000190103334</td>
<td>Local</td>
<td>DMX4-24</td>
<td>5773</td>
<td>65536</td>
<td>101</td>
<td>4526</td>
</tr>
<tr>
<td>000190103232</td>
<td>Remote</td>
<td>DMX4-24</td>
<td>5773</td>
<td>65536</td>
<td>00</td>
<td>7335</td>
</tr>
</tbody>
</table>

```

# sympd list -sid 34

Symmetrix ID: 000190103334

```

<table>
<thead>
<tr>
<th>Device Name</th>
<th>Directors</th>
<th>Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>--------------</td>
<td>-----------</td>
<td>--------</td>
</tr>
<tr>
<td>Sym  Physical</td>
<td>SA :P DA :IT</td>
<td>Config</td>
</tr>
</tbody>
</table>

...                  

0DF2 Not Visible  ?????:?  NA:NA  TDEV  N/Grp'd  RW  17263
0DF3 Not Visible  ?????:?  NA:NA  TDEV  N/Grp'd  RW  17263
0DF4 Not Visible  ?????:?  NA:NA  TDEV  N/Grp'd  RW  17263
0DF5 Not Visible  ?????:?  NA:NA  TDEV  N/Grp'd  RW  17263

```

# symdev list -sid 32 -tdev

Symmetrix ID: 000190103232

```

<table>
<thead>
<tr>
<th>Device Name</th>
<th>Directors</th>
<th>Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sym  Physical</td>
<td>SA :P DA :IT</td>
<td>Config</td>
</tr>
</tbody>
</table>

...                  

A Configuration Change operation is in progress. Please wait...

Establishing a configuration change session..............Established.
Processing symmetrix 000190103334
Performing Access checks.................................Allowed.
Checking Device Reservations..............................Allowed.
Submitting configuration changes.........................Submitted.
Locking devices...........................................Locked.
Validating configuration changes.........................Validated.
Initiating PREPARE of configuration changes..............Prepared.

If the devices are not already configured as dynamic SRDF devices, they need to be converted:

# symconfigure -sid 34 -cmd "set dev 1164:1167 attribute=dyn_rdf;" commit -nop
The configuration change session has successfully completed.

A Configuration Change operation is in progress. Please wait...

Establishing a configuration change session...............Established.
Processing symmetrix 000190103232
Performing Access checks................................Allowed.
Checking Device Reservations..............................Allowed.
Submitting configuration changes..........................Submitted
Locking devices...........................................Locked.
Validating configuration changes..........................Validated.
Initiating PREPARE of configuration changes...............Prepared.
Initiating COMMIT of configuration changes...............Queued.
COMMIT requesting required resources......................Obtained.
Step 003 of 027 steps.....................................Executing.
Step 017 of 027 steps.....................................Executing.
Step 021 of 027 steps.....................................Executing.
Step 022 of 027 steps.....................................Executing.
Step 025 of 027 steps.....................................Executing.
Local:  COMMIT............................................Done.
Terminating the configuration change session..............Done.

The configuration change session has successfully completed.

# symconfigure -sid 32 -cmd "set dev 0DF2:0DF5 attribute=dyn_rdf;" commit -nop

A Configuration Change operation is in progress. Please wait...

Establishing a configuration change session...............Established.
Processing symmetrix 000190103232
Performing Access checks................................Allowed.
Checking Device Reservations..............................Allowed.
Submitting configuration changes..........................Submitted
Locking devices...........................................Locked.
Validating configuration changes..........................Validated.
Initiating PREPARE of configuration changes...............Prepared.
Initiating COMMIT of configuration changes...............Queued.
COMMIT requesting required resources......................Obtained.
Step 003 of 027 steps.....................................Executing.
Step 017 of 027 steps.....................................Executing.
Step 021 of 027 steps.....................................Executing.
Step 022 of 027 steps.....................................Executing.
Step 025 of 027 steps.....................................Executing.
Local:  COMMIT............................................Done.
Terminating the configuration change session..............Done.

The configuration change session has successfully completed.

The RDF group can then be created and a second pair of RAs can be added:

# symrdf addgrp -label THIN_RDF -rdfg 6 -sid 34 -dir 7D -remote_rdfg 6 -remote_dir 7D -remote_sid 32 -nop

Successfully Added Dynamic RDF Group 'THIN_RDF' for Symm: 000190103334

# symrdf modifygrp -add -label THIN_RDF -sid 34 -dir 8D -remote_dir 8D -nop

Successfully Added Directors to Dynamic RDF Group
Group Label: 'THIN_RDF' for Symm: 000190103334

A device file with the local and remote thin devices can be used to create the RDF pairs:

# cat thin_rdf_devs
Virtual Provisioning

# Thin Devices for SRDF
#
# Symm 34 (Local)    Symm 32 (Remote)
#
1164               ODF2
1165               ODF3
1166               ODF4
1167               ODF5

# symrdf -f thin_rdf_devs createpair -type R1 -sid 34 -rdfg 6 -establish -nop

An RDF 'Create Pair' operation execution is in progress for device
file 'thin_rdf_devs'. Please wait...

Create RDF Pair in (3334,006).............................Started.
Create RDF Pair in (3334,006).............................Done.
Mark target device(s) in (3334,006) to refresh from source......Started.
Mark target device(s) in (3334,006) to refresh from source......Done.
Mark target device(s) in (3334,006) for full copy from source....Started.
Mark target device(s) in (3334,006) for full copy from source....Done.
Merge track tables between source and target in (3334,006).......Started.
Merge track tables between source and target in (3334,006).......Merged.
Resume RDF link(s) for device(s) in (3334,006)...................Started.
Resume RDF link(s) for device(s) in (3334,006)...................Done.

The RDF 'Create Pair' operation successfully executed for device
file 'thin_rdf_devs'.

# symrdf -f thin_rdf_devs query -rdfg 6 -sid 34

Symmetrix ID                       : 000190103334    (Microcode Version: 5773)
Remote Symmetrix ID                : 000190103232    (Microcode Version: 5773)
RDF (RA) Group Number              :   6 (05)

<table>
<thead>
<tr>
<th>Source (R1) View</th>
<th>Target (R2) View</th>
<th>MODES</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST</td>
<td>LI</td>
<td>ST</td>
</tr>
<tr>
<td>Standard</td>
<td>A</td>
<td>N</td>
</tr>
<tr>
<td>Logical</td>
<td>T R1 Inv R2 Inv K</td>
<td>T R1 Inv R2 Inv MDA STATE</td>
</tr>
<tr>
<td>Device Dev E Tracks S Dev E Tracks Tracks MDA STATE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

N/A 1164 RW 0 160030 RW ODF2 WD 0 10 S.. SyncInProg
N/A 1165 RW 0 151882 RW ODF3 WD 0 10 S.. SyncInProg
N/A 1166 RW 0 141634 RW ODF4 WD 0 10 S.. SyncInProg
N/A 1167 RW 0 128230 RW ODF5 WD 0 10 S.. SyncInProg

Total Track(s) 0 S81776 0 40 MB(s) 0.0 36361.0 0.0 2.5

Legend for MODES:
M(ode of Operation): A = Async, S = Sync, E = Semi-sync, C = Adaptive Copy
D(omino) : X = Enabled, . = Disabled
A(daptive Copy) : D = Disk Mode, W = WP Mode, . = ACp off
After the devices are synchronized, they can be used for all applicable SRDF operations:

```
# symrdf -f thin_rdf_devs query -sid 34 -rdfg 6
```

<table>
<thead>
<tr>
<th>Symmetrix ID</th>
<th>Remote Symmetrix ID</th>
<th>RDF (RA) Group Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>: 000190103334</td>
<td>: 000190103232</td>
<td>: 6 (05)</td>
</tr>
</tbody>
</table>

Symmetrix ID                       : 000190103334    (Microcode Version: 5773)  
Remote Symmetrix ID                : 000190103232    (Microcode Version: 5773)  
RDF (RA) Group Number              :   6 (05)  

### Source (R1) View  
<table>
<thead>
<tr>
<th>ST</th>
<th>LI</th>
<th>ST</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard</td>
<td>A</td>
<td>N</td>
</tr>
<tr>
<td>Logical</td>
<td>T</td>
<td>R1 Inv</td>
</tr>
<tr>
<td>Device Dev</td>
<td>E</td>
<td>Tracks</td>
</tr>
</tbody>
</table>

| N/A | 1164 RW | 0 | 0 | RW | ODF2 | WD | 0 | 0 | S.. | Synchronized |  
| N/A | 1165 RW | 0 | 0 | RW | ODF3 | WD | 0 | 0 | S.. | Synchronized |  
| N/A | 1166 RW | 0 | 0 | RW | ODF4 | WD | 0 | 0 | S.. | Synchronized |  
| N/A | 1167 RW | 0 | 0 | RW | ODF5 | WD | 0 | 0 | S.. | Synchronized |  

Total
| Track(s)  | 0 | 0 | 0 | 0 |  
| MB(s)     | 0.0 | 0.0 | 0.0 | 0.0 |  

Legend for MODES:

- **M(ode of Operation)**: A = Async, S = Sync, E = Semi-sync, C = Adaptive Copy  
- **D(omino)** : X = Enabled, . = Disabled  
- **A(daptive Copy)** : D = Disk Mode, W = WP Mode, . = ACp off  

```
# symrdf -f thin_rdf_devs split -sid 34 -rdfg 6 -nop  
```

An RDF 'Split' operation execution is in progress for device file 'thin_rdf_devs'. Please wait...  
Suspense RDF link(s) for device(s) in (3334,006)..................Done.  
Read/Write Enable device(s) in (3334,006) on RA at target (R2)...Done.  
The RDF 'Split' operation successfully executed for device file 'thin_rdf_devs'.

```
# symrdf -f thin_rdf_devs query -sid 34 -rdfg 6  
```

<table>
<thead>
<tr>
<th>Symmetrix ID</th>
<th>Remote Symmetrix ID</th>
<th>RDF (RA) Group Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>: 000190103334</td>
<td>: 000190103232</td>
<td>: 6 (05)</td>
</tr>
</tbody>
</table>

| Source (R1) View  | Target (R2) View  | MODES  
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ST</td>
<td>LI</td>
<td>ST</td>
</tr>
<tr>
<td>Standard</td>
<td>A</td>
<td>N</td>
</tr>
<tr>
<td>Logical</td>
<td>T</td>
<td>R1 Inv</td>
</tr>
<tr>
<td>Device Dev</td>
<td>E</td>
<td>Tracks</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th></th>
<th>1164 RW</th>
<th>0</th>
<th>0</th>
<th>NR</th>
<th>0DF2 RW</th>
<th>0</th>
<th>0</th>
<th>S..</th>
<th>Split</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1165 RW</td>
<td>0</td>
<td>0</td>
<td>NR</td>
<td>0DF3 RW</td>
<td>0</td>
<td>0</td>
<td>S..</td>
<td>Split</td>
</tr>
<tr>
<td></td>
<td>1166 RW</td>
<td>0</td>
<td>0</td>
<td>NR</td>
<td>0DF4 RW</td>
<td>0</td>
<td>0</td>
<td>S..</td>
<td>Split</td>
</tr>
<tr>
<td></td>
<td>1167 RW</td>
<td>0</td>
<td>0</td>
<td>NR</td>
<td>0DF5 RW</td>
<td>0</td>
<td>0</td>
<td>S..</td>
<td>Split</td>
</tr>
</tbody>
</table>

Total

| Track(s) | 0   | 0   | 0   | 0   |
| MB(s)     | 0.0 | 0.0 | 0.0 | 0.0 |

Legend for MODES:

M(ode of Operation): A = Async, S = Sync, E = Semi-sync, C = Adaptive Copy
D(omino): X = Enabled, . = Disabled
A(daptive Copy): D = Disk Mode, W = WP Mode, . = ACp off

Remote replication using Open Replicator

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, zeroes will be sent for any regions of the thin device that have not been allocated, or that have been allocated but have not been 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, then a portion of the target thin device, equal in size to the reported size of the source volume, will become allocated. If the target thin device resides on a Symmetrix VMAX, space reclamation can then be used to return all zero extents to the thin pool.

Automated pool rebalancing

Starting in Enginuity 5874 and Solutions Enabler 7.1, automated pool rebalancing allows the user to run a balancing operation that will redistribute data across the enabled data devices in the thin pool. Because the thin extents are allocated from the thin pool in a round-robin fashion, the rebalancing mechanism will be used primarily when adding data devices to increase thin pool capacity.

The balancing algorithm will measure the minimum, maximum, and mean used capacity values of the data devices in the thin pool. The Symmetrix will then move thin device extents from the data devices with the highest used capacity to those with the lowest until the pool is balanced.
In this example, the oraprod pool is 71 percent:

```
# symcfg show -pool oraprod -thin -sid 54
Symmetrix ID: 000192601254
Symmetrix ID                  : 000192601254
Pool Name                     : oraprod
Pool Type                     : Thin
Dev Emulation                 : FBA
Dev Configuration             : 2-Way Mir
Pool State                    : Enabled
# of Devices in Pool          :   4
# of Enabled Devices in Pool  :   4

Enabled Devices(4):
{
------------------------------------------------------
Sym         Total      Alloc       Free Full  Device
Dev        Tracks     Tracks     Tracks  (%)  State
------------------------------------------------------
0F18        69024      49572      19452   71  Enabled
0F19        69024      49476      19548   71  Enabled
0F1A        69024      49584      19440   71  Enabled
0F1B        69024      49596      19428   71  Enabled
-------------------------------
Tracks     276096     198228      77868   71
```

In order to expand the pool, two new data devices are added and enabled:

```
# symconfigure -sid 54 -cmd "create dev count=2, config=2-Way-Mir, attribute=datadev, emulation=FBA, size=4602, in pool oraprod, member_state=ENABLE;" commit –nop
```

The pool now has more capacity, but the spread of extents across the data devices is not balanced.

```
# symcfg show -pool oraprod -thin -sid 54
Symmetrix ID: 000192601254
Symmetrix ID                  : 000192601254
Pool Name                     : oraprod
Pool Type                     : Thin
Dev Emulation                 : FBA
Dev Configuration             : 2-Way Mir
Pool State                    : Enabled
# of Devices in Pool          :   6
# of Enabled Devices in Pool  :   6

Enabled Devices(6):
{

--------------------------------------------------------------
Sym        Total      Used      Free  Full  Device    Session
Dev       Tracks    Tracks    Tracks   (%)  State     Status
--------------------------------------------------------------
0F18       69024      49572      19452   71  Enabled     
0F19       69024      49476      19548   71  Enabled     
0F1A       69024      49584      19440   71  Enabled     
0F1B       69024      49596      19428   71  Enabled     
--------------------------
Tracks     276096     198228      77868   71
```
If the pool is not rebalanced, the original four devices will fill and data will then stripe across only the two newly added devices. In a real production environment with a large, wide striped pool, this could cause severe performance degradation. Running the rebalancing operation will solve the problem:

```
# symconfigure -sid 54 -cmd "start balancing on pool oraprod;"
commit –nop
```

After starting the balancing operation, the pool state changes from “Enabled” to “Balancing”:

```
# symcfg show -pool oraprod -thin -sid 54
```

<table>
<thead>
<tr>
<th>Symmetrix ID: 000192601254</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pool Name                   : oraprod</td>
</tr>
<tr>
<td>Pool Type                   : Thin</td>
</tr>
<tr>
<td>Dev Emulation               : FBA</td>
</tr>
<tr>
<td>Dev Configuration           : 2-Way Mir</td>
</tr>
<tr>
<td>Pool State                  : Balancing</td>
</tr>
<tr>
<td># of Devices in Pool        :  6</td>
</tr>
<tr>
<td># of Enabled Devices in Pool :  6</td>
</tr>
</tbody>
</table>

Enabled Devices(6):

<table>
<thead>
<tr>
<th>Sym</th>
<th>Total</th>
<th>Alloc</th>
<th>Free</th>
<th>Full (%)</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>0F18</td>
<td>69024</td>
<td>49584</td>
<td>19440</td>
<td>71</td>
<td>Enabled</td>
</tr>
<tr>
<td>0F19</td>
<td>69024</td>
<td>49476</td>
<td>19548</td>
<td>71</td>
<td>Enabled</td>
</tr>
<tr>
<td>0F1A</td>
<td>69024</td>
<td>49584</td>
<td>19440</td>
<td>71</td>
<td>Enabled</td>
</tr>
<tr>
<td>0F1B</td>
<td>69024</td>
<td>49596</td>
<td>19428</td>
<td>71</td>
<td>Enabled</td>
</tr>
<tr>
<td>0F40</td>
<td>69024</td>
<td>0</td>
<td>69024</td>
<td>0</td>
<td>Enabled</td>
</tr>
<tr>
<td>0F41</td>
<td>69024</td>
<td>0</td>
<td>69024</td>
<td>0</td>
<td>Enabled</td>
</tr>
</tbody>
</table>

```
Tracks 414144 198240 215904 47
```
Enginuity will give the highest priority to front-end I/O operations. However, rebalancing can be stopped for any reason, such as an upcoming period of very high host I/O or an operation that will cause a large amount of internal copy tasks.

```
# symconfigure -sid 54 -cmd "stop balancing on pool=oraprod type=thin;" commit -nop
```

To resume the balancing operation, the balancing operation can simply be run again:

```
# symconfigure -sid 54 -cmd "start balancing on pool=oraprod type=thin;" commit -nop
```

After the operation has completed, the pool is balanced:

```
# symcfg show -pool oraprod -thin -sid 54
```

```
Symmetrix ID: 000192601254
Symmetrix ID                  : 000192601254
Pool Name                     : oraprod
Pool Type                     : Thin
Dev Emulation                 : FBA
Dev Configuration             : 2-Way Mir
Pool State                    : Enabled
# of Devices in Pool          :   6
# of Enabled Devices in Pool  :   6

Enabled Devices(6):
{

-----------------------------------------
Sym        Total      Alloc       Free Full  Device
Dev        Tracks     Tracks     Tracks  (%)  State
-----------------------------------------
0F18        69024      32844      36180   47  Enabled
0F19        69024      32736      36288   47  Enabled
0F1A        69024      34224      34800   49  Enabled
0F1B        69024      33180      35844   48  Enabled
0F40        69024      32472      36552   47  Enabled
0F41        69024      32784      36240   47  Enabled
-----------------------------------------
Tracks     414144     198240     215904   47

Note: Rebalancing is designed to bring the utilized capacity of all devices within 10 percent tolerance. Testing has found that the natural growth of thin devices bound to the pool can lead to a variance of up to 10 percent. This variation is caused by the ability of the microcode to skip extent allocation on the next data dev in the pool. If that device's DA is busy that DA can pass on the allocation request to the DA that owns the next data dev in the pool so
extent allocation does not have a large effect on host I/O.

**Space reclamation**

Solutions Enabler 7.1 and Enginuity 5874 provide the ability to deallocate thin device extents that are found to contain all zeros.

This feature is an extension of the existing Virtual Provisioning space deallocation mechanism. Running the space reclamation command will spawn a DA background task that will examine the allocated extents on specified thin device. For each allocated extent, all 12 tracks will be brought into cache and scanned to see if they contain all zero data. If the entire extent contains all zero data, the extent will be deallocated and added back into the pool, making it available for a new extent allocation operation.

Space reclamation is designed to be used primarily following a migration from a regular device to a thin device. In this example, clone sessions are used to migrate data to a group of thin devices, which will be subsequently reclaimed.

Prior to the clone operation, the thin devices, which will be the clone targets, have only the initial 12 track thin device extents allocated:

```
# symcfg show -pool betatest -thin -detail -all -sid 54
```

```
Symmetrix ID: 000192601254

Symmetrix ID                  : 000192601254
Pool Name                     : betatest
Pool Type                     : Thin
Dev Emulation                 : FBA
Dev Configuration             : 2-Way Mir
Pool State                    : Enabled
# of Devices in Pool          :   8
# of Enabled Devices in Pool  :   8
Max. Subscription Percent     : None

Enabled Devices(8):
{
  ----------------------------
  Sym  Total Tracks  Alloc Tracks  Free Full Tracks (%) State
  ----------------------------
  0F18   69024       12       69012    0  Enabled
  0F19   69024       12       69012    0  Enabled
  0F1A   69024       12       69000    0  Enabled
  0F1B   69024       12       68976    0  Enabled
  0F40   69024       12       69024    0  Enabled
  0F41   69024       12       69024    0  Enabled
  0F42   69024       12       69024    0  Enabled
  0F43   69024       12       69024    0  Enabled
  ----------------------------
  Tracks  552192  96  552096     0
}
```
Thin Devices(8):

```
Pool          Pool          Pool
Sym          Total Subs    Allocated Tracks (%)  Written Tracks (%)  Status
Dev           Tracks (%)    Tracks (%)     Tracks (%)     tracks (%)    
---------------------------------------------------------------------
0F1C         69030 13       12 0 0 0 0 Bound
0F1D         69030 13       12 0 0 0 0 Bound
0F1E         69030 13       12 0 0 0 0 Bound
0F1F         69030 13       12 0 0 0 0 Bound
0F20         69030 13       12 0 0 0 0 Bound
0F21         69030 13       12 0 0 0 0 Bound
0F22         69030 13       12 0 0 0 0 Bound
0F23         69030 13       12 0 0 0 0 Bound
---------------------------------------------------------------------
Tracks 552240 100 96 0 0 0
```

# symclone -g betatestdg create -precopy -diff -tgt -nop

'Create' operation execution is in progress for device group 'betatestdg'. Please wait...

'Create' operation successfully executed for device group 'betatestdg'.

# symclone -g betatestdg activate -tgt -nop

'Activate' operation execution is in progress for device group 'betatestdg'. Please wait...

'Activate' operation successfully executed for device group 'betatestdg'.

# symclone -g betatestdg query

Device Group (DG) Name: betatestdg
DG's Type             : REGULAR
DG's Symmetrix ID     : 000192601254

```
Source Device                   Target Device            State
--------------------------------- ---------------------------- ----------
Logical Sym  Tracks    Tracks   Logical Sym  Tracks   Tracks   CGDP SRC <=>
TGT (%)               TGT (%)
--------------------------------- ---------------------------- ----------
DEV001    0F24         0        0 TGT001    0F1C        0 XXX. Copied
100       DEV002    0F25         0        0 TGT002    0F1D        0 XXX. Copied
100       DEV003    0F26         0        0 TGT003    0F1E        0 XXX. Copied
100       DEV004    0F27         0        0 TGT004    0F1F        0 XXX. Copied
100       DEV005    0F28         0        0 TGT005    0F20        0 XXX. Copied
100       DEV006    0F29         0        0 TGT006    0F21        0 XXX. Copied
```
### Virtual Provisioning

The thin target devices now have extents that were allocated during the clone copy:

```
# symcfg show -pool betatest -thin -detail -sid 54
```

```
Symmetrix ID: 000192601254
Symmetrix ID                  : 000192601254
Pool Name                     : betatest
Pool Type                     : Thin
Dev Emulation                 : FBA
Dev Configuration             : 2-Way Mir
Pool State                    : Enabled
# of Devices in Pool          : 8
# of Enabled Devices in Pool  : 8
Max. Subscription Percent     : None
```

**Enabled Devices(8):**

<table>
<thead>
<tr>
<th>Sym</th>
<th>Total Tracks</th>
<th>Alloc Tracks</th>
<th>Free Tracks</th>
<th>Full (%)</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>0F18</td>
<td>69024</td>
<td>44172</td>
<td>24852</td>
<td>63</td>
<td>Enabled</td>
</tr>
<tr>
<td>0F19</td>
<td>69024</td>
<td>42540</td>
<td>26484</td>
<td>61</td>
<td>Enabled</td>
</tr>
<tr>
<td>0F1A</td>
<td>69024</td>
<td>45264</td>
<td>23760</td>
<td>65</td>
<td>Enabled</td>
</tr>
<tr>
<td>0F1B</td>
<td>69024</td>
<td>44832</td>
<td>24192</td>
<td>64</td>
<td>Enabled</td>
</tr>
<tr>
<td>0F40</td>
<td>69024</td>
<td>42852</td>
<td>26172</td>
<td>62</td>
<td>Enabled</td>
</tr>
<tr>
<td>0F41</td>
<td>69024</td>
<td>43308</td>
<td>25716</td>
<td>62</td>
<td>Enabled</td>
</tr>
<tr>
<td>0F42</td>
<td>69024</td>
<td>43536</td>
<td>25488</td>
<td>63</td>
<td>Enabled</td>
</tr>
<tr>
<td>0F43</td>
<td>69024</td>
<td>42492</td>
<td>26532</td>
<td>61</td>
<td>Enabled</td>
</tr>
</tbody>
</table>

```
Tracks 552192 348996 203196 63
```

**Thin Devices(8):**

```
```

---

**Pool Summary**

<table>
<thead>
<tr>
<th>Sym</th>
<th>Total Tracks</th>
<th>Subs (%)</th>
<th>Allocated Tracks (%)</th>
<th>Written Tracks (%)</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>0F1C</td>
<td>69030</td>
<td>13</td>
<td>53736</td>
<td>10</td>
<td>52941</td>
</tr>
<tr>
<td>0F1D</td>
<td>69030</td>
<td>13</td>
<td>41880</td>
<td>8</td>
<td>40721</td>
</tr>
<tr>
<td>0F1E</td>
<td>69030</td>
<td>13</td>
<td>45144</td>
<td>8</td>
<td>44013</td>
</tr>
<tr>
<td>0F1F</td>
<td>69030</td>
<td>13</td>
<td>37356</td>
<td>7</td>
<td>36184</td>
</tr>
<tr>
<td>0F20</td>
<td>69030</td>
<td>13</td>
<td>40284</td>
<td>7</td>
<td>39132</td>
</tr>
<tr>
<td>0F21</td>
<td>69030</td>
<td>13</td>
<td>36072</td>
<td>7</td>
<td>34903</td>
</tr>
<tr>
<td>0F22</td>
<td>69030</td>
<td>13</td>
<td>45684</td>
<td>8</td>
<td>44541</td>
</tr>
<tr>
<td>0F23</td>
<td>69030</td>
<td>13</td>
<td>48840</td>
<td>9</td>
<td>47757</td>
</tr>
</tbody>
</table>
Tracks 552240 100 348996 63 340192 62

To return the all zero extents to the thin pool, a free tdev operation can be run against the thin devices to begin the reclaim:

```bash
# symconfigure -sid 54 -cmd "free tdev F1C:F23 start_cyl=0 end_cyl=last_cyl type=reclaim;" commit -nop
```

A Configuration Change operation is in progress. Please wait...

- Establishing a configuration change session..................Established.
- Processing symmetrix 000192601254
- Performing Access checks..................................Allowed.
- Checking Device Reservations..............................Allowed.
- Locking devices...........................................Locked.
- Committing configuration changes.........................Started.
- Committing configuration changes.........................Committed.
- Terminating the configuration change session..............Done.

The configuration change session has successfully completed.

The thin devices now have a status of “Reclaiming”:

```bash
# symcfg show -pool betatest -thin -detail -sid 54
```

Symmetrix ID: 000192601254

<table>
<thead>
<tr>
<th>Symmetrix ID</th>
<th>000192601254</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pool Name</td>
<td>betatest</td>
</tr>
<tr>
<td>Pool Type</td>
<td>Thin</td>
</tr>
<tr>
<td>Dev Emulation</td>
<td>FBA</td>
</tr>
<tr>
<td>Dev Configuration</td>
<td>2-Way Mir</td>
</tr>
<tr>
<td>Pool State</td>
<td>Enabled</td>
</tr>
<tr>
<td># of Devices in Pool</td>
<td>8</td>
</tr>
<tr>
<td># of Enabled Devices in Pool</td>
<td>8</td>
</tr>
<tr>
<td>Max. Subscription Percent</td>
<td>None</td>
</tr>
</tbody>
</table>

Enabled Devices(8):

<table>
<thead>
<tr>
<th>Sym</th>
<th>Total Tracks</th>
<th>Alloc Tracks</th>
<th>Free Full Tracks</th>
<th>Device State</th>
</tr>
</thead>
<tbody>
<tr>
<td>0F18</td>
<td>69024</td>
<td>44172</td>
<td>24852</td>
<td>63 Enabled</td>
</tr>
<tr>
<td>0F19</td>
<td>69024</td>
<td>42540</td>
<td>26484</td>
<td>61 Enabled</td>
</tr>
<tr>
<td>0F1A</td>
<td>69024</td>
<td>45264</td>
<td>23760</td>
<td>65 Enabled</td>
</tr>
<tr>
<td>0F1B</td>
<td>69024</td>
<td>44832</td>
<td>24192</td>
<td>64 Enabled</td>
</tr>
<tr>
<td>0F40</td>
<td>69024</td>
<td>42852</td>
<td>26172</td>
<td>62 Enabled</td>
</tr>
<tr>
<td>0F41</td>
<td>69024</td>
<td>43508</td>
<td>25716</td>
<td>62 Enabled</td>
</tr>
<tr>
<td>0F42</td>
<td>69024</td>
<td>43536</td>
<td>25488</td>
<td>63 Enabled</td>
</tr>
<tr>
<td>0F43</td>
<td>69024</td>
<td>42492</td>
<td>26532</td>
<td>61 Enabled</td>
</tr>
</tbody>
</table>

Tracks 552192 348996 203196 63

Thin Devices(8):

<table>
<thead>
<tr>
<th>Sym</th>
<th>Total</th>
<th>Subs</th>
<th>Allocated</th>
<th>Written</th>
</tr>
</thead>
</table>
### Virtual Provisioning

<table>
<thead>
<tr>
<th>Dev</th>
<th>Tracks (%)</th>
<th>Tracks (%)</th>
<th>Tracks (%)</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>0F1C</td>
<td>69030</td>
<td>13</td>
<td>51736</td>
<td>10 Reclaiming</td>
</tr>
<tr>
<td>0F1D</td>
<td>69030</td>
<td>13</td>
<td>41880</td>
<td>8 Reclaiming</td>
</tr>
<tr>
<td>0F1E</td>
<td>69030</td>
<td>13</td>
<td>45144</td>
<td>8 Reclaiming</td>
</tr>
<tr>
<td>0F1F</td>
<td>69030</td>
<td>13</td>
<td>37356</td>
<td>7 Reclaiming</td>
</tr>
<tr>
<td>0F20</td>
<td>69030</td>
<td>13</td>
<td>40284</td>
<td>7 Reclaiming</td>
</tr>
<tr>
<td>0F21</td>
<td>69030</td>
<td>13</td>
<td>36072</td>
<td>7 Reclaiming</td>
</tr>
<tr>
<td>0F22</td>
<td>69030</td>
<td>13</td>
<td>45684</td>
<td>8 Reclaiming</td>
</tr>
<tr>
<td>0F23</td>
<td>69030</td>
<td>13</td>
<td>48840</td>
<td>9 Reclaiming</td>
</tr>
</tbody>
</table>

As the reclaim operation continues, devices that have completed return to the “Bound” status:

```
# symcfg show -pool betatest -thin -detail -sid 54
```

Symmetrix ID: 000192601254

Symmetrix ID                  : 000192601254
Pool Name                     : betatest
Pool Type                     : Thin
Dev Emulation                 : FBA
Dev Configuration             : 2-Way Mir
Pool State                    : Enabled
# of Devices in Pool          : 8
# of Enabled Devices in Pool  : 8
Max. Subscription Percent     : None

Enabled Devices(8):

```perl

Enabled Devices(8):
{
    Sym          Total Tracks Alloc Tracks Free Full Device
    0F18         69024 23688 45336 34  Enabled
    0F19         69024 22104 46920 32  Enabled
    0F1A         69024 24288 44736 35  Enabled
    0F1B         69024 21812 45192 34  Enabled
    0F40         69024 22560 46464 32  Enabled
    0F41         69024 22512 46512 32  Enabled
    0F42         69024 22296 46728 32  Enabled
    0F43         69024 22044 46980 31  Enabled
}

Thin Devices(8):
{
    Sym          Total Tracks Subs Alloc Tracks Allocated Tracks (%) State
    0F1C         69030 13 39456 7 37635 3 Reclaiming
    0F1D         69030 13 23220 4 21573 4 Reclaiming
    0F1E         69030 13 20568 4 18755 3 Reclaiming
    0F1F         69030 13 12540 2 10918 2 Bound
```
After the reclaim operation has completed on all devices, the all zero space has been returned to the thin pool:

```bash
# symcfg show -pool betatest -thin -detail -sid 54
```

```
Symmetrix ID: 000192601254
Symmetrix ID                  : 000192601254
Pool Name                     : betatest
Pool Type                     : Thin
Dev Emulation                 : FBA
Dev Configuration             : 2-Way Mir
Pool State                    : Enabled
# of Devices in Pool          : 8
# of Enabled Devices in Pool  : 8
Max. Subscription Percent     : None

Enabled Devices(8):

```

```

Thin Devices(8):

```

```
The allocated tracks in the thin pool have been reduced from 348996 or 63 percent of the pool to 164640 or 30 percent, indicating that over half of the allocated space in the pool was either all zero or never written by a host.

**Virtual Provisioning with Symmetrix Management Console**

The following are examples of how to configure, update, and monitor a Symmetrix Virtual Provisioning environment using Symmetrix Management Console (SMC).

The examples used in this technical note have been created for illustrative purposes and do not represent a Virtual Provisioning environment configured for production workloads.

**Configuring and viewing data devices and pools**

Data devices are created in the same way that all Symmetrix devices are created. In this example, SMC is used to create devices with the datadev attribute. All devices with this attribute are used exclusively for populating thin pools:

To create data devices, select the Data Device tab and enter or select the desired properties. Data devices can be created before or after the...
creation of a pool to which they will be added. In this example, three data devices are created but not added to a pool. Because they are not being assigned to a pool, they will be left in a disabled state.

Add the task to the configuration session list and commit the changes. Thin pools can be created without adding data devices so that they can be populated at a later time. In this example, a pool called “HR” is created but no devices are added at pool creation time:
Virtual Provisioning

Best Practices for Fast, Simple Capacity Allocation with Symmetrix Virtual Provisioning
After the pool is created, data devices can be added to the pool:

Devices can exist in a thin pool and be in a disabled state, but disabled devices are not available for extent allocation. In order to enable the device, the Enable New Pool Member box must be selected:
Virtual Provisioning

Adding additional data devices to a pool

Additional data devices can be added to a pool anytime when more space in the pool is required. Because thin device extents are taken from all enabled devices in a pool on which free space exists, care must be taken to ensure that performance is not compromised by adding too few data devices. The Virtual Provisioning best practices documentation provides more information.

In this example, three more data devices are created, added to the HR thin pool, and enabled in a single operation:

![Device Configuration - Create Device](image)

After adding the task and committing it, the HR Pool has six enabled devices:

Best Practices for Fast, Simple Capacity Allocation with Symmetrix Virtual Provisioning
New writes to the thin devices will now be spread equally across all six enabled data devices.

Details about the pools can be viewed in both the top and bottom panes of the Properties view. Selecting the Data Devices tab displays details about the data devices in the HR pool:
Configuring, binding, and viewing thin devices

Data devices are created by selecting the Thin Device tab and entering or selecting the desired properties. Thin devices can be created before or after the creation of a pool to which they will be bound. In this example, three thin devices are created but not added to a pool:

The thin devices have been created and can then be bound to a thin pool:
Virtual Provisioning

Properties - 0001B0102000 Local/Thin Pools (4)

<table>
<thead>
<tr>
<th>Name</th>
<th>Emulation</th>
<th>Raw Config</th>
<th>Total (MB)</th>
<th>Enabled (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gen&amp;vp</td>
<td>FBA</td>
<td>RAID-5 (3 + 1)</td>
<td>8628</td>
<td>0</td>
</tr>
<tr>
<td>&amp;vp2</td>
<td>FBA</td>
<td>RAID-5</td>
<td>25884</td>
<td>25884</td>
</tr>
<tr>
<td>&amp;vp3rd</td>
<td>FBA</td>
<td>RAID-5</td>
<td>12942</td>
<td>12942</td>
</tr>
</tbody>
</table>

Symmetric ID: 0001B0102000
Pool Name: &vp2
Pool Type: TP Data
Raw Emulation: FBA

Enable Device
Disable Device
Create Device Pool...
Add/Remove Pool Members...
Enable All Pool Members
Disable All Pool Members
Delete Device Pool...
Expand Device Pool...

Best Practices for Fast, Simple Capacity Allocation with Symmetrix Virtual Provisioning
Because thin devices can be created with no associated physical storage, thin devices that are not bound to a thin pool will remain Not Ready.

Details about the thin devices can be viewed in the bottom pane of the Properties view. Selecting the Thin Devices tab displays details about the thin devices in the HR pool:

Preallocating pool space for thin devices

Contiguous extents on data devices can be preallocated when the thin devices are bound to a thin pool. Users may wish to preallocate space to certain thin volumes bound to a pool, or may choose to preallocate all of the thin volumes in their entirety.

Each of the data devices in the HR pool is 4314 MB in size.

The amount of preallocation can be specified in cylinders. Here, thin devices are created, bound to the HR pool, and fully preallocated.
The HR pool now has six devices bound to it with three devices (DDC, DDD, and DDE) that are fully preallocated. The preallocated space (207108 tracks, in total) is spread across all of the data devices and comprises 33% percent of the thin pool.
Allocating pool space for thin devices

Space can also be allocated for thin devices that are already bound to a pool. This might be required if it is determined that allocation is warranted for a thin device at some point after the device has been bound to a pool.

The PRODDB pool was created with four data devices and four thin devices, two of which were fully preallocated (DD7:DD8) and the other two which were not:
The user has the choice to allocate the entire device or to choose a starting or ending cylinder number. In this operation, space is allocated for about half of the device (from cylinder 0 to cylinder 2302):

```

<table>
<thead>
<tr>
<th>Name</th>
<th>Emulation</th>
<th>Unit Config</th>
<th>Total (Ml)</th>
<th>Enabled (Ml)</th>
<th>Alloc (Ml)</th>
<th>Alloc %</th>
<th>Subs %</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000_yy</td>
<td>FBA</td>
<td>RAD-5 (3 + 1)</td>
<td>8628</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ETA</td>
<td>FBA</td>
<td>2:Way Mir</td>
<td>4547</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PRODDB</td>
<td>FBA</td>
<td>2:Way Mir</td>
<td>17256</td>
<td>17256</td>
<td>8831</td>
<td>50</td>
<td>10</td>
</tr>
<tr>
<td>SALES</td>
<td>FBA</td>
<td>2:Way Mir</td>
<td>21570</td>
<td>12392</td>
<td>8885</td>
<td>41</td>
<td>16</td>
</tr>
<tr>
<td>VP2</td>
<td>FBA</td>
<td>RAD-5 (3 + 1)</td>
<td>4571</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

The user has the choice to allocate the entire device or to choose a starting or ending cylinder number. In this operation, space is allocated for about half of the device (from cylinder 0 to cylinder 2302):

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

The result is now displayed. Device DDD has 34560 out of 69030 tracks allocated:

<table>
<thead>
<tr>
<th>Device</th>
<th>Pool</th>
<th>Total Triks</th>
<th>Subscribed %</th>
<th>Allocated Triks</th>
<th>Allocated %</th>
<th>Triks WR</th>
<th>Written %</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDD7</td>
<td>PRODDB</td>
<td>60030</td>
<td>25</td>
<td>60030</td>
<td>25</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>DDD8</td>
<td>PRODDB</td>
<td>60030</td>
<td>25</td>
<td>60030</td>
<td>25</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>DDDC</td>
<td>PRODDB</td>
<td>60030</td>
<td>25</td>
<td>7428</td>
<td>3</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>DDDD</td>
<td>PRODDB</td>
<td>64030</td>
<td>25</td>
<td>12</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Best Practices for Fast, Simple Capacity Allocation with Symmetrix Virtual Provisioning
Freeing allocated pool space from thin devices

Space that has been allocated or preallocated for a thin device but not yet written to can be freed and made available to the other devices in the thin pool. Here, all of the allocated space on device DD7, which was preallocated when the device was bound to the pool, is freed:

Device DD7 no longer has any allocated tracks:
Setting oversubscription limit

Thin devices can be oversubscribed, meaning that there can be less physical storage space in the thin pool than total space of all the thin devices that are bound to the pool. The default value for the oversubscription limit is NONE, meaning that oversubscription is not restricted. This means that thin devices can only be bound to a pool if the size of the thin devices will not exceed the size of the enabled data devices in the pool.

If oversubscription is desired, an oversubscription limit can be set to prevent the oversubscription from reaching a specified level. This limit can be set by selecting the Max Subscription % checkbox in Add/Remove Pool Members and adding the desired value.

This value can be set at the time a device or devices are added to a pool. It can also be used to change the value for an existing pool; however, the option will be grayed out if the chosen pool is empty or contains no enabled devices.

In this example, the limit is set at 150 percent for the HR pool, meaning that the total amount of space for bound thin devices cannot exceed 150 percent of the space provided by the data devices:
The properties of the HR pool now show that the oversubscription limit has been set at 150 percent:

<table>
<thead>
<tr>
<th>Pool</th>
<th>Emulation</th>
<th>Dev. Coding</th>
<th>Total (MB)</th>
<th>Enabled (MB)</th>
<th>Alloc (MB)</th>
<th>Alloc %</th>
<th>Subs %</th>
<th>State</th>
<th>Total (Virtual)</th>
<th>Free</th>
</tr>
</thead>
<tbody>
<tr>
<td>HR</td>
<td>FBA</td>
<td>RAID-S (2 + 1)</td>
<td>24025</td>
<td>20024</td>
<td>12946</td>
<td>229</td>
<td>67</td>
<td>Enabled</td>
<td>327219</td>
<td></td>
</tr>
<tr>
<td>VPO</td>
<td>FBA</td>
<td>RAID-S (3 + 1)</td>
<td>4071</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Disabled</td>
<td>12917</td>
<td></td>
</tr>
</tbody>
</table>

The oversubscription limit will prevent the user from adding thin devices that push the oversubscription ratio past 150 percent. In this case, the HR pool contains nine 4602 cylinder data devices and six 4602 cylinder thin devices. Attempting to create and bind twelve 4602 cylinder thin devices will fail:
Unbinding thin devices from a pool

In order to unbind thin devices from a pool they must be unmasked and unmapped or Not Ready. In this case, neither of the two devices being unbound is mapped to an FA. Devices DDD and DDE are selected and then are unbound by clicking Remove, adding the change to the configuration session list, and running the change.
They have now been unbound and can now be bound to another pool or deleted.

Disabling and removing data devices from a pool

Support for the draining of data devices has been added to Enginuity 5874 for Symmetrix VMAX arrays. This allows data devices to be removed from thin pools non-disruptively.

Before removing a data device from a pool, the device must be disabled. Disabling a data device changes its state from Enabled to Draining. The concept of device draining is common to all pool types (data, DSE, and snap pools).

When a device enters a draining state, used tracks on that device are copied to the remaining, enabled devices in the pool. Therefore, data devices can be removed from a pool and repurposed without having to unbind thin devices that may have allocated extents associated with the data device.

A data device can be disabled by right-clicking on the data device and choosing Device Pool Management then Disable Device:
Virtual Provisioning

After it has drained, the state will be Disabled. The used tracks on device DDB have been copied to the other enabled devices in the pool:

Device DDB can now be removed:
Best Practices for Fast, Simple Capacity Allocation with Symmetrix Virtual Provisioning
Deleting thin device pools

Before deleting a thin pool, any bound thin devices must be unbound and all data devices must be disabled and removed.

After unbinding all thin devices, disabling all data devices and removing all data devices, the pool is now empty and can be deleted.

Configuring thin metadevices

Thin devices can also be created and formed into metadevices. Metadevices made from thin volumes are created in the same manner as metadevices created from regular volumes.

Thin metadevices must be created before the thin devices are bound to a thin pool. If an attempt is made to form a metadevice from bound devices, the operation will fail.

In the following example, a four-member, concatenated metadevice is created from existing thin devices:
After the meta has been created it can then be bound to the pool in the same way as non-metadevices:
Meta information is now available in a Meta Members tab in Properties view:

Once the metadevice is bound to the pool, it is read/write enabled. Like metadevices made up of regular Symmetrix volumes, thin metadevices can also be expanded.
Best Practices for Fast, Simple Capacity Allocation with Symmetrix Virtual Provisioning
The configuration change can now be run. No further action is required to read/write enable the new metamember. Because it is now part of a metadevice that is bound to a thin pool, it is bound to the same pool.

Device EF2 is now a five-member meta:

<table>
<thead>
<tr>
<th>Property - 000100102030 LocalMETA</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>General</strong></td>
<td></td>
</tr>
<tr>
<td>Dev</td>
<td>Pool</td>
</tr>
<tr>
<td>EF2</td>
<td>META</td>
</tr>
</tbody>
</table>

Members can also be removed from concatenated metadevices. In order to perform this change, the meta must be unmasked or unmapped. If a metamember is removed from a metadevice it will remain bound to the thin pool.

**Note:** Striped thin metadevices cannot be expanded while preserving data.

When a striped metadevice is expanded, a BCV meta must be established to the standard metadevice so that the data can be restored to the newly grown standard meta following the expansion. Because thin devices cannot have BCVs attached to them, meta expansion while preserving data is not possible with striped thin metadevices.

**Monitoring thin pool space**

In 5773 Enginuity with DMX-3 and DMX-4 arrays, alerts can be configured within SMC by clicking Administration followed by Alerts:
There are three alerts that can be selected that are applicable specifically for device pools and two for thin devices:
The five alerts applicable to Virtual Provisioning will inform the user of various changes in the status of thin devices and thin pools:

- **Device Pool Config Change** – Alerts the user that the status of the pool has changed as a result of adding or removing data devices or binding or unbinding thin devices.
- **Device Pool Free Space** – Alerts the user when the amount of free space remaining in a device pool meets or exceeds a specific percentage.
- **Device Pool Status** – Alerts the user if the state of a pool changes. One example is if the pool state changes from Enabled to Disabled because all of the data devices are disabled by the user, an alert will be triggered.
- **Thin Device Allocation** – Alerts the user when the amount of allocated space on a thin device meets or exceeds a specific percentage.
- **Thin Device Usage** – Alerts the user when the amount of used space on a thin device meets or exceeds a specific percentage.

There are three types and levels of alerts that can be triggered by SMC. Alerts will either be warning, critical, or fatal depending on the type of alert and the threshold that is being equaled or exceeded.

Alerts that are looking at pool or thin device usage or allocation (Device Pool Free Space, Thin Device Allocation, and Thin Device Usage) will trigger when the following thresholds are met or exceeded:

- 60% - Issue a warning alert
- 65% - Issue a warning alert
- 70% - Issue a warning alert
- 80% - Issue a critical alert
- 100% - Issue a fatal alert

Alerts that have not been cleared by the user can be viewed by clicking the Alerts button. They can alternately be viewed by clicking the New Alerts button at the upper right of the browser window. This button will also show the number of new alerts (alerts that have not been acknowledged by the user).
This alert was triggered when a data device in the SALES pool was disabled:

After mapping and masking the thin devices from the SALES pool to a host, file systems were created on devices 0EEE and 0EEF. The file systems were then mounted and files were created on both devices. The file system on device 0EEE was completely filled, while 0EEF was filled to about 85 percent of capacity. The resulting Alerts were captured:
Virtual Provisioning

In SMC 7.0, which supports Symmetrix VMAX arrays running Enginuity 5874, the Config Alerts dialog box can be found within SMC by clicking Tasks and clicking Config Alerts ... under Setup:

Four of the five alerts available in earlier versions of SMC can be configured as they were for 5773 Enginuity. Checkboxes for Device Pool Config Change, Device Pool Status, Thin Device Allocation, and Thin Device Usage can be selected and will provide alerts at the same thresholds that were applicable to 5773 Enginuity:
The remaining alert, for Device Pool Free Space, can now be customized by the user to allow specific pools to be monitored for user-defined thresholds.

The dialog box for configuring pool thresholds can be found in the Tasks screen by clicking Config Pool Utilization Threshold... under Administration:

![Config Alerts dialog box]

Best Practices for Fast, Simple Capacity Allocation with Symmetrix Virtual Provisioning
In the Config Pool Utilization Threshold dialog box, all of the pools configured in the selected Symmetrix VMAX array will appear. Each has a checkbox that allows the pool to be selected. It also has a menu for each alert level (Warning, Critical, and Fatal) that allows the user to set a custom threshold between 5 percent and 100 percent in 5 percent increments for each individual pool.

In this example, the thin pool called “test” is set to log alerts at non-default thresholds of 40 percent, 60 percent, and 80 percent:
Local replication with TimeFinder/Snap

Thin devices can be used with TimeFinder/Snap in the same way that other types of Symmetrix devices can be used. Point-in-time snap copies of thin devices can be created on VDEVs, which can subsequently be presented to a host. The VDEVs can also be updated and the changes on the VDEVs can be restored back to the source thin devices or to different thin devices.

Thin devices are classified as standard devices and can be added to Symmetrix device groups.

A device group called “thinsnapdg” has been created that contains four thin devices along with four virtual devices:

TimeFinder/Snap operations can be performed against the thin devices in the same way they can be performed against any regular device. Here a snap session is created:

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

Best Practices for Fast, Simple Capacity Allocation with Symmetrix Virtual Provisioning
The snap sessions have been created:

They can now be activated:
Virtual Provisioning

Local replication using TimeFinder/Clone

TimeFinder/Clone can be used to create point-in-time copies of thin devices on other thin devices. Starting in Enginuity 5874, clone sessions can also be created between regular and thin devices with the thin device being the clone source, the clone target, or both.

Performing clone operations using thin devices is exactly the same as creating clones of regular devices. Thin clone targets can be presented to a host and accessed just like regular devices or clone source devices. Thin clone targets can also be updated and the changes on the targets can be
restored back to the source thin devices, to different thin devices, or (in Enginuity 5874) to a regular device.

A device group called “thinclone_dg” has been created that contains two thin devices added to the group as standard (STD) devices and two added as target (TGT) devices:

<table>
<thead>
<tr>
<th>Properties: thinclone_dg [102232]</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Vendor ID</strong></td>
<td>BMC Corp</td>
</tr>
<tr>
<td><strong>Application ID</strong></td>
<td>SMIC</td>
</tr>
<tr>
<td><strong>Group Type</strong></td>
<td>Regular</td>
</tr>
<tr>
<td><strong>Valid</strong></td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Device Priority GoS</strong></td>
<td>1</td>
</tr>
<tr>
<td><strong>Dynamic Cache Partition Name</strong></td>
<td>DEFAULT_PARTITION</td>
</tr>
<tr>
<td><strong>Device Group Creation Time</strong></td>
<td>2008-06-13 13:29:23.0</td>
</tr>
<tr>
<td><strong>Device Group Modifying Time</strong></td>
<td>2008-06-13 13:32:27.0</td>
</tr>
<tr>
<td><strong>Symmetrix ID</strong></td>
<td>000190003222</td>
</tr>
<tr>
<td><strong>Remote Symmetrix ID</strong></td>
<td></td>
</tr>
<tr>
<td><strong>RCBV Symmetrix ID</strong></td>
<td></td>
</tr>
<tr>
<td><strong>RRCBV Symmetrix ID</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Number of Associated Gatekeepers</strong></td>
<td>0</td>
</tr>
<tr>
<td><strong>Number of STD Devices in Group</strong></td>
<td>2</td>
</tr>
<tr>
<td><strong>Number of Locality-Associated BCOVs</strong></td>
<td>0</td>
</tr>
<tr>
<td><strong>Number of Locality-Associated VDEVs</strong></td>
<td>0</td>
</tr>
<tr>
<td><strong>Number of Locality-Associated TGTs</strong></td>
<td>2</td>
</tr>
<tr>
<td><strong>Number of Remotely-Associated BCOVs (STD-PDF)</strong></td>
<td>0</td>
</tr>
<tr>
<td><strong>Number of Remotely-Associated BCOVs (RCBV-PDF)</strong></td>
<td>0</td>
</tr>
<tr>
<td><strong>Number of Remotely-Associated VDEVs</strong></td>
<td>0</td>
</tr>
<tr>
<td><strong>Number of Remotely-Associated TGTs</strong></td>
<td>0</td>
</tr>
<tr>
<td><strong>Number of Host 1 BCOVs (Remotely-associated Host 2 BCOV)</strong></td>
<td>0</td>
</tr>
<tr>
<td><strong>Number of Host 2 VDEVs (Remotely-associated Host 1 VDEV)</strong></td>
<td>0</td>
</tr>
<tr>
<td><strong>Number of Host 2 TGTs (Remotely-associated Host 1 TGT)</strong></td>
<td>0</td>
</tr>
</tbody>
</table>

TimeFinder/Clone operations can be performed against the thin devices in the same way they can be performed against a regular device. Here a clone session is created:
In this example, DEV001 (0E0E) is being cloned to target TGT001 (0E10) and DEV002 (0E0F) is being cloned to target TGT002 (0E11):
The clone sessions have been created:

<table>
<thead>
<tr>
<th>Source Logical Device Name</th>
<th>DEV001</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source Symmetrix Device Name</td>
<td>DEV001</td>
</tr>
<tr>
<td>Number of Source Protected Tracks</td>
<td>0</td>
</tr>
<tr>
<td>Number of Source Modified Tracks</td>
<td>0</td>
</tr>
<tr>
<td>Target Logical Device Name</td>
<td>TGT001</td>
</tr>
<tr>
<td>Target Symmetrix Device Name</td>
<td>TGT001</td>
</tr>
<tr>
<td>Number of Target Modified Tracks</td>
<td>0</td>
</tr>
<tr>
<td>CGBP</td>
<td>XX</td>
</tr>
<tr>
<td>Snap Pair State</td>
<td>Created</td>
</tr>
</tbody>
</table>

Best Practices for Fast, Simple Capacity Allocation with Symmetrix Virtual Provisioning
They can now be activated:

![Replication - TimeFinder Clone](image1)

![Replication - TimeFinder Clone](image2)
Remote replication using SRDF

SRDF/Synchronous, SRDF/Asynchronous, or SRDF/Adaptive Copy can be used to perform replication of thin devices to thin devices on a remote Symmetrix array. Other than the requirement that both the source and target devices are thin, creating and managing thin SRDF devices is exactly the same as creating and managing SRDF using regular devices.

In this example, a R1 device group is created with four bound thin R1 devices in a local Symmetrix (devices 1164:1167 in Symm 3334):
The devices are in a synchronized state and can be the targets of an SRDF operation; in this example, a split:
**Automated pool rebalancing**

Starting in Enginuity 5874 and SMC 7.1, automated pool rebalancing allows the user to run a balancing operation that will redistribute data across the enabled data devices in the thin pool. Because the thin extents are allocated from the thin pool in a round-robin fashion, the rebalancing...
Virtual Provisioning

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

mechanism will be used primarily when adding data devices to increase thin pool capacity.

The balancing algorithm will determine the minimum, maximum, and mean used capacity values of the data devices in the thin pool. The Symmetrix will then move thin device extents from the data devices with the highest used capacity to those with the lowest until the pool is balanced.

In this example, the oraprod pool contains four enabled data devices and is 85 percent allocated:

In order to expand the pool, two new data devices have been added and enabled. The pool now has more capacity, but the spread of extents across the data devices is not balanced:

<table>
<thead>
<tr>
<th>Properties - 000192601254 Remote/oraprod</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>General</strong></td>
</tr>
<tr>
<td>Symmetrix ID</td>
</tr>
<tr>
<td>Pool Name</td>
</tr>
<tr>
<td>Pool Type</td>
</tr>
<tr>
<td>Pool Emulation</td>
</tr>
<tr>
<td>Dev Configuration</td>
</tr>
<tr>
<td>Number of Devices</td>
</tr>
<tr>
<td>Disabled Devices</td>
</tr>
<tr>
<td>Enabled Devices</td>
</tr>
<tr>
<td>Capacity (GB / MB / Trks)</td>
</tr>
<tr>
<td>Enabled Capacity (GB / MB / Trks)</td>
</tr>
<tr>
<td>Allocated Capacity (GB / MB / Trks)</td>
</tr>
<tr>
<td>Free Capacity (GB / MB / Trks)</td>
</tr>
<tr>
<td>Allocated %</td>
</tr>
<tr>
<td>Current Subscription %</td>
</tr>
<tr>
<td>Maximum Subscription %</td>
</tr>
<tr>
<td>Pool State</td>
</tr>
</tbody>
</table>
If the pool is not rebalanced, the original four devices will fill and data will then stripe only across the two newly added devices. In a real production environment with a large, wide striped pool, this could cause severe performance degradation. Running the rebalance operation will solve the problem:

![Image of Symmetrix Virtual Provisioning interface showing the option to start write balancing]

---

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

---
After starting the balancing operation, the second pool state changes from “N/A” to “Balancing”:

**Properties - 000192601254 Remote/oraprod**

<table>
<thead>
<tr>
<th>General</th>
<th>DATA Devices</th>
<th>Bound Thin Devices</th>
</tr>
</thead>
<tbody>
<tr>
<td>Symmetrix ID</td>
<td>000192601254</td>
<td></td>
</tr>
<tr>
<td>Pool Name</td>
<td>oraprod</td>
<td></td>
</tr>
<tr>
<td>Pool Type</td>
<td>Thin</td>
<td></td>
</tr>
<tr>
<td>Pool Emulation</td>
<td>FBA</td>
<td></td>
</tr>
<tr>
<td>Dev Configuration</td>
<td>RAID-5 (7 + 1)</td>
<td></td>
</tr>
<tr>
<td>Number of Devices</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>Disabled Devices</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Enabled Devices</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>Capacity (GB / MB / Trks)</td>
<td>25 / 25384 / 414144</td>
<td></td>
</tr>
<tr>
<td>Enabled Capacity (GB / MB / Trks)</td>
<td>25 / 25384 / 414144</td>
<td></td>
</tr>
<tr>
<td>Allocated Capacity (GB / MB / Trks)</td>
<td>14 / 14781 / 236496</td>
<td></td>
</tr>
<tr>
<td>Free Capacity (GB / MB / Trks)</td>
<td>10 / 11103 / 177848</td>
<td></td>
</tr>
<tr>
<td>Allocated %</td>
<td>57</td>
<td></td>
</tr>
<tr>
<td>Current Subscription %</td>
<td>133</td>
<td></td>
</tr>
<tr>
<td>Maximum Subscription %</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>Pool State</td>
<td>Enabled</td>
<td></td>
</tr>
<tr>
<td>Pool State</td>
<td>Balancing</td>
<td></td>
</tr>
</tbody>
</table>

The pool is in the process of copying extents from the first four data devices in the pool to the newly added devices:

**Properties - 000192601254 Remote/oraprod**

<table>
<thead>
<tr>
<th>Dev</th>
<th>Emulation</th>
<th>Dev Class</th>
<th>Dev ID</th>
<th>Pool State</th>
<th>% Used</th>
<th>Used (MB)</th>
<th>Used Tracks</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>FBA</td>
<td>RAID-5 (7 + 1)</td>
<td>Enabled</td>
<td></td>
<td>81</td>
<td>3810</td>
<td>96180</td>
</tr>
<tr>
<td>0001</td>
<td>FBA</td>
<td>RAID-5 (7 + 1)</td>
<td>Enabled</td>
<td></td>
<td>81</td>
<td>3496</td>
<td>95968</td>
</tr>
<tr>
<td>0002</td>
<td>FBA</td>
<td>RAID-5 (7 + 1)</td>
<td>Enabled</td>
<td></td>
<td>81</td>
<td>3205</td>
<td>96070</td>
</tr>
<tr>
<td>0003</td>
<td>FBA</td>
<td>RAID-5 (7 + 1)</td>
<td>Enabled</td>
<td></td>
<td>81</td>
<td>3000</td>
<td>96004</td>
</tr>
<tr>
<td>0004</td>
<td>FBA</td>
<td>RAID-5 (7 + 1)</td>
<td>Enabled</td>
<td></td>
<td>81</td>
<td>3814</td>
<td>6144</td>
</tr>
<tr>
<td>0005</td>
<td>FBA</td>
<td>RAID-5 (7 + 1)</td>
<td>Enabled</td>
<td></td>
<td>3</td>
<td>384</td>
<td>6144</td>
</tr>
</tbody>
</table>

After the operation has completed, the pool is balanced:
During a balancing operation, Enginuity will give the highest priority to front-end I/O operations. However, rebalancing can be stopped for any reason, such as an upcoming period of very high host I/O or an operation that will cause a large amount of internal copy tasks:

If balancing is stopped for any reason, it can be re-started by starting write balancing in the same way it was initially started.

Note: Rebalancing is designed to bring the utilized capacity of all devices within 10 percent tolerance. Testing has found that the natural growth of thin devices bound to the pool can lead to a variance of up to 10 percent. This variation is caused by the ability of the microcode to skip extent allocation on the next data dev in the pool. If that device’s DA is busy that DA can pass on
the allocation request to the DA that owns the next data dev in the pool so extent allocation does not have a large effect on host I/O.

Space reclamation

SMC 7.1 and Enginuity 5874 provide the ability to de-allocate thin device extents that are found to contain all zeros.

This feature is an extension of the existing Virtual Provisioning space de-allocation mechanism. Running the space reclamation command will spawn a DA background task that will examine the allocated extents on specified thin device. For each allocated extent, all 12 tracks will be brought into cache and scanned 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.

Space reclamation is designed to be used primarily following a migration from a regular device to a thin device. In this example, clone sessions are used to migrate data to a group of thin devices, which will be subsequently reclaimed.

Prior to the clone operation, the thin devices, which will be the clone targets, have no pool space other than the initial 12 track thin device extents allocated:

<table>
<thead>
<tr>
<th>Properties</th>
<th>000192501254 Remote/saprood</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dev</td>
<td>Pool</td>
</tr>
<tr>
<td>0039</td>
<td>4314</td>
</tr>
<tr>
<td>003A</td>
<td>4314</td>
</tr>
<tr>
<td>00AB</td>
<td>4314</td>
</tr>
<tr>
<td>00AC</td>
<td>4314</td>
</tr>
<tr>
<td>00AD</td>
<td>4314</td>
</tr>
<tr>
<td>00AE</td>
<td>4314</td>
</tr>
<tr>
<td>00AF</td>
<td>4314</td>
</tr>
<tr>
<td>00B0</td>
<td>4314</td>
</tr>
</tbody>
</table>

Note: Thin device extents are 768 KB in size. Because SMC shows the allocated capacity in MB, the value will be 0 until more than one extent is allocated.

The clone source devices, which are regular devices, contain file systems that are in use and have been written to:

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

```
# df -k

Filesystem            kbytes    used   avail capacity  Mounted on
/dev/dsk/c4t5000097208139959d39s2  4341486 3130353 1167719    73%    /mp/d39mp
/dev/dsk/c4t5000097208139959d40s2  4341486 1819569 2478503    43%    /mp/d40mp
/dev/dsk/c4t5000097208139959d41s2  4341486 2741481 1556591    64%    /mp/d41mp
/dev/dsk/c4t5000097208139959d42s2  4341486 1773993 2524079    42%    /mp/d42mp
/dev/dsk/c4t5000097208139959d43s2  4341486 2227761 2070311    52%    /mp/d43mp
/dev/dsk/c4t5000097208139959d44s2  4341486 1956553 2341519    46%    /mp/d44mp
/dev/dsk/c4t5000097208139959d45s2  4341486 2533417 1764655    59%    /mp/d45mp
/dev/dsk/c4t5000097208139959d46s2  4341486 2981529 1316543    70%    /mp/d46mp
```

A clone session is created between the regular source devices and the thin target devices and the data is copied:

```
The thin target devices now have extents that were allocated during the clone copy:
```
To return the all zero extents to the thin pool, a “Free Thin Device” operation can be run against the thin devices. The “Free Thin Device” operation is accessed by selecting a device or group of devices, right-clicking, and then by clicking “Device Configuration” and “Free Thin
Virtual Provisioning

Device”.

Multiple devices can be selected by either shift-clicking or ctrl-clicking on the devices in the pool:

In the “Start Allocate/Free Thin Device” dialog box, the “Start Free” button can be selected, which will un-grey the “Free Type” buttons.

Two actions can be chosen; either “Unwritten” to free only extents that have never been written to by a host or “Reclaim” to free both never written and all zero extents. There is also a choice to free the entire device or to choose a range of cylinders to operate on:
After clicking the “Add to Config Session List” button and running the config change, the thin devices now have a status of “Reclaiming”:

After the reclaim operation has completed, the devices are back in a “Bound” state and the all zero and never written extents have been returned to the thin pool:
The pool is now only 30 percent allocated:

![Properties Table]

- **Symmetrix ID**: 000192601254
- **Pool Name**: oraprod
- **Pool Type**: Thin
- **Pool Emulation**: FBA
- **Dev Configuration**: RAID-5 (7 + 1)
- **Number of Devices**: 8
- **Disabled Devices**: 0
- **Enabled Devices**: 8
- **Capacity (GB/MB/Trks)**: 33 / 34512 / 552192
- **Allocated Capacity (GB/MB/Trks)**: 10 / 10459 / 167352
- **Free Capacity (GB/MB/Trks)**: 23 / 24052 / 384840
- **Allocated %**: 30
- **Current Subscription %**: 100
- **Maximum Subscription %**: None
- **Pool State**: Enabled
- **Pool State**: N/A
Conclusion

It is critical to define the processes, applications, and workloads for which Virtual Provisioning can be used most effectively, as well as the specific benefits that can be achieved. When implemented appropriately, Virtual Provisioning can be a powerful complement to organizations’ processes and technologies for improving ease of use, enhancing performance, and utilizing storage capacity more efficiently. Symmetrix Virtual Provisioning integrates well with existing management and business continuity technologies, and is an important advancement in capabilities for DMX-3, DMX-4, and Symmetrix VMAX customers.
Appendix A: Terminology

Basic Symmetrix terms

- **Device** — A logical unit of storage defined within an array.
- **Device Capacity** — The storage capacity of a device.
- **Device Extent** — Specifies a quantum of logically contiguous blocks of storage.
- **Host Accessible Device** — A device that can be made available for host use.
- **Internal Device** — A device used for a Symmetrix internal function that cannot be made accessible to a host.
- **Metadevice** — An aggregation of host accessible devices seen from the host as a single device.
- **Pool** — A collection of internal devices for some specific purpose.

Thin provisioning terms

- **Bind** — Refers to the act of associating one or more thin devices with a thin pool.
- **Data Device** — An internal device that provides storage capacity to be used by thin devices.
- **Disabled Data Device** — A data device belonging to a thin pool from which capacity cannot be allocated for thin devices. This state is under user control.
- **Enabled Data Device** — A data device belonging to a thin pool from which capacity can be allocated for thin devices. This state is under user control.
- **Extent Mapping** — Specifies the relationship between thin device and data device extents.
- **Oversubscribed Thin Pool** — A thin pool whose thin pool capacity is less than the sum of the reported sizes of the thin devices bound to the pool.
- **Pre-provisioning** — Allocating a range of extents from the thin pool to the thin device at the time the thin device is bound. Sometimes used to reduce the operational impact of allocating extents or to eliminate the potential for a thin device to run out of available extents.
Virtual Provisioning

**Thin Device** — A host accessible device that has no storage directly associated with it.

**Thin Device Allocated Capacity** — The capacity that has been allocated from the thin pool enabled capacity for the exclusive use of a thin device.

**Thin Device Extent** — The minimum quantum of storage that must be mapped at a time to a thin device.

**Thin Device Subscription Ratio** — This is the ratio between a thin device subscribed capacity and the thin pool enabled capacity. This value is expressed as a percent.

**Thin Device User Preallocated Capacity** — The initial amount of capacity that is allocated when a thin device is bound to a thin pool. This property is under user control.

**Thin Device Utilization Ratio** — The ratio between the thin device written capacity and thin device allocated capacity. This value is expressed as a percent.

**Thin Device Written Capacity** — The capacity on a thin device that was written to by a host. In most implementations this is a subset of the thin device allocated capacity.

**Thin Pool** — A container that can have data devices and bound thin devices.

**Thin Pool Allocated Capacity** — A subset of thin pool enabled capacity that has been allocated for the exclusive use of all thin devices bound to that thin pool.

**Thin Pool Allocation Ratio** — The ratio between the thin pool allocated capacity and thin pool enabled capacity. This value is expressed as a percent.

**Thin Pool Capacity** — The sum of the capacities of the member data devices.

**Thin Pool Enabled Capacity** — The sum of the capacities of enabled data devices belonging to a thin pool.

**Thin Pool Subscription Ratio** — The ratio between the sum of the thin device subscribed capacity of all its bound thin devices and the associated thin pool enabled capacity. This value is expressed as a percent.
Appendix B: Virtual Provisioning and host file systems

One of the factors in a Virtual Provisioning environment that needs to be considered when file systems are used is the amount of metadata written to the thin device when a file system is created. File systems that are inefficient and write large amounts of metadata can cause thin devices to become fully allocated more quickly than with efficient file systems. File system efficiency with Virtual Provisioning varies depending on the file system being used. Newer journaling file systems are very space efficient and will not cause large amounts of thin pool space to be allocated when the file systems are created. Other factors in an application environment may also cause thin devices placed under the control of an application to rapidly become fully allocated. These are covered in the section entitled “Thin-hostile environments.”

The following chart shows how much space is allocated from the thin pool when common file systems are created. The example file systems were built on thin devices with a capacity of 18414 cylinders (276210 tracks or 17263 MB). For the purpose of these tests, file systems were created directly on the LUN with Solaris, Linux, and Windows. With HP-UX and with AIX and VxFS, a logical volume was created that spanned the entire physical volume. With AIX and JFS/JFS2, a logical volume was created that spanned the entire physical volume with the exception of one physical partition reserved for the log logical volume.

<table>
<thead>
<tr>
<th>Host</th>
<th>File System</th>
<th>Allocated at File System Creation - Devices are 276210 tracks / 17263 MB</th>
<th>Efficiency with Thin Devices</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Number of Tracks</td>
<td>Megabytes</td>
</tr>
<tr>
<td>AIX</td>
<td>JFS</td>
<td>35040</td>
<td>2190</td>
</tr>
<tr>
<td></td>
<td>JFS2</td>
<td>612</td>
<td>38</td>
</tr>
<tr>
<td></td>
<td>VxFS</td>
<td>612</td>
<td>38</td>
</tr>
<tr>
<td>HP-UX</td>
<td>HFS</td>
<td>27924</td>
<td>1745</td>
</tr>
<tr>
<td></td>
<td>JFS (VxFS)</td>
<td>300</td>
<td>19</td>
</tr>
<tr>
<td>Linux</td>
<td>VxFS</td>
<td>1080</td>
<td>1030</td>
</tr>
<tr>
<td></td>
<td>Ext2</td>
<td>6144</td>
<td>384</td>
</tr>
<tr>
<td></td>
<td>Ext3</td>
<td>6648</td>
<td>416</td>
</tr>
<tr>
<td>Solaris</td>
<td>JFS</td>
<td>8676</td>
<td>542</td>
</tr>
<tr>
<td></td>
<td>VxFS</td>
<td>1080</td>
<td>68</td>
</tr>
<tr>
<td>Windows</td>
<td>NTFS</td>
<td>1116</td>
<td>70</td>
</tr>
</tbody>
</table>
Note: Host logical volume managers take a minimal amount of space for a disk header or private region when devices are initialized for use. As an example, Veritas Volume Manager 4.1 on Solaris causes 36 tracks to be allocated from the thin pool; HP-UX LVM (11.23) causes 30 tracks to be allocated. LVM information is written to that space only. Although pool written tracks may increase with further LVM activity, pool allocated tracks do not increase because the size of the private region or disk header does not change after the disk is initialized by the LVM.