

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

Technical Note

P/N 300-006-718

REV A07

April 14, 2009

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	5
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.....	16
Virtual Provisioning configuration guidelines	17
Adding capacity to a thin pool that is running out of space.....	20
Ensuring that DATA devices are spread wide across the back end .	21
Migrating to thin devices	21
Dynamic Cache Partitioning	22

Local and remote replication	23
Local replication with SRDF/A devices	23
Data device protection	24
Data and thin device restrictions	24
Thin pool restrictions	25
Virtual Provisioning with Solutions Enabler	25
Configuring and viewing data devices and pools	25
Configuring, binding, and viewing thin devices	30
Adding additional data devices to a pool	33
Binding additional thin devices to a pool	36
Preallocating pool space for thin devices	37
Allocating pool space for thin devices	39
Freeing allocated pool space from thin devices	42
Setting a maximum subscription percent	43
Unbinding thin devices from a pool	46
Disabling and removing data devices from a pool	47
Deleting thin pools	49
Configuring thin metadevices	50
Monitoring thin pool space	58
Local replication using TimeFinder/Snap	61
Local replication using TimeFinder/Clone	64
Remote replication using SRDF	67
Remote replication using Open Replicator	72
Virtual Provisioning with Symmetrix Management Console	72
Configuring and viewing data devices and pools	72
Adding additional data devices to a pool	77
Configuring, binding, and viewing thin devices	79
Preallocating pool space for thin devices	81
Allocating pool space for thin devices	83
Freeing allocated pool space from thin devices	86

Setting oversubscription limit	87
Unbinding thin devices from a pool.....	89
Disabling and removing data devices from a pool	90
Deleting thin device pools	93
Configuring thin metadevices.....	93
Monitoring thin pool space.....	97
Local replication with TimeFinder/Snap	104
Local replication using TimeFinder/Clone.....	107
Remote replication using SRDF	112
Conclusion	116
Appendix A: Terminology	117
Basic Symmetrix terms	117
Thin provisioning terms.....	117
Appendix B: Virtual Provisioning and host file systems.....	120

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 tiers. 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.

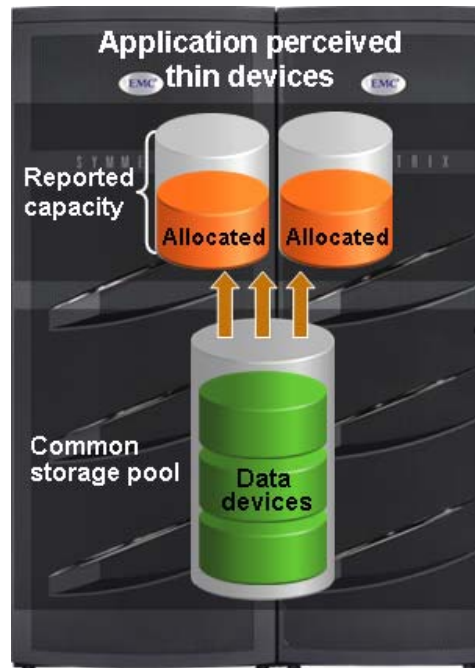


Figure 1. Symmetrix Virtual Provisioning

Audience

This technical note is intended for the technology professional who works in an environment with Symmetrix DMX-3 and DMX-4, and Symmetrix V-Max™ with Enginuity™ disk 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 disk 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.” 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*.

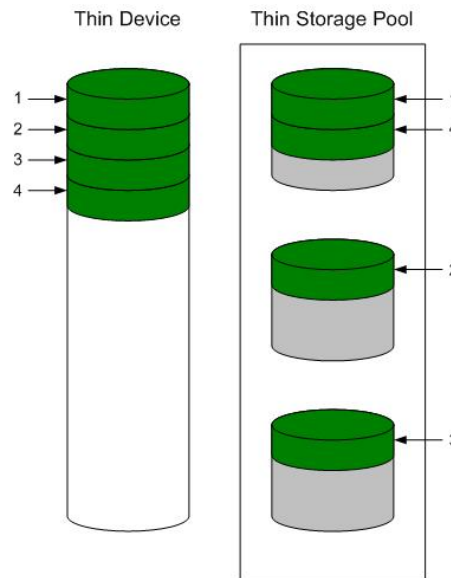


Figure 2. Thin device and thin storage pool containing data devices

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.

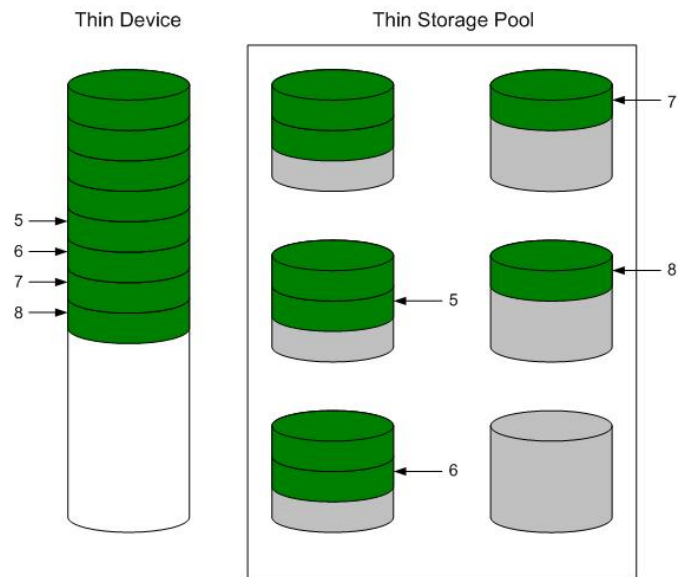


Figure 3. Thin device and thin storage pool with added data devices

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.

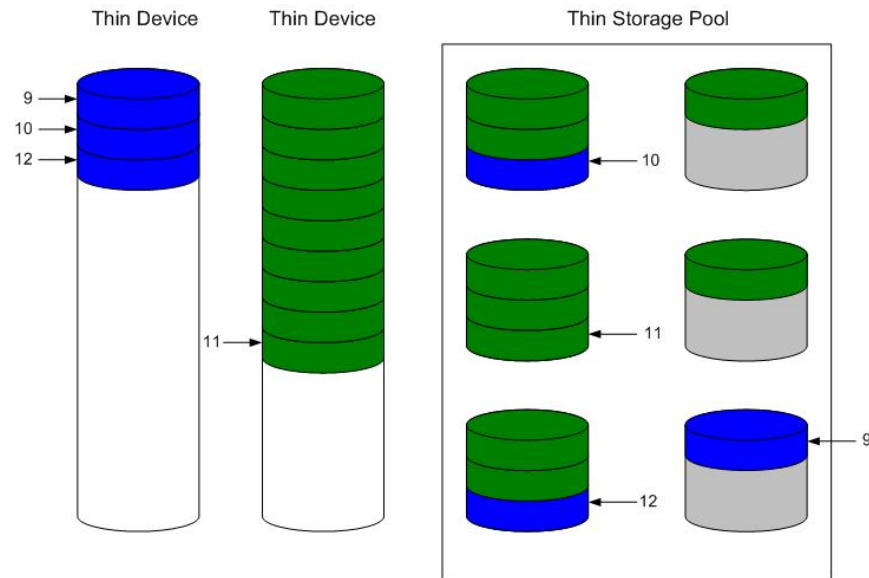


Figure 4. Addition of a thin device

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 only requires that data devices be 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 as well as 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

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 be 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. Prior to disabling a data device in Enginuity 5773, all allocated tracks on the data device must be released by unbinding the thin devices that have allocated extents on the data device. Because data on the thin device will be striped across all enabled devices in the pool, in practice all thin devices in a pool will need to be unbound before data devices can be disabled and removed from the pool. Starting in Enginuity 5874, data devices that contain allocated extents can be disabled. This 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.
- Execution of a set of array management operations to create and present new LUNs or to enlarge existing LUNs.
- Execution of 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.
- Execution of 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 disk 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 of purchase and installation of additional physical disks.

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 disk capacity until it is bound to a pool, creating a thin device in advance will not consume

any disk 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 metadvice is striped evenly across a set of physical disks. 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.

Greater flexibility with concatenated 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 meta member. 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 existing data by adding meta members to an existing metadevice.
- Nonmetadevices can be converted to concatenated metadevices without destroying existing data.

Caution: Striped thin metadevices cannot be expanded while preserving data. When a regular 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.

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 disks 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. For details, see the *EMC Host Connectivity Guide* for the specific host platform that will be accessing thin devices bound to an oversubscribed pool. Host Connectivity Guides for all supported platforms are available on Powerlink®.

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 disk 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 nondisruptively, 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 on 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 be 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 disk 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 be 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 disk won't 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 the data devices are evenly spread across the back end.
- The devices in the pool should use an equal amount of space from each drive.
- It is recommended that all data devices in a pool be 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 disks 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 disk 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 nondisruptively. 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.
- They should use an equal amount of space from each drive underlying the addition.
- It is recommended that all data devices in a pool be 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.

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). When greater pool capacity is required, the pool should be expanded by the same number of disks that were in the original pool.

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.

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 nonoptimal 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 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.

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

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.

Note: The ACOFW performance optimization is not supported when using TimeFinder/Snap with a thin device. Data devices may not be used as the source or target of any type of TimeFinder operation.

Local replication with SRDF/A devices

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 can't 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.

Note: 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.

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 data devices will not be copied. To protect data devices, permanent member sparing must be used.
- Data devices cannot be meta devices.
- 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.

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 Tech 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
```

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 020 of 078 steps.....Executing.

Step 027 of 078 steps.....Executing.

Step 029 of 078 steps.....Executing.

Step 040 of 078 steps.....Executing.

Step 040 of 078 steps.....Executing.

Step 056 of 151 steps.....Executing.

Step 065 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 082 of 173 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 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 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

Device Name		Directors		Device			
Sym	Physical	SA :P DA :IT	Config	Attribute	Sts	Cap (MB)	
10E4	Not Visible	??? : 01A:C4	2-Way Mir	N/A	(DT) RW	4314	
10E5	Not Visible	??? : 16C:D4	2-Way Mir	N/A	(DT) RW	4314	
10E6	Not Visible	??? : 15B:D4	2-Way Mir	N/A	(DT) RW	4314	
10E7	Not Visible	??? : 02D:C4	2-Way Mir	N/A	(DT) RW	4314	
10E8	Not Visible	??? : 16A:D4	2-Way Mir	N/A	(DT) RW	4314	
10E9	Not Visible	??? : 01C:C4	2-Way Mir	N/A	(DT) RW	4314	
10EA	Not Visible	??? : 16B:C4	2-Way Mir	N/A	(DT) RW	4314	
10EB	Not Visible	??? : 01D:D4	2-Way Mir	N/A	(DT) RW	4314	

10EC	Not Visible	????: 02A:C7	2-Way Mir	N/A	(DT) RW	4314
10ED	Not Visible	????: 15C:D7	2-Way Mir	N/A	(DT) RW	4314
10EE	Not Visible	????: 02B:D7	2-Way Mir	N/A	(DT) RW	4314
10EF	Not Visible	????: 15D:C7	2-Way Mir	N/A	(DT) RW	4314
10F0	Not Visible	????: 15A:D7	2-Way Mir	N/A	(DT) RW	4314
10F1	Not Visible	????: 02C:C7	2-Way Mir	N/A	(DT) RW	4314
10F2	Not Visible	????: 15B:C7	2-Way Mir	N/A	(DT) RW	4314
10F3	Not Visible	????: 02D:D7	2-Way Mir	N/A	(DT) RW	4314

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 Delta Set Extension (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
```

S Y M M E T R I X P O O L S								
Pool Name	Type	Dev Emul	Dev Config	Total Tracks	Enabled Tracks	Used Tracks	Free Tracks	Full (%)
HR	TH	FBA	2-Way Mir	138048	138048	0	138048	0

```

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

```

          S Y M M E T R I X   P O O L S
-----
Pool      Ty      Dev      Dev      Total  Enabled  Used    Free  Full  Sta
Name      p Dev    Config    MBs    MBs    MBs    MBs    (%)   t
-----
HR         TH FBA    2-Way Mir  8628    8628    0      8628    0   Ena

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):
{

```

```

-----
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

```

          S Y M M E T R I X   D A T A   D E V I C E S
-----
Sym   Dev  Dev      Pool      Total      Used Full
Emul  Config  Type Pool Name  State  Tracks  Tracks (%)
-----
10E4 FBA  2-Way Mir  TH  HR          Ena      69024      0    0
10E5 FBA  2-Way Mir  TH  HR          Ena      69024      0    0
10E6 FBA  2-Way Mir  -  -          Dis      69024      0    0
10E7 FBA  2-Way Mir  -  -          Dis      69024      0    0
10E8 FBA  2-Way Mir  -  -          Dis      69024      0    0
10E9 FBA  2-Way Mir  -  -          Dis      69024      0    0
10EA FBA  2-Way Mir  -  -          Dis      69024      0    0
10EB FBA  2-Way Mir  -  -          Dis      69024      0    0
10EC FBA  2-Way Mir  -  -          Dis      69024      0    0
10ED FBA  2-Way Mir  -  -          Dis      69024      0    0
10EE FBA  2-Way Mir  -  -          Dis      69024      0    0
10EF FBA  2-Way Mir  -  -          Dis      69024      0    0
10F0 FBA  2-Way Mir  -  -          Dis      69024      0    0
10F1 FBA  2-Way Mir  -  -          Dis      69024      0    0
10F2 FBA  2-Way Mir  -  -          Dis      69024      0    0
10F3 FBA  2-Way Mir  -  -          Dis      69024      0    0
1104 FBA  2-Way Mir  -  -          Dis      69045      0    0
1105 FBA  2-Way Mir  -  -          Dis      69045      0    0
1106 FBA  2-Way Mir  -  -          Dis      69045      0    0
1107 FBA  2-Way Mir  -  -          Dis      69045      0    0
1108 FBA  2-Way Mir  -  -          Dis      69045      0    0
1109 FBA  2-Way Mir  -  -          Dis      69045      0    0
110A FBA  2-Way Mir  -  -          Dis      69045      0    0
110B FBA  2-Way Mir  -  -          Dis      69045      0    0
110C FBA  2-Way Mir  -  -          Dis      69045      0    0
110D FBA  2-Way Mir  -  -          Dis      69045      0    0
110E FBA  2-Way Mir  -  -          Dis      69045      0    0
110F FBA  2-Way Mir  -  -          Dis      69045      0    0
-----
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

Device Name		Directors		Device			
Sym	Physical	SA :P	DA :IT	Config	Attribute	Sts	Cap (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
10FC	Not Visible	???:?	NA:NA	TDEV	N/Grp'd	NR	4314
10FD	Not Visible	???:?	NA:NA	TDEV	N/Grp'd	NR	4314
10FE	Not Visible	???:?	NA:NA	TDEV	N/Grp'd	NR	4314
10FF	Not Visible	???:?	NA:NA	TDEV	N/Grp'd	NR	4314
1100	Not Visible	???:?	NA:NA	TDEV	N/Grp'd	NR	4314
1101	Not Visible	???:?	NA:NA	TDEV	N/Grp'd	NR	4314

```

1102 Not Visible      ????:?  NA:NA  TDEV          N/Grp'd      NR      4314
1103 Not Visible      ????:?  NA:NA  TDEV          N/Grp'd      NR      4314

```

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
```

```
Symmetrix ID: 000190103334
```

```

Enabled Capacity (Tracks) :    138048
Bound   Capacity (Tracks) :    276120

```

S Y M M E T R I X T H I N D E V I C E S								
Sym	Pool Name	Dev Emul	Total Tracks	Pool Subs (%)	Alloc Tracks	Pool Alloc (%)	Wrtn Tracks	Pool Wrtn (%)
10F4	HR	FBA	69030	50	12	0	1	0
10F5	HR	FBA	69030	50	12	0	1	0
10F6	HR	FBA	69030	50	12	0	1	0
10F7	HR	FBA	69030	50	12	0	1	0
10F8	-	FBA	69030	0	0	0	0	0
10F9	-	FBA	69030	0	0	0	0	0
10FA	-	FBA	69030	0	0	0	0	0
10FB	-	FBA	69030	0	0	0	0	0
10FC	-	FBA	69030	0	0	0	0	0
10FD	-	FBA	69030	0	0	0	0	0
10FE	-	FBA	69030	0	0	0	0	0
10FF	-	FBA	69030	0	0	0	0	0
1100	-	FBA	69030	0	0	0	0	0
1101	-	FBA	69030	0	0	0	0	0
1102	-	FBA	69030	0	0	0	0	0
1103	-	FBA	69030	0	0	0	0	0
1110	-	FBA	69045	0	0	0	0	0
1111	-	FBA	69045	0	0	0	0	0
1112	-	FBA	69045	0	0	0	0	0
1113	-	FBA	69045	0	0	0	0	0
1114	-	FBA	69045	0	0	0	0	0
1115	-	FBA	69045	0	0	0	0	0
1116	-	FBA	69045	0	0	0	0	0
1117	-	FBA	69045	0	0	0	0	0
1118	-	FBA	69045	0	0	0	0	0
1119	-	FBA	69045	0	0	0	0	0
111A	-	FBA	69045	0	0	0	0	0
111B	-	FBA	69045	0	0	0	0	0
Total Tracks			1933020	1400	48	0	4	0

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

the thin devices bound to the pool are displayed:

```
# 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
-----
10E4      4314        775        3538   17   Enabled
10E5      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. See Virtual Provisioning best practices documentation for 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):
```

```
{
-----
Sym      Total      Alloc      Free Full  Device
Dev      MBs        MBs        MBs  (%)  State
-----
10E4      4314        775        3538   17  Enabled
10E5      4314        769        3544   17  Enabled
-----
MBs      8628        1545        7083   17
}
```

```
Disabled Devices(2):
```

```
{
-----
Sym      Total      Alloc      Free Full  Device
Dev      MBs        MBs        MBs  (%)  State
-----
10E6      4314         0        4314    0  Disabled
10E7      4314         0        4314    0  Disabled
-----
MBs      8628         0        8628    0
}
```

```
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
}

```

They can be enabled with a separate operation:

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

```
# 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):
```

```

{
-----
Sym      Total    Alloc    Free Full  Device
Dev      MBs      MBs      MBs  (%)  State
-----
10E4      4314      775      3538   17  Enabled
10E5      4314      769      3544   17  Enabled
10E6      4314        0      4314    0  Enabled
10E7      4314        0      4314    0  Enabled
-----
MBs      17256      1545      15711   8
}

```

```
Thin Devices(4):
```

```

{
-----
Sym      Total  Subs    Alloc Alloc    Wrtn  Wrtn
Dev      MBs   (%)    MBs  (%)    MBs   (%)
-----
10F4      4314   25      386    2      324    2
10F5      4314   25      386    2      324    2
10F6      4314   25      386    2      324    2
10F7      4314   25      386    2      324    2
-----
MBs      17257  100      1545    9      1297    8
}

```

New writes to the thin devices will now be spread equally across all four enabled data 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  : 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
-----
10E4      4314      1023      3291   23  Enabled
10E5      4314      1020      3293   23  Enabled
10E6      4314       248      4065    5  Enabled
10E7      4314       249      4065    5  Enabled
-----
MBs        17256      2541      14715   14
}

```

Thin Devices(4):

```

{
-----
Sym      Total  Subs      Alloc Alloc      Wrtn  Wrtn
Dev      MBs   (%)      MBs   (%)      MBs   (%)
-----
10F4      4314    25      635    4      574    3
10F5      4314    25      635    4      574    3
10F6      4314    25      635    4      574    3
10F7      4314    25      635    4      574    3
-----
MBs        17257   100      2541   15      2297   13
}

```

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
```

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):

Sym Dev	Total MBs	Alloc MBs	Free MBs	Full (%)	Device State
10E4	4314	1023	3290	23	Enabled
10E5	4314	1020	3293	23	Enabled
10E6	4314	248	4065	5	Enabled
10E7	4314	249	4064	5	Enabled
MBs	17256	2542	14713	14	

Thin Devices(6):

Sym Dev	Total MBs	Subs (%)	Alloc MBs	Alloc (%)	Wrtn MBs	Wrtn (%)
10F4	4314	25	635	4	574	3
10F5	4314	25	635	4	574	3
10F6	4314	25	635	4	574	3
10F7	4314	25	635	4	574	3
10F8	4314	25	0	0	0	0
10F9	4314	25	0	0	0	0
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

```

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):

```

{
-----
Sym      Total      Used      Free  Full  Device  Session
Dev      MBs        MBs        MBs   (%)  State   Status
-----
10E8      4314         0        4314    0  Enabled  N/A
10E9      4314         0        4314    0  Enabled  N/A
10EA      4314         0        4314    0  Enabled  N/A
10EB      4314         0        4314    0  Enabled  N/A
-----
MBs      17256         0        17256    0
}

```

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

```

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 Dev	Total MBs	Alloc MBs	Free MBs	Full (%)	Device State
10E8	4314	2202	2111	51	Enabled
10E9	4314	2173	2140	50	Enabled
10EA	4314	2165	2148	50	Enabled
10EB	4314	2092	2221	48	Enabled
MBs	17256	8634	8622	50	

Thin Devices(8):

Sym Dev	Total MBs	Subs (%)	Alloc MBs	Alloc (%)	Wrtn MBs	Wrtn (%)
10FA	4314	25	4314	25	0	0
10FB	4314	25	4314	25	0	0
10FC	4314	25	0	0	0	0
10FD	4314	25	0	0	0	0
10FE	4314	25	0	0	0	0
10FF	4314	25	0	0	0	0
1100	4314	25	0	0	0	0
1101	4314	25	0	0	0	0
MBs	34515	200	8634	50	0	0

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 PRODDDB 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 PRODDDB -thin -details -all -mb
```

```
Symmetrix ID: 000190103334
```

```
Symmetrix ID           : 000190103334
Pool Name               : PRODDDB
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 Dev	Total MBs	Alloc MBs	Free MBs	Full (%)	Device 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):

```
{
-----

```

Sym Dev	Total MBs	Pool Subs (%)	Pool Allocated MBs	Pool (%)	Pool Written MBs	Pool (%)	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
-----
1108         4315      3205      1111  74  Enabled
1109         4315      3134      1182  72  Enabled
110A         4315      2957      1358  68  Enabled
110B         4315      3652      664   84  Enabled
-----
MBs          17261      12947      4314  75
}

Thin Devices(4):
{
-----
Sym          Total    Pool      Pool      Pool
Dev          MBs      Subs      Allocated  Written
              MBs      (%)      MBs  (%)      MBs  (%)  Status
-----
1110         4315      25       4316      25       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      12947      75       0    0
}

```

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:

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

To allocate 1,000 cylinders starting at cylinder 2303:

```
# symconfigure -sid 34 -cmd "allocate tdev 1111 start_cyl=2303 size=1000;" commit
-nop
```

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

```
Symmetrix ID: 000190103334
```

```

Symmetrix ID          : 000190103334
Pool Name             : PRODDDB
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 Dev	Total MBs	Alloc MBs	Free MBs	Full (%)	Device State
1108	4315	3422	894	79	Enabled
1109	4315	3364	952	77	Enabled
110A	4315	3191	1125	73	Enabled
110B	4315	3910	406	90	Enabled
MBs	17261	13886	3376	80	

}

Thin Devices(4):

{

Sym Dev	Total MBs	Pool Subs (%)	Pool Allocated MBs	Pool (%)	Pool Written MBs	Pool (%)	Status
1110	4315	25	4316	25	0	0	Bound
1111	4315	25	939	5	0	0	Bound
1112	4315	25	4316	25	0	0	Bound
1113	4315	25	4316	25	0	0	Bound
MBs	17261	100	13886	80	0	0	

}

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 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
```

```

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
-----
1108      4315        2897        1418   67  Enabled
1109      4315        2892        1423   67  Enabled
110A      4315        2652        1663   61  Enabled
110B      4315        3288        1027   76  Enabled
-----
MBs      17261        11729        5532   67
}

```

Thin Devices(4):

```

{
-----
Sym      Total      Pool      Pool      Pool
Dev      MBs        Subs      Allocated  Written
              (%)      MBs    (%)      MBs    (%)  Status
-----
1110      4315        25        2159    13         0     0  Bound
1111      4315        25         939     5         0     0  Bound
1112      4315        25        4316    25         0     0  Bound
1113      4315        25        4316    25         0     0  Bound
-----
MBs      17261       100        11729    68         0     0
}

```

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% for a pool called DOCS, meaning that the total amount of configured space for bound thin devices cannot exceed 150% of the space provided by the

data devices:

```
# 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% because it has twice as much thin device space as data device space. This is indicated by the total in the "Subscription %" column (Subs %).

```
# 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 : None
```

Enabled Devices(4):

```
{
-----
Sym      Total      Alloc      Free Full  Device
Dev      MBs        MBs        MBs  (%)  State
-----
10E8      4314        2202        2111   51  Enabled
10E9      4314        2173        2140   50  Enabled
10EA      4314        2165        2148   50  Enabled
10EB      4314        2092        2221   48  Enabled
-----
MBs      17256        8634        8622   50
}
```

Thin Devices(8):

```
{
-----
Sym      Total  Subs      Alloc Alloc      Wrtn  Wrtn
Dev      MBs   (%)      MBs   (%)      MBs   (%)
-----
10FA      4314    25      4314    25          0    0
10FB      4314    25      4314    25          0    0
10FC      4314    25          0    0          0    0
10FD      4314    25          0    0          0    0
10FE      4314    25          0    0          0    0
10FF      4314    25          0    0          0    0
1100      4314    25          0    0          0    0
1101      4314    25          0    0          0    0
-----
}
```

```

MBS          34515    200      8634    50          0      0
}

```

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%. In this case, each thin device represents 25% 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.
```

```
The configuration change session has failed.
```

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
```

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
```

...

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:

```
# 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:
```

```
# symcfg show -pool SALES -thin -detail -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 : 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  Subs      Alloc Alloc      Wrtn  Wrtn
Dev      Tracks  (%)     Tracks  (%)     Tracks  (%)
-----
```

```

-----
10FA      69030    25      69036    25          1    0
10FB      69030    25      69036    25          1    0
10FC      69030    25          12     0          1    0
10FD      69030    25          12     0          1    0
10FE      69030    25          12     0          1    0
10FF      69030    25          12     0          1    0
-----
Tracks    414180    150      138120    50          6    0
}

```

Disabling and removing data devices from a pool

Support for the draining of data devices has been added to 5874 Engenuity for Symmetrix V-Max arrays. This allows data devices to be removed from thin pools nondisruptively.

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:

```
# 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:

```
# 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):
{
-----
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
-----
Tracks    207072    138120    68952      66
}

```

Disabled Devices(1):

```

{
-----
Sym      Total      Used      Free Full Device Session
Dev      Tracks     Tracks   Tracks  (%)  State  Status
-----
10EB      69024      9864      59160     14 Draining N/A
-----
Tracks    69024      9864      59160     14
}

```

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):

```

{
-----
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
-----
Tracks    207072    138120    68952     66
}

```

Disabled Devices(1):

```

{
-----
Sym      Total      Used      Free Full Device Session
Dev      Tracks     Tracks   Tracks  (%)  State  Status
-----
10EB      69024          0      69024     0 Disabled N/A
-----
Tracks    69024          0      69024     0
}

```

Device 10EB can now be removed:

```
# 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
```

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

```
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
-----
Tracks    207072     138120    68952    66
}
```

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.

```
# 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.
```

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

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

```
Symmetrix ID: 000190103334
```

S Y M M E T R I X P O O L S									
Pool Name	Ty p e	Dev Emul	Dev Config	Total Tracks	Enabled Tracks	Used Tracks	Free Tracks	Full (%)	Sta t e
HR	TH	FBA	2-Way Mir	276096	276096	40680	235416	14	Ena
Total Tracks				276096	276096	40680	235416	14	

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 metadvice from bound devices, the command will fail.

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

```
# symconfigure -sid 34 -cmd "form meta from dev 1110, config=concatenated,
count=4;" 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.

The following is the output of `symdev show` run against the newly created thin metadvice:

```
# 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                : 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     : 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)

Meta Configuration        : Concatenated
Meta Device Members (4)   :
{
-----
                        BCV  DATA                      RDF  DATA
-----
Sym   Cap   Std Inv BCV Inv Pair          R1 Inv R2 Inv Pair
Dev   (MB)   Tracks Tracks State          Tracks Tracks State

```

```

-----
--> 1110  4315      -      -  N/A      -      -  N/A
     1111  4315      -      -  N/A      -      -  N/A
     1112  4315      -      -  N/A      -      -  N/A
     1113  4315      -      -  N/A      -      -  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			
Sym	Physical	SA :P	DA :IT	Config	Attribute	Sts	Cap (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
10FC	Not Visible	?????	NA:NA	TDEV	N/Grp'd	NR	4314
10FD	Not Visible	?????	NA:NA	TDEV	N/Grp'd	NR	4314
10FE	Not Visible	?????	NA:NA	TDEV	N/Grp'd	NR	4314
10FF	Not Visible	?????	NA:NA	TDEV	N/Grp'd	NR	4314
1100	Not Visible	?????	NA:NA	TDEV	N/Grp'd	NR	4314
1101	Not Visible	?????	NA:NA	TDEV	N/Grp'd	NR	4314
1102	Not Visible	?????	NA:NA	TDEV	N/Grp'd	NR	4314
1103	Not Visible	?????	NA:NA	TDEV	N/Grp'd	NR	4314

1110	Not Visible	???:?	NA:NA	TDEV	N/Grp'd	(M)	NR	17261
1111	Not Visible	???:?	NA:NA	TDEV	N/Grp'd	(m)	NR	-
1112	Not Visible	???:?	NA:NA	TDEV	N/Grp'd	(m)	NR	-
1113	Not Visible	???:?	NA:NA	TDEV	N/Grp'd	(m)	NR	-
1114	Not Visible	???:?	NA:NA	TDEV	N/Grp'd		NR	4315
1115	Not Visible	???:?	NA:NA	TDEV	N/Grp'd		NR	4315
1116	Not Visible	???:?	NA:NA	TDEV	N/Grp'd		NR	4315
1117	Not Visible	???:?	NA:NA	TDEV	N/Grp'd		NR	4315
1118	Not Visible	???:?	NA:NA	TDEV	N/Grp'd		NR	4315
1119	Not Visible	???:?	NA:NA	TDEV	N/Grp'd		NR	4315
111A	Not Visible	???:?	NA:NA	TDEV	N/Grp'd		NR	4315
111B	Not Visible	???:?	NA:NA	TDEV	N/Grp'd		NR	4315

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.

Please 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

Device Name		Directors		Device			
Sym	Physical	SA :P	DA :IT	Config	Attribute	Sts	Cap (MB)
10F4	c2t50060482D52FC9A8d1*	09A:1	NA:NA	TDEV	N/Grp'd	RW	4314
10F5	c2t50060482D52FC9A8d1*	09A:1	NA:NA	TDEV	N/Grp'd	RW	4314
10F6	c2t50060482D52FC9A8d1*	09A:1	NA:NA	TDEV	N/Grp'd	RW	4314
10F7	c2t50060482D52FC9A8d1*	09A:1	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
10FC	Not Visible	?????	NA:NA	TDEV	N/Grp'd	NR	4314
10FD	Not Visible	?????	NA:NA	TDEV	N/Grp'd	NR	4314
10FE	Not Visible	?????	NA:NA	TDEV	N/Grp'd	NR	4314
10FF	Not Visible	?????	NA:NA	TDEV	N/Grp'd	NR	4314
1100	Not Visible	?????	NA:NA	TDEV	N/Grp'd	NR	4314
1101	Not Visible	?????	NA:NA	TDEV	N/Grp'd	NR	4314
1102	Not Visible	?????	NA:NA	TDEV	N/Grp'd	NR	4314
1103	Not Visible	?????	NA:NA	TDEV	N/Grp'd	NR	4314
1110	Not Visible	***:*	NA:NA	TDEV	N/Grp'd	(M) RW	17261
1111	Not Visible	***:*	NA:NA	TDEV	N/Grp'd	(m) RW	-
1112	Not Visible	***:*	NA:NA	TDEV	N/Grp'd	(m) RW	-
1113	Not Visible	***:*	NA:NA	TDEV	N/Grp'd	(m) RW	-
1114	Not Visible	?????	NA:NA	TDEV	N/Grp'd	NR	4315
1115	Not Visible	?????	NA:NA	TDEV	N/Grp'd	NR	4315
1116	Not Visible	?????	NA:NA	TDEV	N/Grp'd	NR	4315
1117	Not Visible	?????	NA:NA	TDEV	N/Grp'd	NR	4315
1118	Not Visible	?????	NA:NA	TDEV	N/Grp'd	NR	4315
1119	Not Visible	?????	NA:NA	TDEV	N/Grp'd	NR	4315
111A	Not Visible	?????	NA:NA	TDEV	N/Grp'd	NR	4315
111B	Not Visible	?????	NA:NA	TDEV	N/Grp'd	NR	4315

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 meta member. 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
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):

```
{
-----
                                POWERPATH  DIRECTOR  PORT  LUN
                                -----
PdevName                      Type      Type Num  Sts  VBUS TID SYMM Host
-----
Not Visible                   N/A      FA   08A:0  RW   000  00  009  N/A
Not Visible                   N/A      FA   09A:1  RW   000  00  009  N/A
}
```

Meta Configuration : Concatenated

Meta Device Members (5) :

```
{
-----
                                BCV  DATA                                RDF  DATA
                                -----
Sym   Cap   Std Inv BCV Inv Pair                                R1 Inv R2 Inv Pair
Dev   (MB)   Tracks Tracks State                                Tracks Tracks State
-----
--> 1110  4315   -    -  N/A                                -    -  N/A
    1111  4315   -    -  N/A                                -    -  N/A
    1112  4315   -    -  N/A                                -    -  N/A
    1113  4315   -    -  N/A                                -    -  N/A
    1114  4315   -    -  N/A                                -    -  N/A
-----
                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.

```

# 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 meta member is removed from a metadvice it will remain bound to the thin pool.

```
# 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         36      69009    0  Enabled
-----
Tracks    138090         60      138030    0
}
```

```
Thin Devices(2):
```

```
{
-----
Sym      Total      Pool      Pool      Pool
Dev      Tracks     Subs      Alloc Alloc      Wrtn  Wrtn
              (%)      Tracks (%)      Tracks (%)
-----
1110      276180      200         48    0         4    0
1114      69045       50         12    0         1    0
-----
Tracks    345225      250         60    0         5    0
}
```

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

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

```
# 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      Pool      Pool      Pool
Dev      Tracks     Subs      Alloc Alloc      Wrtn  Wrtn
-----
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

```

          S Y M M E T R I X   D A T A   D E V I C E S
-----
Dev  Dev  Pool      Total  Used  Full
Sym Emul Config  Type Pool Name  State  MBs   MBs   (%)
-----
10E4 FBA  2-Way Mir  TH  HR      Ena    4314  1023  23
10E5 FBA  2-Way Mir  TH  HR      Ena    4314  1020  23
10E6 FBA  2-Way Mir  TH  HR      Ena    4314   248   5
10E7 FBA  2-Way Mir  TH  HR      Ena    4314   249   5
-----
Total
MBs                                17256  2542  14

```

Legend for Pool Types:

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

Symmetrix ID: 000190103334

S Y M M E T R I X D A T A D E V I C E S									
Sym	Dev Emul	Dev Config	Pool Type	Pool Name	State	Total MBs	Used MBs	Full (%)	
10E4	FBA	2-Way Mir	TH	HR	Ena	4314	1270	29	
10E5	FBA	2-Way Mir	TH	HR	Ena	4314	1267	29	
10E6	FBA	2-Way Mir	TH	HR	Ena	4314	500	11	
10E7	FBA	2-Way Mir	TH	HR	Ena	4314	500	11	
Total									
	MBs					17256	3538	20	

Legend for Pool Types:

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

#

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 full. 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% full.

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:

```
# 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
```

S Y M M E T R I X D A T A D E V I C E S									
Sym	Dev Emul	Dev Config	Pool Type	Pool Name	State	Total Tracks	Used Tracks	Full (%)	
10E4	FBA	2-Way Mir	TH	HR	Ena	69024	20328	29	
10E5	FBA	2-Way Mir	TH	HR	Ena	69024	20280	29	
10E6	FBA	2-Way Mir	TH	HR	Ena	69024	8004	11	
10E7	FBA	2-Way Mir	TH	HR	Ena	69024	8004	11	
Total									
	Tracks					276096	56616	20	

Legend for Pool Types:

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

S Y M M E T R I X D A T A D E V I C E S								
Sym	Dev Emul	Dev Config	Pool Type	Pool Name	State	Total Tracks	Used Tracks	Full (%)
10E4	FBA	2-Way Mir	TH	HR	Ena	69024	35916	52
10E5	FBA	2-Way Mir	TH	HR	Ena	69024	36084	52
10E6	FBA	2-Way Mir	TH	HR	Ena	69024	23952	34
10E7	FBA	2-Way Mir	TH	HR	Ena	69024	23964	34
Total Tracks						276096	119916	43

Legend for Pool Types:

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

S Y M M E T R I X D A T A D E V I C E S								
Sym	Dev Emul	Dev Config	Pool Type	Pool Name	State	Total Tracks	Used Tracks	Full (%)
10E4	FBA	2-Way Mir	TH	HR	Ena	69024	54372	78
10E5	FBA	2-Way Mir	TH	HR	Ena	69024	54492	78
10E6	FBA	2-Way Mir	TH	HR	Ena	69024	42540	61
10E7	FBA	2-Way Mir	TH	HR	Ena	69024	42636	61
Total Tracks						276096	194040	70

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 virtual devices (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 RBCV's (RBCV RDF) : 0
Number of Remotely-assoc'd BCV's (Hop-2 BCV) : 0
Number of Remotely-assoc'd VDEV's (Hop-2 VDEV) : 0
Number of Remotely-assoc'd TGT's (Hop-2 TGT) : 0
```

Standard (STD) Devices (4):

LdevName	PdevName	Sym Dev	Att.	Sts	Cap (MB)

DEV001	rdmp/c2t50060482D52FC9*	10F4		RW	4314
DEV002	rdmp/c2t50060482D52FC9*	10F5		RW	4314
DEV003	rdmp/c2t50060482D52FC9*	10F6		RW	4314
DEV004	rdmp/c2t50060482D52FC9*	10F7		RW	4314
}					

VDEV Devices Locally-associated (4):

LdevName	PdevName	Sym Dev	Att.	Sts	Cap (MB)

VDEV001	N/A	1130		NR	4314
VDEV002	N/A	1131		NR	4314
VDEV003	N/A	1132		NR	4314
VDEV004	N/A	1133		NR	4314
}					

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

S Y M M E T R I X P O O L S								

Pool Name	Ty p	Dev e Emul	Dev Config	Total Tracks	Enabled Tracks	Used Tracks	Free Tracks	Full (%) Sta t

DEFAULT_POOL	SN	FBA	Unknown	0	0	0	0	0 Dis
DEFAULT_POOL	SN	3390	2-Way Mir	5659605	5659605	0	5659605	0 Ena
DEFAULT_POOL	SN	3380	Unknown	0	0	0	0	0 Dis
DEFAULT_POOL	SN	AS400	Unknown	0	0	0	0	0 Dis
thindev_snap	SN	FBA	2-Way Mir	276120	276120	0	276120	0 Ena

Total Tracks				5935725	5935725	0	5935725	0

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
```

Source Device			Target Device			State	Copy
Logical	Sym	Protected Tracks	Logical	Sym	G	Changed Tracks	SRC <=> TGT (%)
DEV001	10F4	69030	VDEV001	1130	X	0	Created 0
DEV002	10F5	69030	VDEV002	1131	X	0	Created 0
DEV003	10F6	69030	VDEV003	1132	X	0	Created 0
DEV004	10F7	69030	VDEV004	1133	X	0	Created 0
Total		-----				-----	
Track(s)		276120				0	
MB(s)		17257.5				0.0	

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
```

Source Device			Target Device			State	Copy
Logical	Sym	Protected Tracks	Logical	Sym	G	Changed Tracks	SRC <=> TGT (%)
DEV001	10F4	69030	VDEV001	1130	X	0	CopyOnWrite 0

DEV002	10F5	69030	VDEV002	1131	X	0	CopyOnWrite	0
DEV003	10F6	69030	VDEV003	1132	X	0	CopyOnWrite	0
DEV004	10F7	69030	VDEV004	1133	X	0	CopyOnWrite	0

Total	-----	-----
Track(s)	276120	0
MB(s)	17257.5	0.0

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. Other than the requirement that both the source and target device be thin, creating clones of thin devices is exactly the same as creating clones of regular devices. Thin clones 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 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. 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
...
1160 rdmp/c2t50060482D52FC* 09A:1 NA:NA TDEV N/Grp'd RW 17263
1161 rdmp/c2t50060482D52FC* 09A:1 NA:NA TDEV N/Grp'd RW 17263
1162 rdmp/c2t50060482D52FC* 09A:1 NA:NA TDEV N/Grp'd RW 17263
```

```
1163 rdm/c2t50060482D52FC* 09A:1 NA:NA TDEV N/Grp'd RW 17263
```

```
# symdg create thinclonedg
```

```
# symld -g thinclonedg addall dev -range 1160:1161 -sid 34
```

```
# symld -g thinclonedg addall dev -range 1162:1163 -sid 34 -tgt
```

```
# symdg show thinclonedg
```

```
Group Name: thinclonedg
```

```
Group Type                : REGULAR
Device Group in GNS       : No
Valid                     : Yes
Symmetrix ID              : 000190103334
Group Creation Time       : Thu Jun  5 16:01:21 2008
Vendor ID                 : EMC Corp
Application ID            : SYMCLI
```

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

```
Standard (STD) Devices (2):
```

```
{
-----
LdevName          PdevName          Sym   Dev  Att.  Sts   Cap
                    (MB)
-----
DEV001            rdm/c2t50060482D52FC9* 1160      RW   17263
DEV002            rdm/c2t50060482D52FC9* 1161      RW   17263
}
```

```
TGT Devices Locally-associated (2):
```

```
{
-----
LdevName          PdevName          Sym   Dev  Att.  Sts   Cap
                    (MB)
-----
TGT001            rdm/c2t50060482D52FC9* 1162      RW   17263
TGT002            rdm/c2t50060482D52FC9* 1163      RW   17263
}
```

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

Source Device				Target Device			State		Copy
Logical	Sym	Protected Tracks	Modified Tracks	Logical	Sym	Modified Tracks	CGDP	SRC <=> TGT	(%)
DEV001	1160	0	0	TGT001	1162	0	XXXX	PreCopy	100
DEV002	1161	0	0	TGT002	1163	0	XXXX	PreCopy	100
Total		-----	-----			-----			
Track(s)		0	0			0			
MB(s)		0.0	0.0			0.0			

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 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
DG's Symmetrix ID : 000190103334

Source Device				Target Device				State	Copy
Logical	Sym	Protected Tracks	Modified Tracks	Logical	Sym	Modified Tracks	CGDP	SRC <=> TGT	(%)
DEV001	1160	0	0	TGT001	1162	0	XXX.	Copied	100
DEV002	1161	0	0	TGT002	1163	0	XXX.	Copied	100
Total									
Track(s)		0	0			0			
MB(s)		0.0	0.0			0.0			

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 device be 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:

```
# symcfg list
```

S Y M M E T R I X

SymmID	Attachment	Model	Mcode Version	Cache Size (MB)	Num Phys Devices	Num Symm Devices
000190103334	Local	DMX4-24	5773	65536	101	4526
000190103232	Remote	DMX4-24	5773	65536	00	7335

```
# sympd list -sid 34
```

```
Symmetrix ID: 000190103334
```

Device Name		Directors				Device			
Physical	Sym	SA	:P	DA	:IT	Config	Attribute	Sts	Cap (MB)
...									
c2t50060482D52FC9A8d7*	1164	09A:1	NA:NA	TDEV		N/Grp'd	RW	17263	
c2t50060482D52FC9A8d7*	1165	09A:1	NA:NA	TDEV		N/Grp'd	RW	17263	
c2t50060482D52FC9A8d7*	1166	09A:1	NA:NA	TDEV		N/Grp'd	RW	17263	
c2t50060482D52FC9A8d7*	1167	09A:1	NA:NA	TDEV		N/Grp'd	RW	17263	
...									

```
# symdev list -sid 32 -tdev
```

```
Symmetrix ID: 000190103232
```

Device Name		Directors				Device			
Sym	Physical	SA	:P	DA	:IT	Config	Attribute	Sts	Cap (MB)
...									
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	
...									

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
```

```
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 of configuration changes.....Queued.
COMMIT requesting required resources.....Obtained.
```

```

Step 002 of 027 steps.....Executing.
Step 008 of 027 steps.....Executing.
Step 019 of 027 steps.....Executing.
Step 022 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 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
```

```
# Thin Devices for SRDF
#
```

```
# Symm 34 (Local)      Symm 32 (Remote)
#
```

```
1164      0DF2
1165      0DF3
1166      0DF4
1167      0DF5
```

```
# symrdf -file 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.
Devices: 0DF2-0DF5 in (3334,006)..... Marked.
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.
Devices: 1164-1167 in (3334,006)..... Marked.
Mark target device(s) in (3334,006) for full copy from source...Done.
Merge track tables between source and target in (3334,006).....Started.
Devices: 1164-1167 in (3334,006)..... Merged.
Merge track tables between source and target in (3334,006).....Done.
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)
```

Source (R1) View					Target (R2) View					MODES	
-----					-----					-----	
		ST			LI	ST					
Standard		A			N	A					
Logical		T	R1 Inv	R2 Inv	K	T	R1 Inv	R2 Inv	RDF Pair		
Device	Dev	E	Tracks	Tracks	S	Dev	E	Tracks	Tracks	MDA	STATE
-----					-----					-----	
N/A	1164	RW	0	160030	RW	0DF2	WD	0	10	S..	SyncInProg
N/A	1165	RW	0	151882	RW	0DF3	WD	0	10	S..	SyncInProg
N/A	1166	RW	0	141634	RW	0DF4	WD	0	10	S..	SyncInProg
N/A	1167	RW	0	128230	RW	0DF5	WD	0	10	S..	SyncInProg
Total											
		-----					-----				
Track(s)		0					0			40	
MB(s)		0.0					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 -rdfig 6
```

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

Source (R1) View					Target (R2) View				MODES	
-----					-----				-----	
Standard	ST		LI		Standard	ST				
Logical	A		N		Logical	A				
Device	T	R1 Inv	R2 Inv	K	Device	T	R1 Inv	R2 Inv	MDA	RDF Pair
Dev	E	Tracks	Tracks	S	Dev	E	Tracks	Tracks		STATE
-----					-----				-----	
N/A	1164	RW	0	0	RW	0DF2	WD	0	0	S.. Synchronized
N/A	1165	RW	0	0	RW	0DF3	WD	0	0	S.. Synchronized
N/A	1166	RW	0	0	RW	0DF4	WD	0	0	S.. Synchronized
N/A	1167	RW	0	0	RW	0DF5	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 -rdfig 6 -nop
```

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

```
Suspend 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 -rdfig 6
```

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

Source (R1) View					Target (R2) View				MODES	
-----					-----				-----	
Standard	ST		LI		Standard	ST				
Logical	A		N		Logical	A				
Device	T	R1 Inv	R2 Inv	K	Device	T	R1 Inv	R2 Inv	MDA	RDF Pair
Dev	E	Tracks	Tracks	S	Dev	E	Tracks	Tracks		STATE
-----					-----				-----	
N/A	1164	RW	0	0	NR	0DF2	RW	0	0	S.. Split

N/A	1165	RW	0	0	NR	ODF3	RW	0	0	S..	Split
N/A	1166	RW	0	0	NR	ODF4	RW	0	0	S..	Split
N/A	1167	RW	0	0	NR	ODF5	RW	0	0	S..	Split

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

Virtual Provisioning with Symmetrix Management Console

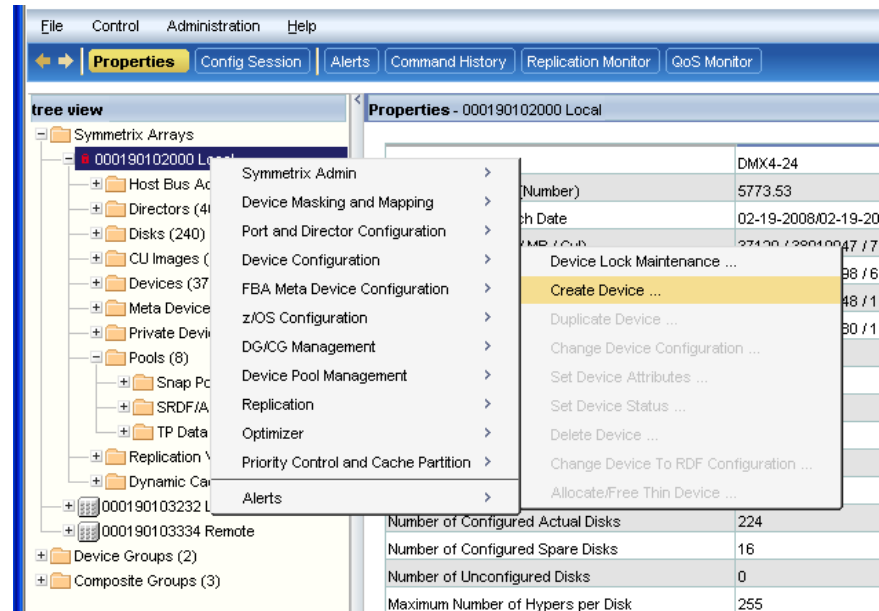
The following are examples of how to configure, update, and monitor a Symmetrix Virtual Provisioning environment using Symmetrix Management Console (SMC).

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

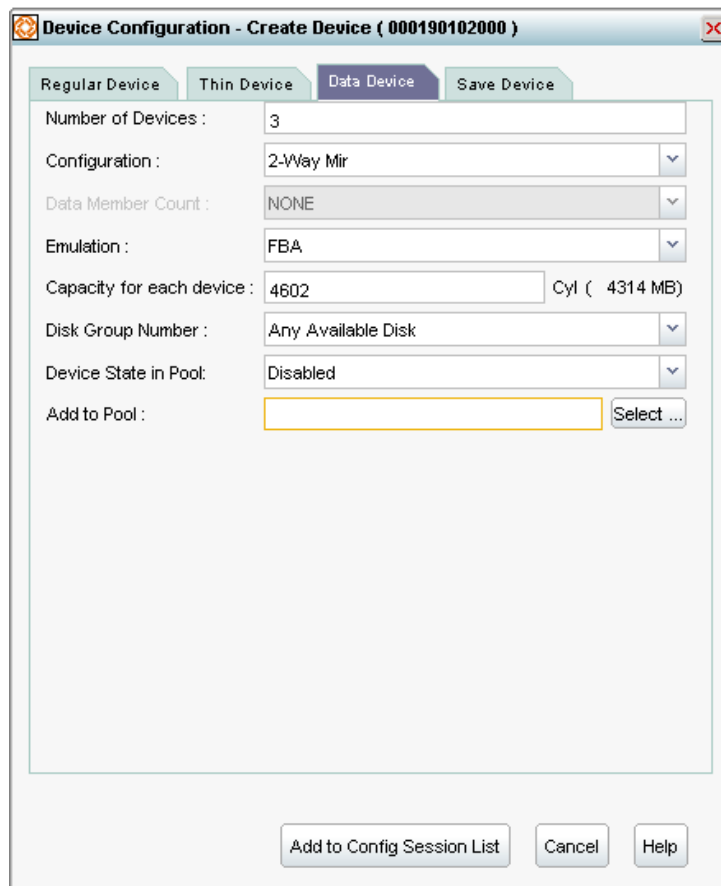
Configuring and viewing data devices and pools

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.



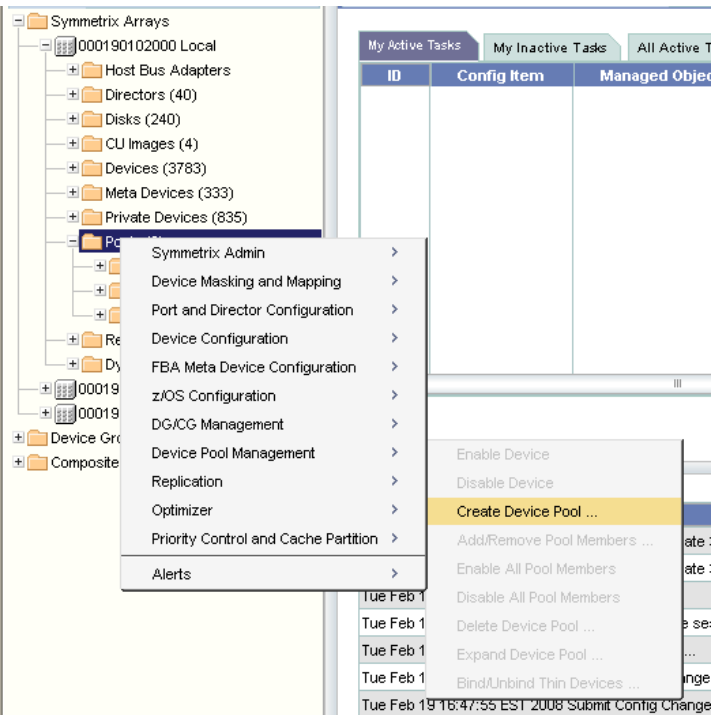
The image shows a 'Device Configuration - Create Device (000190102000)' dialog box. It has four tabs: 'Regular Device', 'Thin Device', 'Data Device' (which is selected), and 'Save Device'. The 'Data Device' tab contains the following fields:

- Number of Devices : 3
- Configuration : 2-Way Mir (dropdown)
- Data Member Count : NONE (dropdown)
- Emulation : FBA (dropdown)
- Capacity for each device : 4602 Cyl (4314 MB)
- Disk Group Number : Any Available Disk (dropdown)
- Device State in Pool: Disabled (dropdown)
- Add to Pool : [empty text box] Select ...

At the bottom of the dialog are three buttons: 'Add to Config Session List', 'Cancel', and 'Help'.

Add the task to the configuration session list and commit -nop 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:



Device Pool Management - Create Device Pool (000190102000)

Pool Name :

Pool Type : ☐ Max Subscription % (0 -- 65534) :

Emulation : ☐ Enable New Pool Member

Filter Available list

Dev Config : Data Member Count :

Emulation : Cap (MB) :

Pool	Dev	State	Cap
N/A	0CAC	Disabled	
N/A	0C8A	Disabled	
N/A	0ED2	Disabled	
N/A	0E71	Disabled	
N/A	0D65	Disabled	

Page 1 of 3 Items 1 to 250 of 511

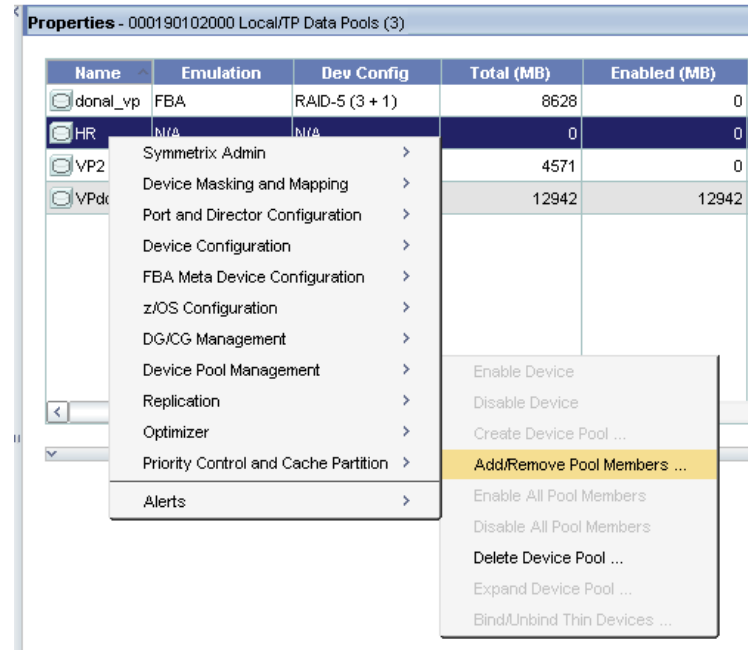
Add Add All Remove Remove All

Pool	Dev	State	Cap (M
------	-----	-------	--------

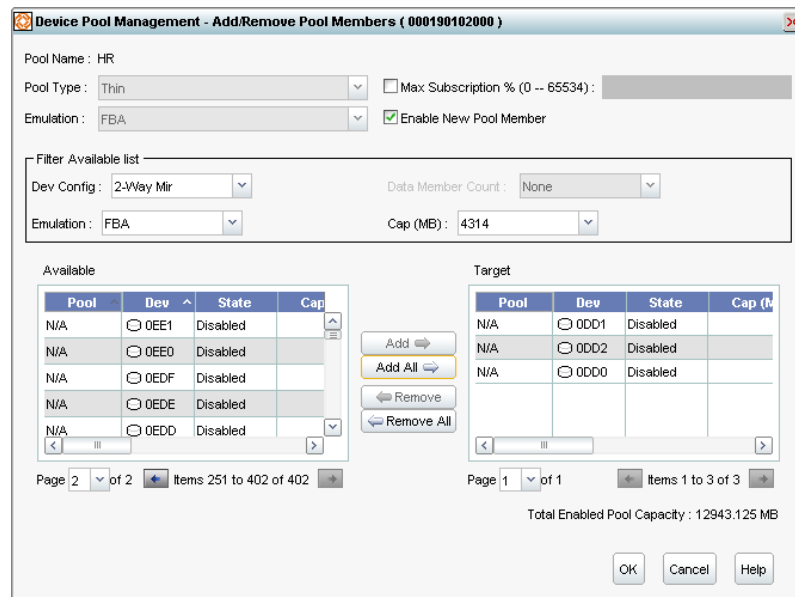
Total Enabled Pool Capacity : 0 MB

OK Cancel Help

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:



Adding additional data devices to a pool

Additional data devices can be added to a pool anytime 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. See Virtual Provisioning best practices documentation for 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 (000190102000)

Regular Device Thin Device **Data Device** Save Device

Number of Devices : 3

Configuration : 2-Way Mir

Data Member Count : NONE

Emulation : FBA

Capacity for each device : 4602 Cyl (4314 MB)

Disk Group Number : Any Available Disk

Device State in Pool: Enabled

Add to Pool : HR Select ...

Add to Config Session List Cancel Help

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

tree view

- Symmetrix Arrays
 - 000190102000 Local
 - Host Bus Adapters
 - Directors (40)
 - Disks (240)
 - CU Images (4)
 - Devices (3783)
 - Meta Devices (333)
 - Private Devices (838)
 - Pools (9)
 - Snap Pools (1)
 - SRDF/A DSE Pools (4)
 - TP Data Pools (4)**
 - Replication Views
 - Dynamic Cache Partitions
 - 000190103232 Local
 - 000190103334 Remote
 - Device Groups (2)
 - Composite Groups (3)

Properties - 000190102000 Local/TP Data Pools (4)

Name	Emulation	Dev Config	Total (MB)	Enabled (MB)	Alloc (MB)	Alloc %
dona_vp	FBA	RAID-5 (3 + 1)	8628	0	0	0
HR	FBA	2-Way Mir	25884	25884	0	0
VP2	FBA	RAID-5 (3 + 1)	4571	0	0	0
VPdona	FBA	RAID-5 (3 + 1)	12942	12942	0	0

General DATA Devices (6)

Symmetrix ID	000190102000
Pool Name	HR
Pool Type	TP Data
Pool Emulation	FBA
Dev Configuration	2-Way Mir
Number of Devices	6
Disabled Devices	0
Enabled Devices	6
Total Capacity (MB)	25884
Total Used (MB)	0
Total Free (MB)	25884
Pool State	Enabled
Maximum Subscription %	None

Applet Nexaweb Applet started Internet

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:

tree view

- Symmetrix Arrays
 - 000190102000 Local
 - Host Bus Adapters
 - Directors (40)
 - Disks (240)
 - CU Images (4)
 - Devices (3783)
 - Meta Devices (333)
 - Private Devices (838)
 - Pools (9)
 - Snap Pools (1)
 - SRDF/A DSE Pools (4)
 - Thin Pools (4)**
 - Replication Views
 - Dynamic Cache Partitions
 - 000190103232 Local
 - 000190103334 Remote
 - Device Groups (2)
 - Composite Groups (3)

Properties - 000190102000 Local/Thin Pools (4)

Name	Emulation	Dev Config	Total (MB)	Enabled (MB)	Alloc (MB)	Alloc %	Subs %
dona_vp	FBA	RAID-5 (3 + 1)	8628	0	0	0	
HR	FBA	2-Way Mir	25884	25884	0	0	
VP2	FBA	RAID-5 (3 + 1)	4571	0	0	0	
VPdona	FBA	RAID-5 (3 + 1)	12942	12942	0	0	

General DATA Devices (6)

Dev	Emulation	Dev Config	State	% Used	Used (MB)	Used Tracks	Free (MB)
0DD0	FBA	2-Way Mir	Enabled	0.0	0.0	0	4314.0
0DD1	FBA	2-Way Mir	Enabled	0.0	0.0	0	4314.0
0DD2	FBA	2-Way Mir	Enabled	0.0	0.0	0	4314.0
0DD3	FBA	2-Way Mir	Enabled	0.0	0.0	0	4314.0
0DD4	FBA	2-Way Mir	Enabled	0.0	0.0	0	4314.0
0DD5	FBA	2-Way Mir	Enabled	0.0	0.0	0	4314.0

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:

Device Configuration - Create Device (000190102000)

Regular Device | **Thin Device** | Data Device | Save Device

Number of Devices : 3

Configuration : TDEV

Emulation : FBA

Capacity for each device : 4602 Cyl (4314 MB)

SSD : 1 Select ...

RDF Group Number : Select ...

Remote Configuration : TDEV

Remote SSD : Select ...

Dynamic Capability : NONE

Bind to Pool : Select ...

Remote Pool : Select ...

Preallocated Capacity for each device : Cyl (MB)

Meta Member Size : 0 Cyl

Meta Member Config : N/A

Add to Config Session List Cancel Help

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

Properties - 000190102000 Local/Thin Pools (4)

Name	Emulation	Dev Config	Total (MB)	Enabled (MB)
donal_vp	FBA	RAID-5 (3 + 1)	8628	0
HR	FBA	2-Way Mir	25884	25884
VP2			4571	0
VPdonal			12942	12942

Symmetrix Admin >
 Device Masking and Mapping >
 Port and Director Configuration >
 Device Configuration >
 FBA Meta Device Configuration >
 z/OS Configuration >
 DG/CG Management >
 Device Pool Management >
 Replication >
 Optimizer >
 Priority Control and Cache Partition >
 Alerts >

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 ...
 Bind/Unbind Thin Devices ...

General
 DATA Devices (6)
 Symmetrix ID: 000190102000
 Pool Name: HR
 Pool Type: TP Data
 Pool Emulation: FBA

Device Pool Management - Bind/Unbind Thin Devices (000190102000)

Pool Name : HR Pool Type : Data Device Pool Emulation : FBA
 Preallocated Capacity for each new device : Cyl (MB) Total Enabled Pool Capacity : 25886.25 MB

Filter Available list
 Dev Config : TDEV Emulation : FBA Cap (MB) : 4314 Res : Unres

Available			Target		
Device	Cap (MB)	Dev Config	Device	Cap (MB)	Dev Config
OB50	4314	TDEV	ODD6	4314	TDEV
OB51	4314	TDEV	ODD7	4314	TDEV
OB52	4314	TDEV	ODD8	4314	TDEV
OB53	4314	TDEV			
OB54	4314	TDEV			

Add Add All Remove Remove All

Page 1 of 2 Items 1 to 250 of 280 Page 1 of 1 Items 1 to 3 of 3

Max Subscription % : None Current subscription % : 50.00 Total Bound Capacity : 12943.12 MB

☐ Reserve Comments : ☐ Expiration : Never

Add to Config Session List Cancel Help

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:

Properties - 000190102000 Local/Thin Pools (4)

Name	Emulation	Dev Config	Total (MB)	Enabled (MB)	Alloc (MB)	Alloc %	Subs %
donal_vp	FBA	RAID-5 (3 + 1)	8628	0	0	0	
HR	FBA	2-Way Mir	25884	25884	2	0	5
VP2	FBA	RAID-5 (3 + 1)	4571	0	0	0	
VPdonal	FBA	RAID-5 (3 + 1)	12942	12942	0	0	

General DATA Devices (6) Bound Thin Devices (3)

Dev	Pool	Total Trks	Subscribed %	Allocated Trks	Allocated %	Trks WR	Written %
0DD6	HR	69030	17	12	0	1	
0DD7	HR	69030	17	12	0	1	
0DD8	HR	69030	17	12	0	1	

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.

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.

Device Configuration - Create Device (000190102000)

Regular Device | **Thin Device** | Data Device | Save Device

Number of Devices : 3

Configuration : TDEV

Emulation : FBA

Capacity for each device : 4602 Cyl (4314 MB)

SSD : 1 Select ...

RDF Group Number : Select ...

Remote Configuration : TDEV

Remote SSD : Select ...

Dynamic Capability : NONE

Bind to Pool : HR Select ...

Remote Pool : Select ...

Preallocated Capacity for each device : 4602 Cyl (4314 MB)

Meta Member Size : 0 Cyl

Meta Member Config : N/A

Add to Config Session List Cancel Help

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% of the thin pool.

Properties - 000190102000 Local/Thin Pools (3)

Name	Emulation	Dev Config	Total (MB)	Enabled (MB)	Alloc (MB)	Alloc %	Subs %
dona_vp	FBA	RAID-5 (3 + 1)	8628	0	0	0	
HR	FBA	2-Way Mir	38826	38826	12946	33	6
VP2	FBA	RAID-5 (3 + 1)	4571	0	0	0	

General DATA Devices (9) Bound Thin Devices (8)

Dev	Pool	Total Trks	Subscribed %	Allocated Trks	Allocated %	Trks WR	Written %
DD6	HR	69030	11	12	0	1	
DD7	HR	69030	11	12	0	1	
DD8	HR	69030	11	12	0	1	
DDC	HR	69030	11	69036	11	1	
DDD	HR	69030	11	69036	11	1	
DDDE	HR	69030	11	69036	11	1	

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:

Properties - 000190102000 Local/Thin Pools (5)							
Name	Emulation	Dev Config	Total (MB)	Enabled (MB)	Alloc (MB)	Alloc %	Subs %
dona1_vp	FBA	RAID-5 (3 + 1)	8628	0	0	0	
META	FBA	2-Way Mir	4647	0	0	0	
PRODD8	FBA	2-Way Mir	17256	17256	8631	50	10
SALES	FBA	2-Way Mir	21570	12942	8865	41	16
VP2	FBA	RAID-5 (3 + 1)	4571	0	0	0	

General DATA Devices (4) Bound Thin Devices (4)							
Dev	Pool	Total Trks	Subscribed %	Allocated Trks	Allocated %	Trks WR	Written %
0DD7	PRODD8	69030	25	69036	25	1	
0DD8	PRODD8	69030	25	69036	25	1	
0DDC	PRODD8	69030	25	7428	3	1	
0DDD	PRODD8	69030	25	12	0	1	

Properties - 000190102000 Local/Thin Pools (5)							
Name	Emulation	Dev Config	Total (MB)	Enabled (MB)	Alloc (MB)	Alloc %	
dona1_vp	FBA	RAID-5 (3 + 1)	8628	0	0	0	
META	FBA	2-Way Mir	4647	0	0	0	
PRODD8	FBA	2-Way Mir	17256	17256	8631	50	
SALES	FBA	2-Way Mir	21570	12942	8865	41	
VP2	FBA	RAID-5 (3 + 1)	4571	0	0	0	

General DATA Devices (4) Bound Thin Devices (4)							
Dev	Pool	Total	Subscribed %	Allocated Trks	Allocated %	Trks WR	Written %
0DD7	PRODD8	69030	25	69036	25	1	
0DD8	PRODD8	69030	25	69036	25	1	
0DDC	PRODD8	69030	25	7428	3	1	
0DDD	PRODD8	69030	25	12	0	1	

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):

Command History Replication Monitor QoS Monitor

Properties - 000190102000 Local/Thin Pools (5)

Name	Emulation	Dev Config	Total
dona1_vp	FBA	RAID-5 (3 + 1)	
META	FBA	2-Way Mir	
PRODDb	FBA	2-Way Mir	
SALES	FBA	2-Way Mir	
VP2	FBA	RAID-5 (3 + 1)	

Device Configuration - Allocate/Free Thin Device (000190102000)

Device(s) : 0DDD Show Current Values ...

☒ Allocate ☐ Free ☐ Full Device

Start Cylinder : 0

End Cylinder : 2302

☐ Reserve Comments : Expiration : Never

Add to Config Session List Cancel Help

General DATA Devices (4) Bound Thin Devices (4)

Dev	Pool	Total Trks	Subscribed %	Allocated Trks	Allocated %	Trks WR	Written %
0DD7	PRODDb	69030	25	69036	25	1	
0DD8	PRODDb	69030	25	69036	25	1	
0DDC	PRODDb	69030	25	7428	3	1	
0DDD	PRODDb	69030	25	12	0	1	

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

Properties - 000190102000 Local/Thin Pools (5)

Name	Emulation	Dev Config	Total (MB)	Enabled (MB)	Alloc (MB)	Alloc %	Subs %
dona1_vp	FBA	RAID-5 (3 + 1)	8628	0	0	0	
META	FBA	2-Way Mir	4647	0	0	0	
PRODDb	FBA	2-Way Mir	17256	17256	10790	62	100
SALES	FBA	2-Way Mir	21570	12942	8865	41	16
VP2	FBA	RAID-5 (3 + 1)	4571	0	0	0	

General DATA Devices (4) Bound Thin Devices (4)

Dev	Pool	Total Trks	Subscribed %	Allocated Trks	Allocated %	Trks WR	Written %
0DD7	PRODDb	69030	25	69036	25	1	
0DD8	PRODDb	69030	25	69036	25	1	
0DDC	PRODDb	69030	25	7428	3	1	
0DDD	PRODDb	69030	25	34560	13	1	

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 Configuration - Allocate/Free Thin Device (000190102000)

Device(s) : 0DD7 Show Current Values ...

☐ Allocate ☒ Free ☒ Full Device

Start Cylinder :

End Cylinder :

☐ Reserve Comments : Expiration : Never

Add to Config Session List Cancel Help

Properties - 000190102000 Local/Thin Pools (5)

Name	Emulation	Dev Config	Total
dona1_vp	FBA	RAID-5 (3 + 1)	
META	FBA	2-Way Mir	
PRODD8	FBA	2-Way Mir	
SALES	FBA	2-Way Mir	
VP2	FBA	RAID-5 (3 + 1)	

General DATA Devices (4) Bound Thin Devices (4)

Dev	Pool	Total Trks	Subscribed %	Allocated Trks	Allocated %	Trks WR	Written %
0DD7	PRODD8	69030	25	69036	25	1	
0DD8	PRODD8	69030	25	69036	25	1	
0DDC	PRODD8	69030	25	7428	3	1	
0DDD	PRODD8	69030	25	34560	13	1	

Device DD7 no longer has any allocated tracks:

Properties - 000190102000 Local/Thin Pools (5)

Name	Emulation	Dev Config	Total (MB)	Enabled (MB)	Alloc (MB)	Alloc %	Subs %
dona1_vp	FBA	RAID-5 (3 + 1)	8628	0	0	0	
META	FBA	2-Way Mir	4647	0	0	0	
PRODD8	FBA	2-Way Mir	17256	17256	6475	37	100
SALES	FBA	2-Way Mir	21570	12942	8865	41	16
VP2	FBA	RAID-5 (3 + 1)	4571	0	0	0	

General DATA Devices (4) Bound Thin Devices (4)

Dev	Pool	Total Trks	Subscribed %	Allocated Trks	Allocated %	Trks WR	Written %
0DD7	PRODD8	69030	25	0	0	0	
0DD8	PRODD8	69030	25	69036	25	1	
0DDC	PRODD8	69030	25	7428	3	1	
0DDD	PRODD8	69030	25	34560	13	1	

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% for the HR pool, meaning that the total amount of space for bound thin devices cannot exceed 150% of the space provided by the data devices:

Device Pool Management - Add/Remove Pool Members (000190102000)

Pool Name : HR

Pool Type : Thin ☒ Max Subscription % (0 -- 65534) : 150

Emulation : FBA ☐ Enable New Pool Member

Filter Available list

Dev Config : 2-Way Mir Data Member Count : None

Emulation : FBA Cap (MB) : None

Pool	Dev	State	Cap
N/A	0ED7	Disabled	
N/A	0C6F	Disabled	
N/A	0C94	Disabled	
N/A	0D8B	Disabled	
N/A	0D55	Disabled	

Page 1 of 2 Items 1 to 250 of 434

Pool	Dev	State	Cap
HR	0DD0	Enabled	
HR	0DD1	Enabled	
HR	0DD2	Enabled	
HR	0DD3	Enabled	
HR	0DD4	Enabled	

Page 1 of 1 Items 1 to 9 of 9

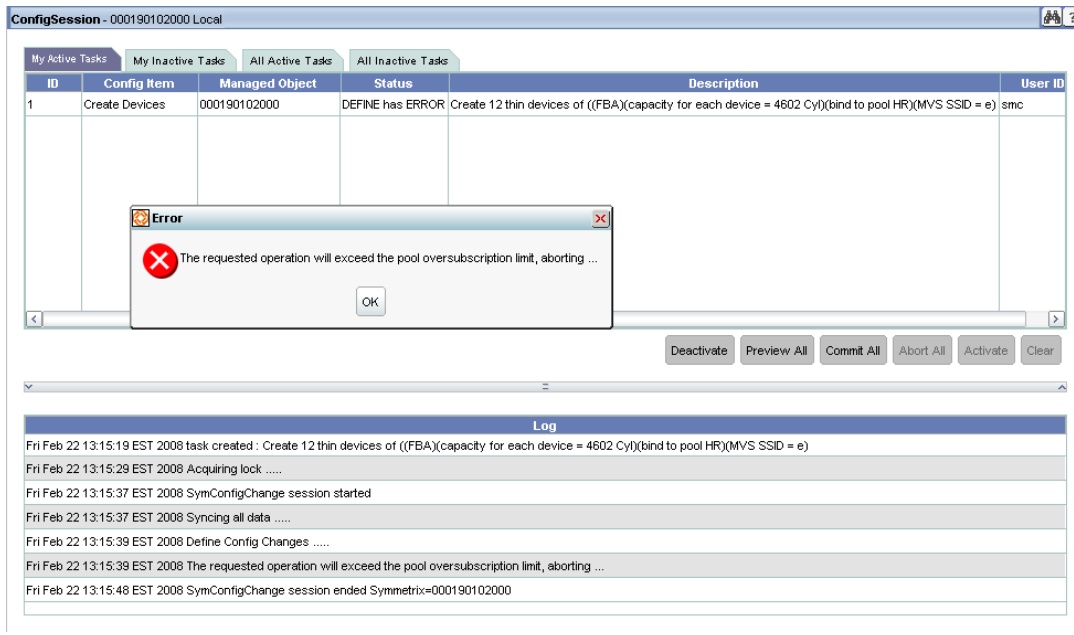
Total Enabled Pool Capacity : 36829.375 MB

OK Cancel Help

The properties of the HR pool now show that the oversubscription limit has been set at 150%:

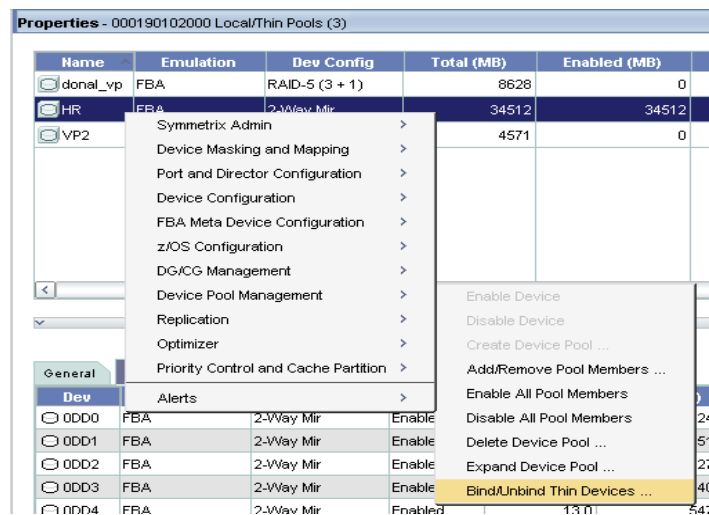
[illegible]

The oversubscription limit will prevent the user from adding thin devices that push the oversubscription ratio past 150%. 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:



Device Pool Management - Bind/Unbind Thin Devices (000190102000)

Pool Name : HR Pool Type : Data Device Pool Emulation : FBA

Preallocated Capacity for each new device : Cyl (MB) Total Enabled Pool Capacity : 38829.38 MB

Filter Available list

Dev Config : TDEV Emulation : FBA Cap (MB) : None Res : Unres

Device	Cap (MB)	Dev Config
OB50	4314	TDEV
OB51	4314	TDEV
OB52	4314	TDEV
OB53	4314	TDEV
OB54	4314	TDEV

Available

Target

Device	Cap (MB)	Dev Config
ODD7	4314	TDEV
ODD8	4314	TDEV
ODDC	4314	TDEV
ODDD	4314	TDEV
ODDE	4314	TDEV

Target

Page 1 of 2 Items 1 to 250 of 293 Page 1 of 1 Items 1 to 6 of 6

Max Subscription % : 150 Current subscription % : 66.67 Total Bound Capacity : 25886.25 MB

☐ Reserve Comments : ☐ Expiration : Never

Add to Config Session List Cancel Help

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 5874 Engenuity for Symmetrix V-Max arrays. This allows data devices to be removed from thin pools nondisruptively.

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:

Properties - 000190102000 Local/Thin Pools (3)

Name	Emulation	Dev Config	Total (MB)	Enabled (MB)	Alloc (MB)
donal_vp	FBA	RAID-5 (3 + 1)	8628	0	
HR	FBA	2-Way Mir	38826	38826	4312
VP2	FBA	RAID-5 (3 + 1)	4571	0	

General DATA Devices (3)

Dev	Emulation
0DD0	FBA
0DD1	FBA
0DD2	FBA
0DD3	FBA
0DD4	FBA
0DD5	FBA
0DD9	FBA
0DDA	FBA
0DDB	FBA

Symmetrix Admin >
 Device Masking and Mapping >
 Port and Director Configuration >
 Device Configuration >
 FBA Meta Device Configuration >
 z/OS Configuration >
 DG/CG Management >
 Device Pool Management >
 Replication >
 Optimizer >
 Priority Control and Cache Partition >
 Alerts >

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 ...
 Bind/Unbind Thin Devices ...

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:

General DATA Devices (3)		Bound Thin Devices (4)							
Dev	Emulation	Dev Config	State	% Used	Used (MB)	Used Tracks	Free (MB)	Free Tracks	Session Status
0DD0	FBA	2-Way Mir	Enabled	12.0	524.0	8376	3791.0	60648	Active
0DD1	FBA	2-Way Mir	Enabled	13.0	551.0	8808	3764.0	60216	Active
0DD2	FBA	2-Way Mir	Enabled	12.0	527.0	8424	3788.0	60600	Active
0DD3	FBA	2-Way Mir	Enabled	13.0	540.0	8640	3774.0	60384	Active
0DD4	FBA	2-Way Mir	Enabled	13.0	547.0	8748	3767.0	60276	Active
0DD5	FBA	2-Way Mir	Enabled	13.0	542.0	8676	3772.0	60348	Active
0DD9	FBA	2-Way Mir	Enabled	13.0	542.0	8676	3772.0	60348	Active
0DDA	FBA	2-Way Mir	Enabled	13.0	545.0	8724	3769.0	60300	Active
0DDB	FBA	2-Way Mir	Disabled	0.0	0.0	0	4314.0	69024	Inactive

Device DDB can now be removed:

Properties - 000190102000 Local/Thin Pools (3)

Name	Emulation	Dev Config	Total (MB)	Enabled (MB)
dona_vp	FBA	RAID-5 (3 + 1)	8628	0
HR			38826	38826
VP2			4571	0

Context menu for HR:

- Symmetrix Admin >
- Device Masking and Mapping >
- Port and Director Configuration >
- Device Configuration >
- FBA Meta Device Configuration >
- z/OS Configuration >
- DG/CG Management >
- Device Pool Management >
- Replication >
- Optimizer >
- Priority Control and Cache Partition >
- Alerts >
- 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 ...
- Bind/Unbind Thin Devices ...

General tab:

Symmetrix ID	000190102000
Pool Name	HR
Pool Type	TP Data
Pool Emulation	FBA

Device Pool Management - Add/Remove Pool Members (000190102000)

Pool Name : HR

Pool Type : Thin ☒ Max Subscription % (0 -- 65534) : 150

Emulation : FBA ☐ Enable New Pool Member

Filter Available list

Dev Config : 2-Way Mir Data Member Count : None

Emulation : FBA Cap (MB) : None

Pool	Dev	State	Cap
N/A	0ED7	Disabled	
N/A	0C6F	Disabled	
N/A	0C94	Disabled	
N/A	0D8B	Disabled	
N/A	0D55	Disabled	

Page 1 of 2 Items 1 to 250 of 434

Pool	Dev	State	Cap
HR	0DD4	Enabled	
HR	0DD5	Enabled	
HR	0DD9	Enabled	
HR	0DDA	Enabled	
HR	0DDB	Disabled	

Page 1 of 1 Items 1 to 9 of 9

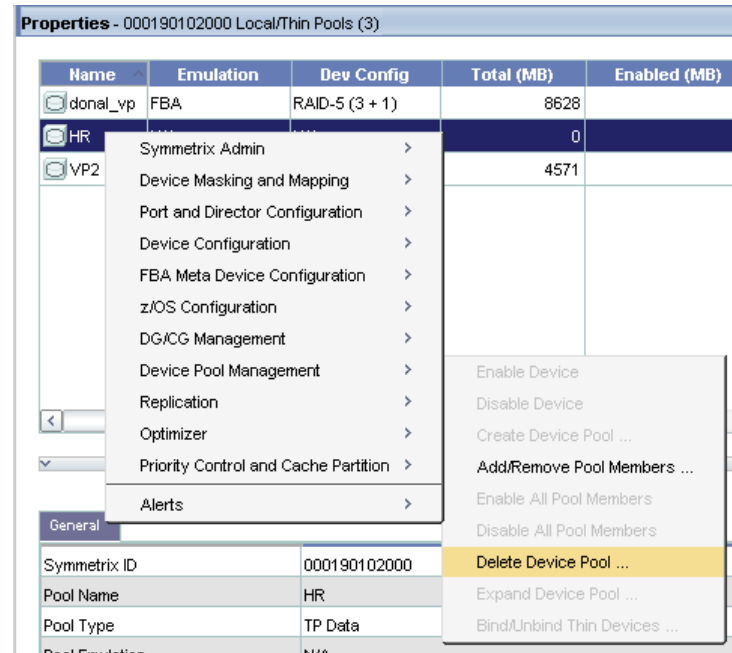
Total Enabled Pool Capacity : 34515.0 MB

OK Cancel Help

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.

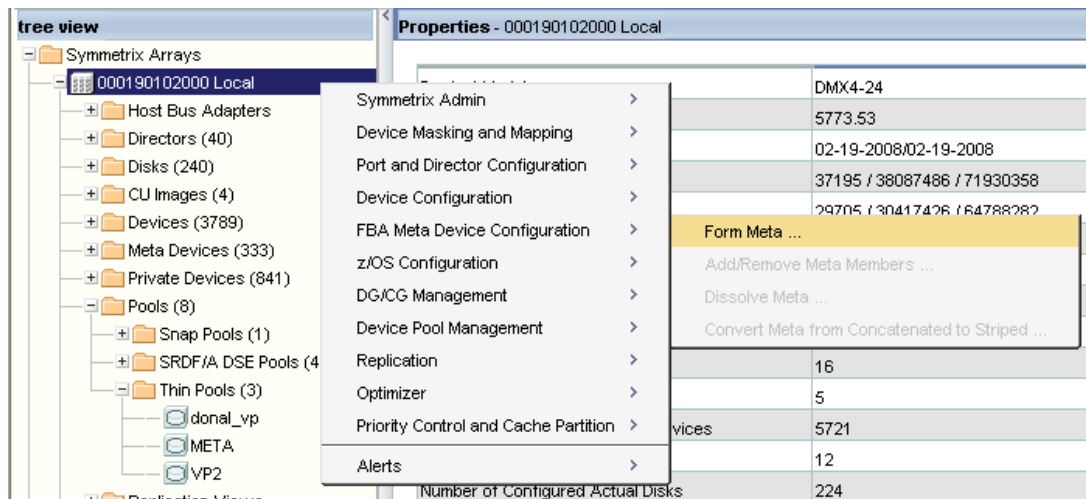


Configuring thin metadevices

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:



FBA Meta Device Configuration - Form FBA Meta (000190102000)

Device Type: Meta Config: Stripe Size (cyl / blocks) :

☐ Auto Select Number of Metas : Number of Meta Members including Head :

Filter Available list

Dev Config : Data Member Count : Emulation : Cap (MB) : Res :

Available Unmapped Devices				Meta Members			
Dev	Cap (MB)	Dev Config	Res	Dev	Cap (MB)	Dev Config	Res
0B51	4314	TDEV		0EF2	4314	TDEV	
0B52	4314	TDEV		0EF3	4314	TDEV	
0B53	4314	TDEV		0EF4	4314	TDEV	
0B54	4314	TDEV		0EF5	4314	TDEV	
0B55	4314	TDEV					

Page 1 of 2 Items 1 to 250 of 258

Page 1 of 1 Items 1 to 4 of 4

Meta Head :

☐ Reserve Comments : Expiration :

After the meta has been created it can then be bound to the pool in the same way as nonmetadevices:

Device Pool Management - Bind/Unbind Thin Devices (000190102000)

Pool Name : META Pool Type : Data Device Pool Emulation : FBA

Preallocated Capacity for each new device : Cyl (MB) Total Enabled Pool Capacity : 34515.00 MB

Filter Available list

Dev Config : TDEV Emulation : FBA Cap (MB) : None Res : Unres

Device	Cap (MB)	Dev Config
0B50	4314	TDEV
0B51	4314	TDEV
0B52	4314	TDEV
0B53	4314	TDEV
0B54	4314	TDEV

Page 1 of 2 Items 1 to 250 of 295

Max Subscription % : 200 Current subscription % : 50.00 Total Bound Capacity : 17257.50 MB

☐ Reserve Comments : Expiration : Never

Add to Config Session List Cancel Help

Meta information is now available in a Meta Members tab in Properties:

Properties - 000190102000 Local/META

Dev	Pool	Total Trks	Subscribed %	Allocated Trks	Allocated %	Trks WR	Written %
0EF2	META	276120	50	48	0	4	

Dev	Dev Config	Cap (MB/Cyl)	Meta	Meta Config	Stripe Size (Cyl/Blks)	Stripe Size (Cyl/Tr)
0EF3	TDEV	4314 / 4602	Member	Concatenated	N/A	N/A
0EF4	TDEV	4314 / 4602	Member	Concatenated	N/A	N/A
0EF5	TDEV	4314 / 4602	Member	Concatenated	N/A	N/A

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

Like metadvice made up of regular Symmetrix volumes, thin metadvice can also be expanded.

Properties - 000190102000 Local/META

General DATA Devices (8) Bound Thin Devices (1)

Dev	Pool	Total Trks	Subscribed %	Allocated Trks	Allocated %
0EF2			50	48	

- Symmetrix Admin >
- Device Masking and Mapping >
- Port and Director Configuration >
- Device Configuration >
- FBA Meta Device Configuration >
- z/OS Configuration >
- DG/CG Management >
- Device Pool Management >
- Replication >
- Optimizer >
- Priority Control and Cache Partition >
- Alerts >

Form Meta ...

Add/Remove Meta Members ...

Dissolve Meta ...

Convert Meta from Concatenated to Striped ...

General Meta Members (3) Back End Paths (2) TDev Info (1)

FBA Meta Device Configuration - Add/Remove Meta Members (000190102000)

Device Type: Regular

Meta Config: Concatenated

Filter Available list

Dev Config: TDEV Data Member Count: None

Emulation: FBA Cap (MB): 4314 Res: Unres

Available Unmapped Devices

Dev	Cap (MB)	Dev Config	R
0B51	4314	TDEV	
0B52	4314	TDEV	
0B53	4314	TDEV	
0B54	4314	TDEV	
0B55	4314	TDEV	
0B56	4314	TDEV	

Meta Members

Dev	Cap (MB)	Dev Config	Res
0EF5	4314	TDEV	
0EF4	4314	TDEV	
0EF3	4314	TDEV	
0EF2	17258	TDEV	
0EF6	4314	TDEV	

Page 1 of 2 Items 1 to 257 of 257

Page 1 of 1 Items 1 to 5 of 5

Meta Head: 0EF2

☐ Reserve Comments: Expiration: Never

Add to Config Session List Cancel Help

The configuration change can now be run. No further action is required to read/write enable the new meta member. Because it is now part of a metadvice that is bound to a thin pool, it is bound to the same pool.

Device EF2 is now a five-member meta:

Properties - 000190102000 Local/META

DATA Devices (8)							
Dev	Pool	Total Trks	Subscribed %	Allocated Trks	Allocated %	Trks WR	Written %
DEF2	META	345150	63	60	0	5	

Meta Members (4)						
Dev	Dev Config	Cap (MB/Cyl)	Meta	Meta Config	Stripe Size (Cyl/Blks)	Stripe Size (Cyl/Tr)
DEF3	TDEV	4314 / 4602	Member	Concatenated	N/A	N/A
DEF4	TDEV	4314 / 4602	Member	Concatenated	N/A	N/A
DEF5	TDEV	4314 / 4602	Member	Concatenated	N/A	N/A
DEF6	TDEV	4314 / 4602	Member	Concatenated	N/A	N/A

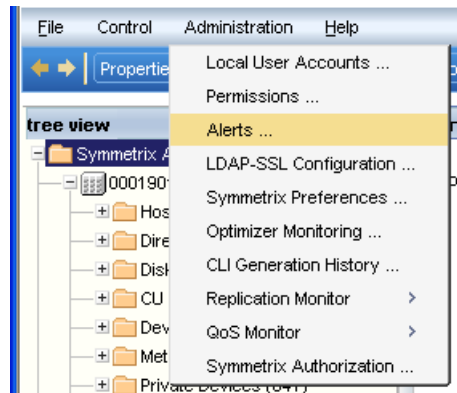
Members can also be removed from concatenated metadvice. In order to perform this change, the meta must be unmasked or unmapped. If a meta member is removed from a metadvice it will remain bound to the thin pool.

Note: Striped thin metadvice cannot be expanded while preserving data.

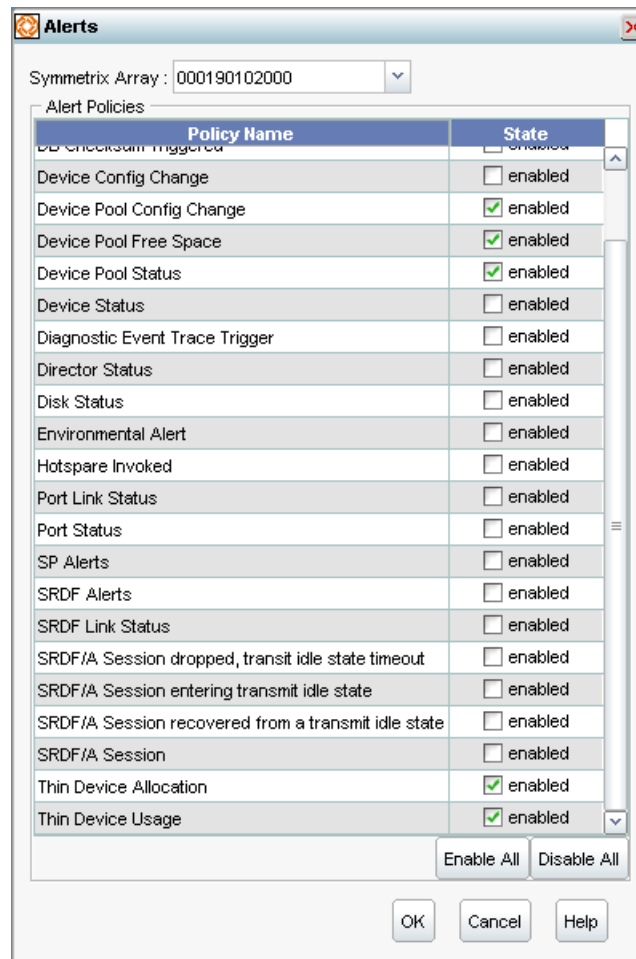
When a striped metadvice is expanded, a BCV meta must be established to the standard metadvice so that the data can be restored to the newly grown standard meta following the expansion. Because thin devices cannot have BCVs attached to them, meta expansion while preserving data is not possible with striped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

Monitoring thin pool space

In 5773 Engenuity 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:

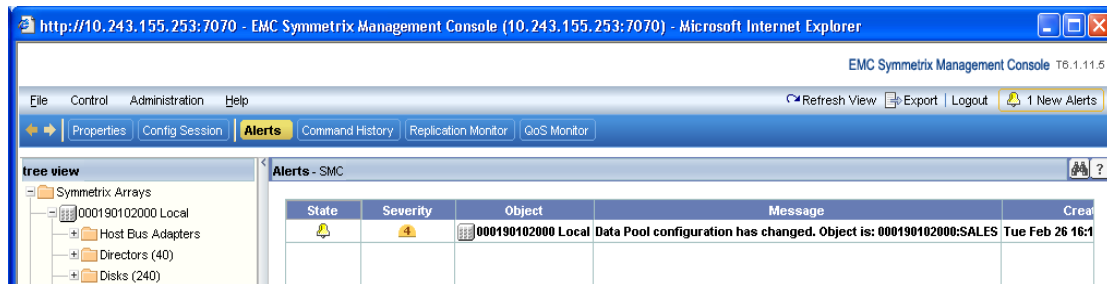
- Device Pool Config Change – Alerts the user that the status of the pool has changed as a result of adding or removing data devices or binding or unbinding thin devices.
- Device Pool Free Space – Alerts the user when the amount of free space remaining in a device pool meets or exceeds a specific percentage.
- 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.

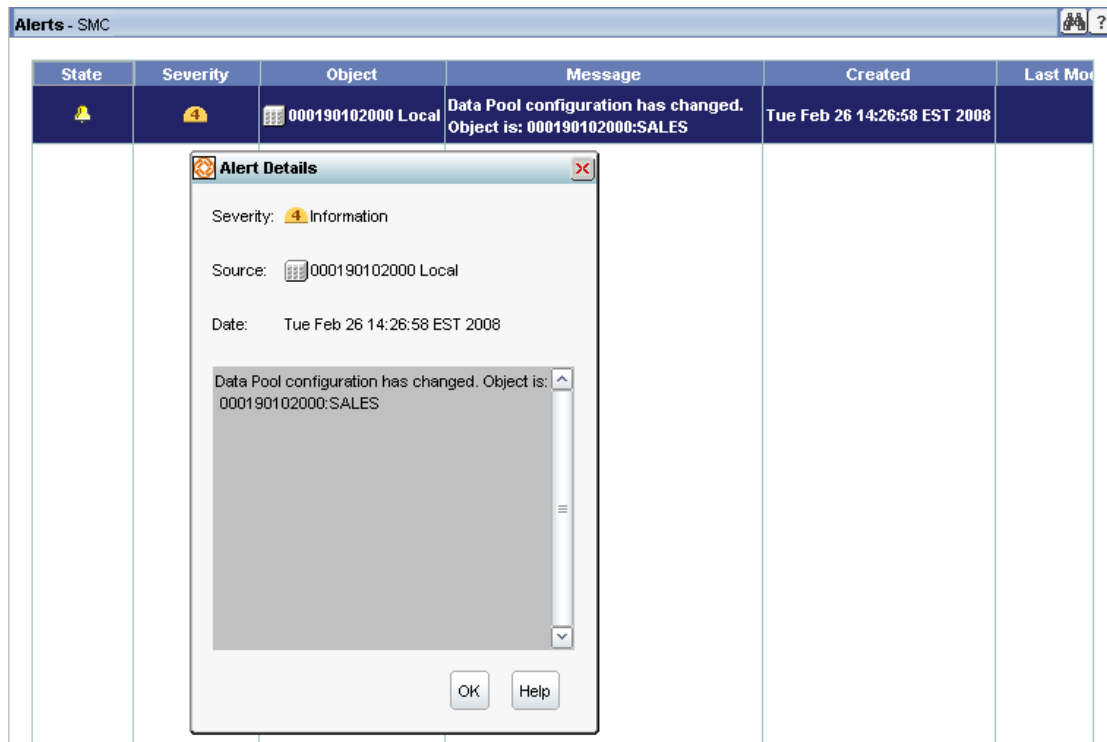
Alerts that are looking at pool or thin device usage or allocation (Device Pool Free Space, Thin Device Allocation, and Thin Device Usage) will trigger when the following thresholds are met or exceeded:

- 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).



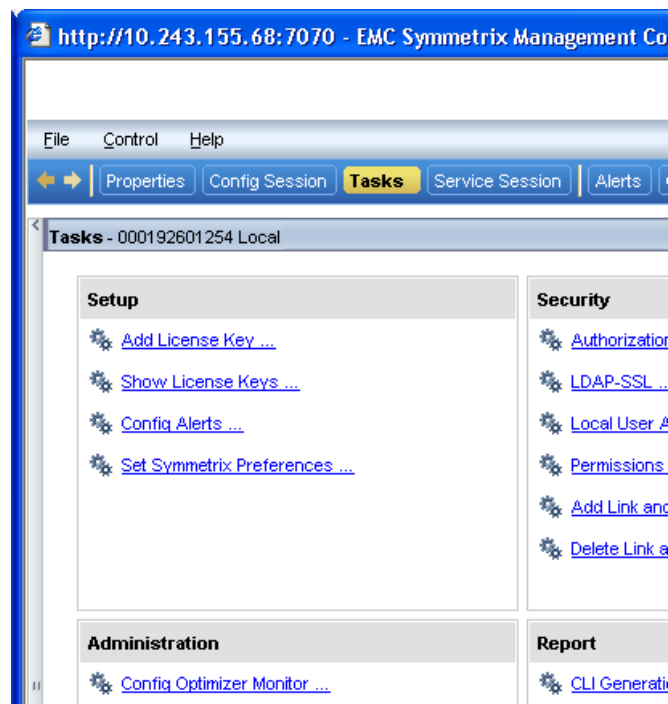
This alert was triggered when a data device in the SALES pool was disabled:



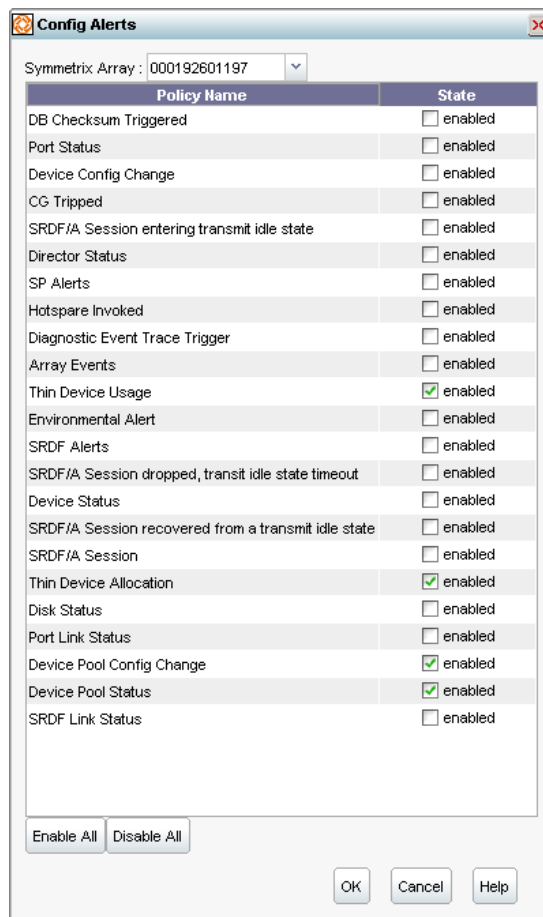
After mapping and masking the thin devices from the SALES pool to a host, file systems were created on devices 0EEE and 0EEF. The file systems were then mounted and files were created on both devices. The file system on device 0EEE was completely filled while 0EEF was filled to about 85% of capacity.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

Alerts - SMC					
State	Severity	Object	Message	Created	Last M
		000190102000 Local	Data Pool utilization is now 65 percent. Object is: 000190102000:SALES	Tue Feb 26 15:15:07 EST 2008	
		000190102000:0eef	Thin Device is now 85 percent allocated. Object is: 000190102000:0eef	Tue Feb 26 15:14:01 EST 2008	
		000190102000:0eef	Thin Device is now 80 percent used. Object is: 000190102000:0eef	Tue Feb 26 15:14:01 EST 2008	
		000190102000 Local	Data Pool utilization is now 60 percent. Object is: 000190102000:SALES	Tue Feb 26 15:14:01 EST 2008	
		000190102000:0eee	Thin Device is now 100 percent allocated. Object is: 000190102000:0eee	Tue Feb 26 14:59:03 EST 2008	
		000190102000:0eee	Thin Device is now 100 percent used. Object is: 000190102000:0eee	Tue Feb 26 14:59:03 EST 2008	
		000190102000:0eee	Thin Device is now 65 percent allocated. Object is: 000190102000:0eee	Tue Feb 26 14:57:57 EST 2008	
		000190102000:0eee	Thin Device is now 65 percent used. Object is: 000190102000:0eee	Tue Feb 26 14:57:57 EST 2008	

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

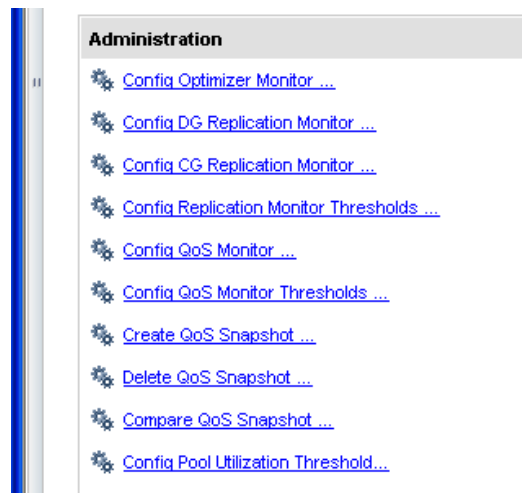


Four of the five alerts available in earlier versions of SMC can be configured as they were for 5773 Enginuity. Checkboxes for Device Pool Config Change, Device Pool Status, Thin Device Allocation, and Thin Device Usage can be selected and will provide alerts at the same thresholds that were applicable to 5773 Enginuity:



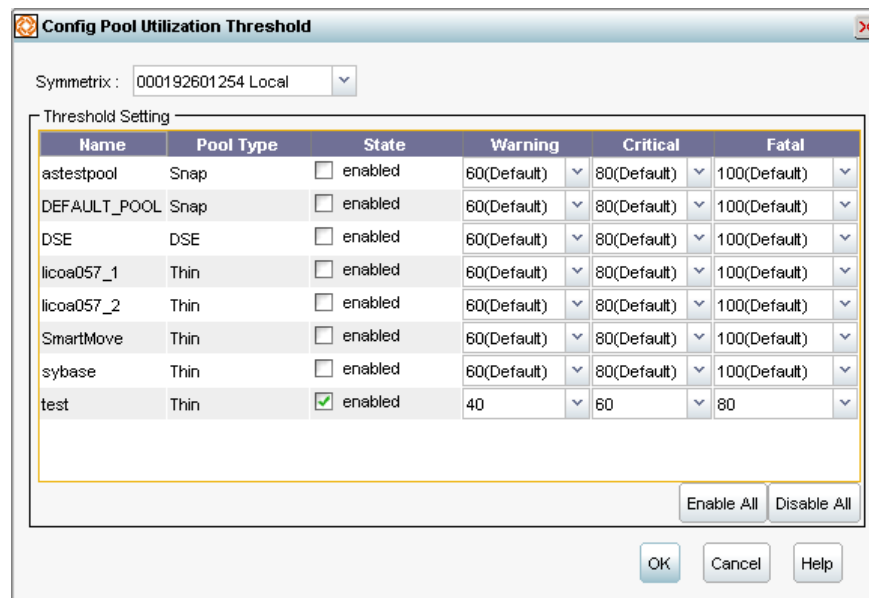
The remaining alert, for Device Pool Free Space, can now be customized by the user to allow specific pools to be monitored for user defined thresholds.

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



In the Config Pool Utilization Threshold dialog box, all of the pools configured in the selected Symmetrix V-Max array will appear. Each has a checkbox that allows the pool to be selected. It also has a pulldown 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 nondefault 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 virtual devices (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:

Properties - Device Groups (1)

Name	Type	SymmlID	STDs	BCVs	VDEVs	TGTs	GKs	Valid
thindg [102000]	Regular	000190102000	4	0	4	0	0	Yes

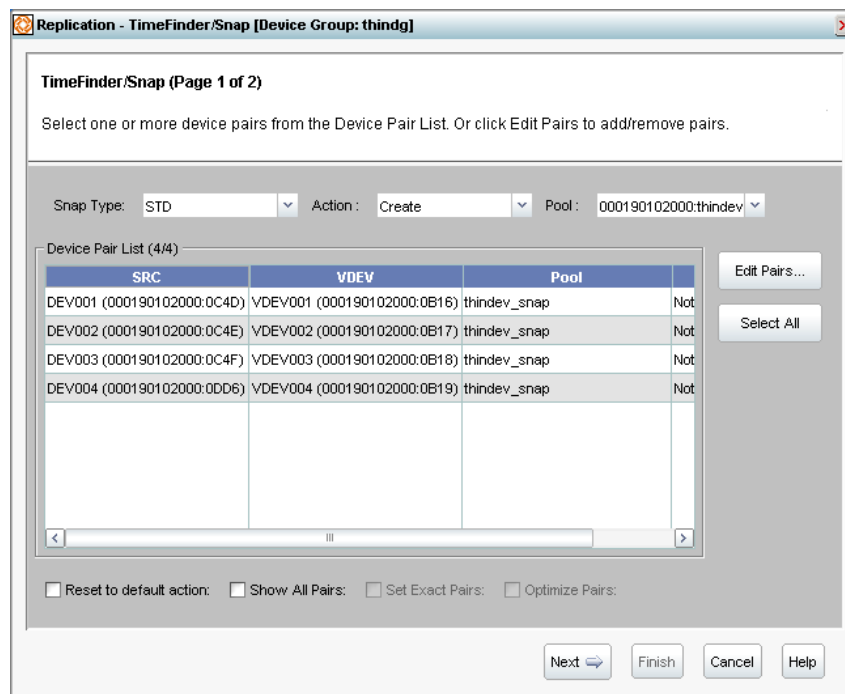
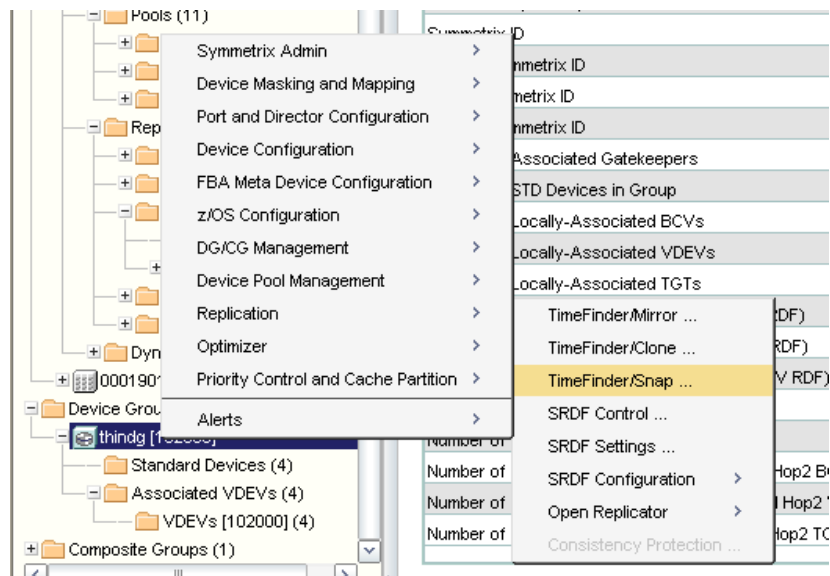
General

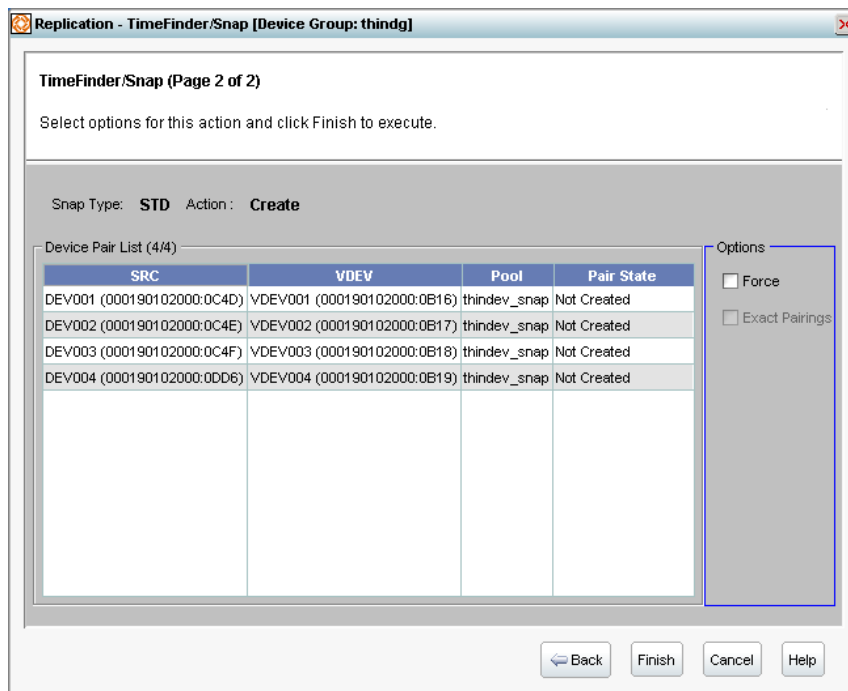
Vendor ID	EMC Corp
Application ID	SMC
Group Type	Regular
Valid	Yes
Device Priority QoS	1
Dynamic Cache Partition Name	DEFAULT_PARTITION
Device Group Creation Time	2008-02-28 15:15:13.0
Device Group Modify Time	2008-02-28 15:15:14.0
Symmetrix ID	000190102000
Remote Symmetrix ID	
RBCV Symmetrix ID	
RRBCV Symmetrix ID	
Number of Associated Gatekeepers	0
Number of STD Devices in Group	4
Number of Locally-Associated BCVs	0
Number of Locally-Associated VDEVs	4

Properties - thindg [102000]/Standard Devices (4)

Dev	LDev	Dev Config	Cap (MB/Cyl)	Dev Status
0C4D	DEV001	TDEV	4314 / 4602	Not Ready
0C4E	DEV002	TDEV	4314 / 4602	Not Ready
0C4F	DEV003	TDEV	4314 / 4602	Not Ready
0DD6	DEV004	TDEV	4314 / 4602	Not Ready

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:



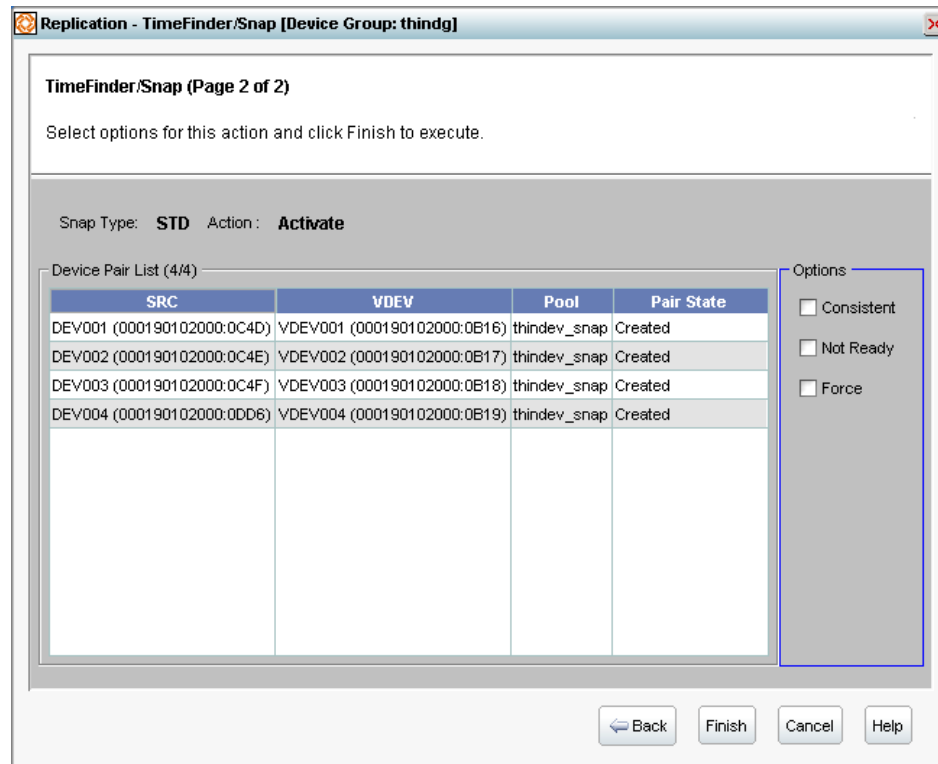


The snap sessions have been created:

Properties - thindg [102000]

General		Snap Query (4)						
LDev	STD	Prot Tracks	LDev	VDev	Changed Tracks	State	Pool Name	% Cor
DEV001	0C4D	69030	VDEV001	0B16	0	Created	thindev_snap	0
DEV002	0C4E	69030	VDEV002	0B17	0	Created	thindev_snap	0
DEV003	0C4F	69030	VDEV003	0B18	0	Created	thindev_snap	0
DEV004	0DD6	69030	VDEV004	0B19	0	Created	thindev_snap	0

They can now be activated:



Properties - thindg [102000]

General Snap Query (4)

LDev	STD	Prot Tracks	LDev	VDev	Changed Tracks	State	Pool Name	% Cor
DEV001	0C4D	69030	VDEV001	0B16	0	CopyOnWrite	thindev_snap	0
DEV002	0C4E	69030	VDEV002	0B17	0	CopyOnWrite	thindev_snap	0
DEV003	0C4F	69030	VDEV003	0B18	0	CopyOnWrite	thindev_snap	0
DEV004	0DD6	69030	VDEV004	0B19	0	CopyOnWrite	thindev_snap	0

Local replication using TimeFinder/Clone

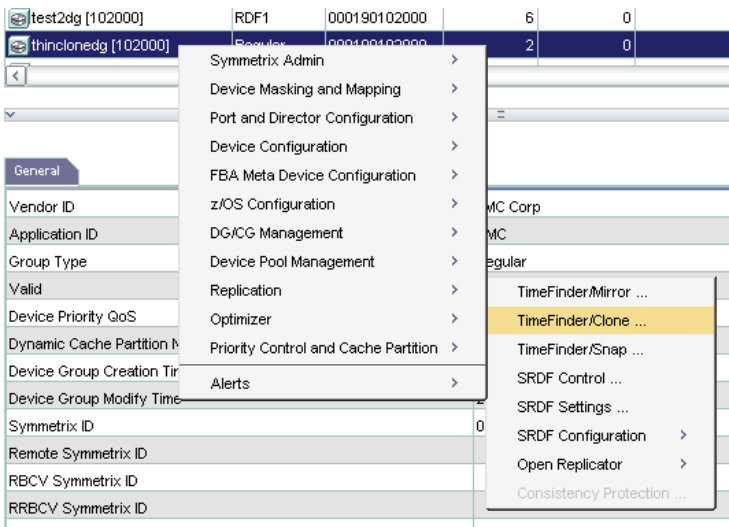
TimeFinder/Clone can be used to create point-in-time copies of thin devices on other thin devices. Other than the requirement that both the source and target device be thin, creating clones of thin devices is exactly the same as creating clones of regular devices. Thin clones can be presented to a host 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 or to different thin devices.

A device group called "thinclone_dg" has been created that contains two

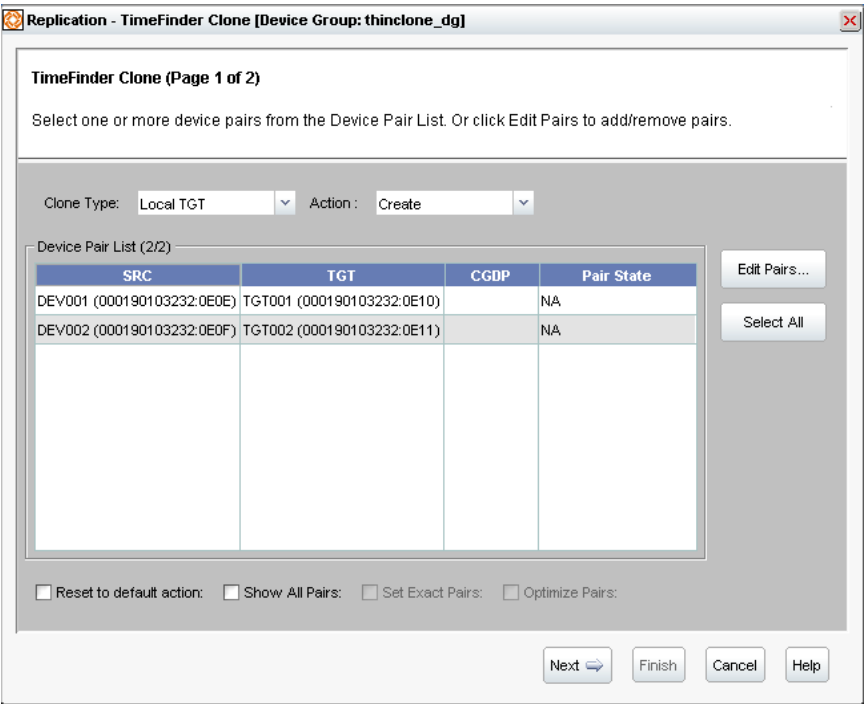
thin devices added to the group as standard (STD) devices and two added as target (TGT) devices:

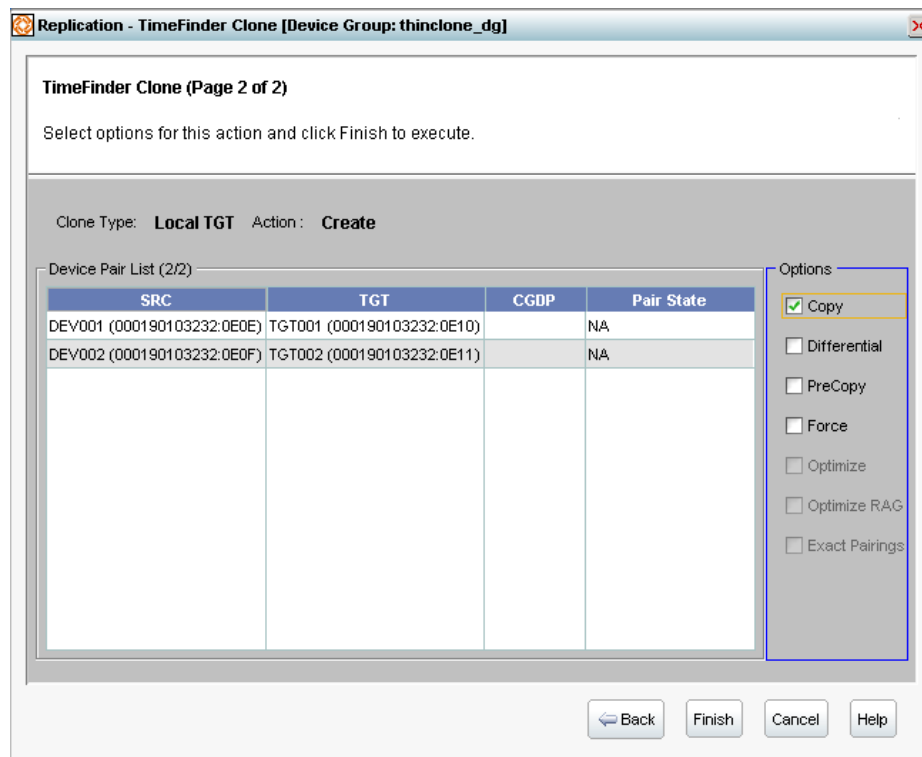
Properties - thincclone_dg [103232]	
General	
Vendor ID	EMC Corp
Application ID	SMC
Group Type	Regular
Valid	Yes
Device Priority QoS	1
Dynamic Cache Partition Name	DEFAULT_PARTITION
Device Group Creation Time	2008-06-13 16:29:23.0
Device Group Modify Time	2008-06-16 11:32:27.0
Symmetrix ID	000190103232
Remote Symmetrix ID	
RBCV Symmetrix ID	
RRBCV Symmetrix ID	
Number of Associated Gatekeepers	0
Number of STD Devices in Group	2
Number of Locally-Associated BCVs	0
Number of Locally-Associated VDEVs	0
Number of Locally-Associated TGTs	2
Number of Remotely-Associated BCVs (STD RDF)	0
Number of Remotely-Associated BCVs (BCV RDF)	0
Number of Remotely-Associated RBCVs (RBCV RDF)	0
Number of Remotely-Associated VDEVs	0
Number of Remotely-Associated TGTs	0
Number of Hop2 BCVs (Remotely-associated Hop2 BCV)	0
Number of Hop2 VDEVs (Remotely-associated Hop2 VDEV)	0
Number of Hop2 TGTs (Remotely-associated Hop2 TGT)	0

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:

Properties - thincclone_dg [103232]										
General		Clone Query (2)								
LDev	STD	Prot Tracks	Src Mod Tracks	LDev	Tgt	Tgt Mod Tracks	CGDP	State	% Copied	
DEV001	0E0E	66606	0	TGT001	0E10	0	XX..	Created	75	
DEV002	0E0F	197718	0	TGT002	0E11	0	XX..	Created	28	

DEV001	
Source Logical Device Name	DEV001
Source Symmetrix Device Name	0E0E
Number of Source Protected Tracks	66606
Number of Source Modified Tracks	0
Target Logical Device Name	TGT001
Target Symmetrix Device Name	0E10
Number of Target Modified Tracks	0
CGDP	XX..
Snap Pair State	Created
Percent Copied	75

They can now be activated:

Replication - TimeFinder Clone [Device Group: thinclone_dg]

TimeFinder Clone (Page 1 of 2)

Select one or more device pairs from the Device Pair List. Or click Edit Pairs to add/remove pairs.

Clone Type: Local TGT Action: Activate

Device Pair List (2/2)

SRC	TGT	CGDP	Pair State
DEV001 (000190103232:0E0E)	TGT001 (000190103232:0E10)	XX..	Created
DEV002 (000190103232:0E0F)	TGT002 (000190103232:0E11)	XX..	Created

☒ Reset to default action: ☐ Show All Pairs: ☐ Set Exact Pairs: ☐ Optimize Pairs:

Next Finish Cancel Help

Replication - TimeFinder Clone [Device Group: thinclone_dg]

TimeFinder Clone (Page 2 of 2)

Select options for this action and click Finish to execute.

Clone Type: Local TGT Action: Activate

Device Pair List (2/2)

SRC	TGT	CGDP	Pair State
DEV001 (000190103232:0E0E)	TGT001 (000190103232:0E10)	XX..	Created
DEV002 (000190103232:0E0F)	TGT002 (000190103232:0E11)	XX..	Created

Options

☐ Consistent

☐ Not Ready

☐ Force

Back Finish Cancel Help

Properties - thinclose_dg [103232]

General Clone Query (2)

LDev	STD	Prot Tracks	Src Mod Tracks	LDev	Tgt	Tgt Mod Tracks	CGDP	State	% Copied
DEV001	0E0E	65110	0	TGT001	0E10	0	XX..	CopyInProg	76
DEV002	0E0F	196710	0	TGT002	0E11	0	XX..	CopyInProg	28

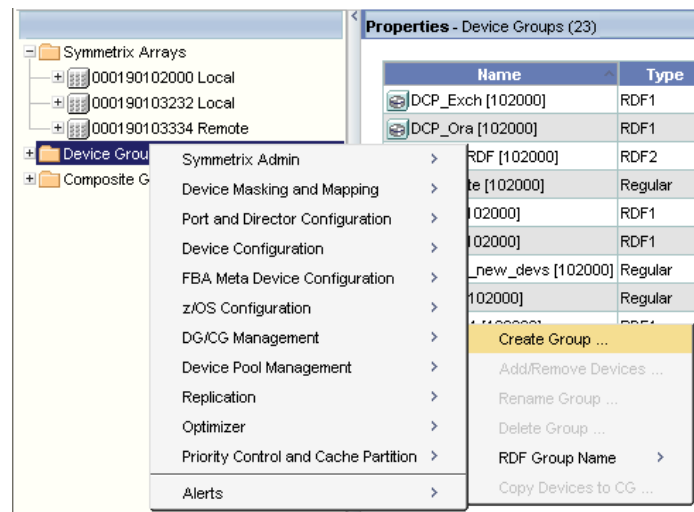
DEV001

Source Logical Device Name	DEV001
Source Symmetrix Device Name	0E0E
Number of Source Protected Tracks	65110
Number of Source Modified Tracks	0
Target Logical Device Name	TGT001
Target Symmetrix Device Name	0E10
Number of Target Modified Tracks	0
CGDP	XX..
Snap Pair State	CopyInProg
Percent Copied	76

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 device be 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):



Device Group Management

Device Group Name: Symmetrix ID: Device Group Type:

Device Type:

Dev Config: Emulation: Cap (MB): Res:

Available

Dev	Dev Config	Cap (MB)	SRDF G

Add Add All Remove Remove All

Group Members

Dev	Dev Config	Cap (MB)	SRDF G
0DF2	RDF2+TDEV	17263	
0DF3	RDF2+TDEV	17263	
0DF4	RDF2+TDEV	17263	
0DF5	RDF2+TDEV	17263	

Page 1 of 1 Items 1 to 4 of 4

☐ Force

Device Group Summary:

STD (4)

Dev	Dev Config	Cap (MB)
0DF2	RDF2+TDEV	17263
0DF3	RDF2+TDEV	17263
0DF4	RDF2+TDEV	17263
0DF5	RDF2+TDEV	17263

OK Cancel Help

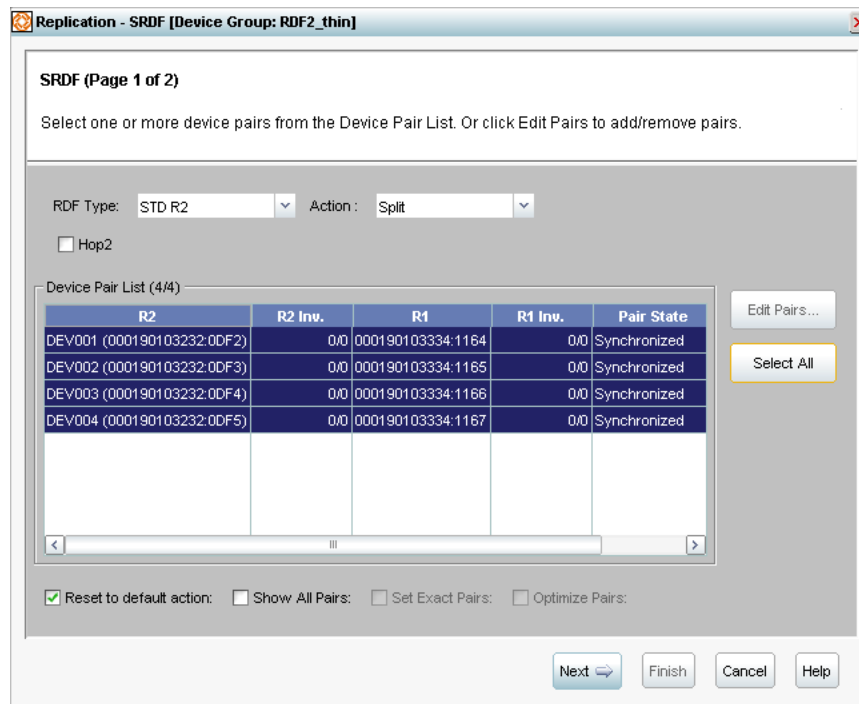
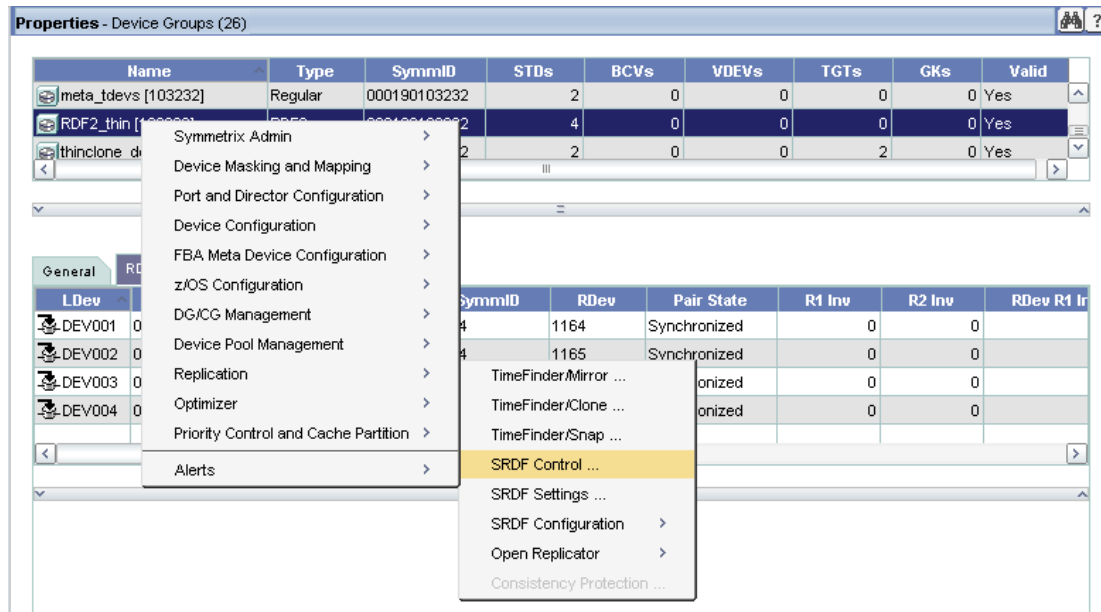
The devices are in a synchronized state and can be the targets of an SRDF operation; in this example, a split:

Properties - Device Groups (26)

Name	Type	SymmlID	STDs	BCVs	VDEVs	TGTs	GKs	Valid
meta_tdevs [103232]	Regular	000190103232	2	0	0	0	0	Yes
RDF2_thin [103232]	RDF2	000190103232	4	0	0	0	0	Yes
thinclone dq [103232]	Regular	000190103232	2	0	0	2	0	Yes

General RDF Query (4)

LDev	Dev	Link Status	Remote SymmlID	RDev	Pair State	R1 Inv	R2 Inv	RDev R1 Ir
DEV001	0DF2	Ready	000190103334	1164	Synchronized	0	0	
DEV002	0DF3	Ready	000190103334	1165	Synchronized	0	0	
DEV003	0DF4	Ready	000190103334	1166	Synchronized	0	0	
DEV004	0DF5	Ready	000190103334	1167	Synchronized	0	0	



Replication - SRDF [Device Group: RDF2_thin]

SRDF (Page 2 of 2)

Select options for this action and click Finish to execute.

RDF Type: **STD R2** Action: **Split**

Device Pair List (4/4)

R2	R2 Inv.	R1	R1 Inv.	Pair State
DEV001 (000190103232:ODF2)	0/0	000190103334:1164	0/0	Synchronized
DEV002 (000190103232:ODF3)	0/0	000190103334:1165	0/0	Synchronized
DEV003 (000190103232:ODF4)	0/0	000190103334:1166	0/0	Synchronized
DEV004 (000190103232:ODF5)	0/0	000190103334:1167	0/0	Synchronized

Options

☐ Immediate

☐ Bypass

☐ Force

☐ SymForce

Back Finish Cancel Help

Properties - Device Groups (26)

Name	Type	SymmlID	STDs	BCVs	VDEVs	TGTs	GKs	Valid
meta_tdevs [103232]	Regular	000190103232	2	0	0	0	0	Yes
RDF2_thin [103232]	RDF2	000190103232	4	0	0	0	0	Yes
thinclone_da [103232]	Regular	000190103232	2	0	0	2	0	Yes

General RDF Query (4)

LDev	Dev	Link Status	Remote SymmlID	RDev	Pair State	R1 Inv	R2 Inv	RDev R1 Ir
DEV001	ODF2	Not Ready	000190103334	1164	Split	0	0	
DEV002	ODF3	Not Ready	000190103334	1165	Split	0	0	
DEV003	ODF4	Not Ready	000190103334	1166	Split	0	0	
DEV004	ODF5	Not Ready	000190103334	1167	Split	0	0	

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 V-Max 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.

Storage 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.

Data Device Extent The minimum quantum of storage that is allocated at a time when dedicating storage from a thin pool for use with a specific thin device.

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 on 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. The extent sizes between a thin and a data device do not need to be the same.

Oversubscribed Thin Pool A thin pool whose thin pool capacity is less than the sum of the reported sizes of the thin devices using the pool.

Pre-provisioning An approach sometimes used to reduce the operational impact of provisioning storage. The approach consists of

satisfying provisioning operations with larger devices that was needed initially, so that the future cycles of the storage provisioning process can be deferred or avoided.

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

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

Thin Device Allocation Limit This refers to a capacity limit that a thin device is entitled to withdraw from a thin pool, which may be equal to or less than the thin device Subscribed capacity.

Thin Device Allocation Ratio The ratio between the thin device allocated capacity and thin device subscribed capacity. This value is expressed as a percent.

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

Thin Device Subscribed Capacity This refers to a total capacity that a thin device is entitled to withdraw from a thin pool, which may be equal to or less than the thin device capacity. In the current implementation they are equal.

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 collection of data devices that provide storage capacity for 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.

Host	File System	Allocated at File System Creation - Devices are 276210 tracks / 17263 MB			Efficiency with Thin Devices
		Number of Tracks	Megabytes	% of Total Device	
AIX	JFS	35040	2190	12.69%	Low
	JFS2	612	38	0.22%	High
	VxFS	612	38	0.22%	High
HP-UX	HFS	27924	1745	10.11%	Low
	JFS (VxFS)	300	19	0.11%	High
Linux	VxFS	1080	1030	0.39%	High
	Ext2	6144	384	2.22%	Medium
	Ext3	6648	416	2.41%	Medium
Solaris	UFS	8676	542	3.14%	Medium
	VxFS	1080	68	0.39%	High
Windows	NTFS	1116	70	0.40%	High

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.

Copyright © 2008, 2009 EMC Corporation. All Rights Reserved.

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

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

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

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

All other trademarks used herein are the property of their respective owners.