DataKeeper for Windows

v7.6

Technical Documentation

June 2013

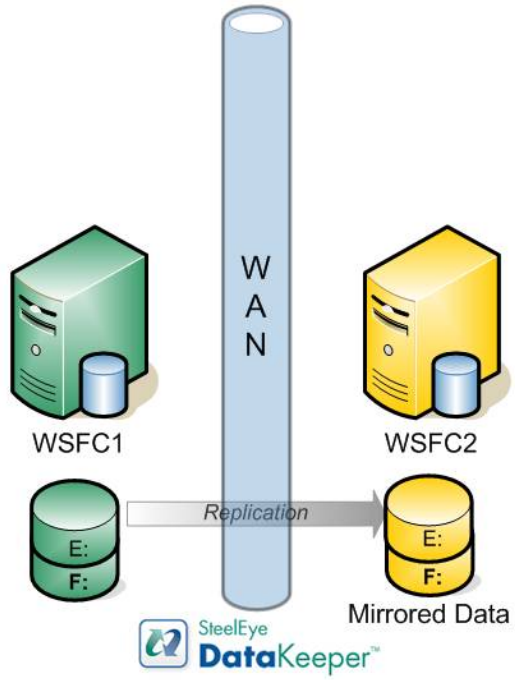
To maintain the quality of our publications, we welcome your comments on the accuracy, clarity, organization, and value of this document.

Address correspondence to:
ip@us.sios.com

Table of Contents

Chapter 1: Introduction

SteelEye DataKeeper is a highly optimized host-based replication solution which ensures your data is replicated as quickly and as efficiently as possible from your source server across the network to one or more target servers.



Features

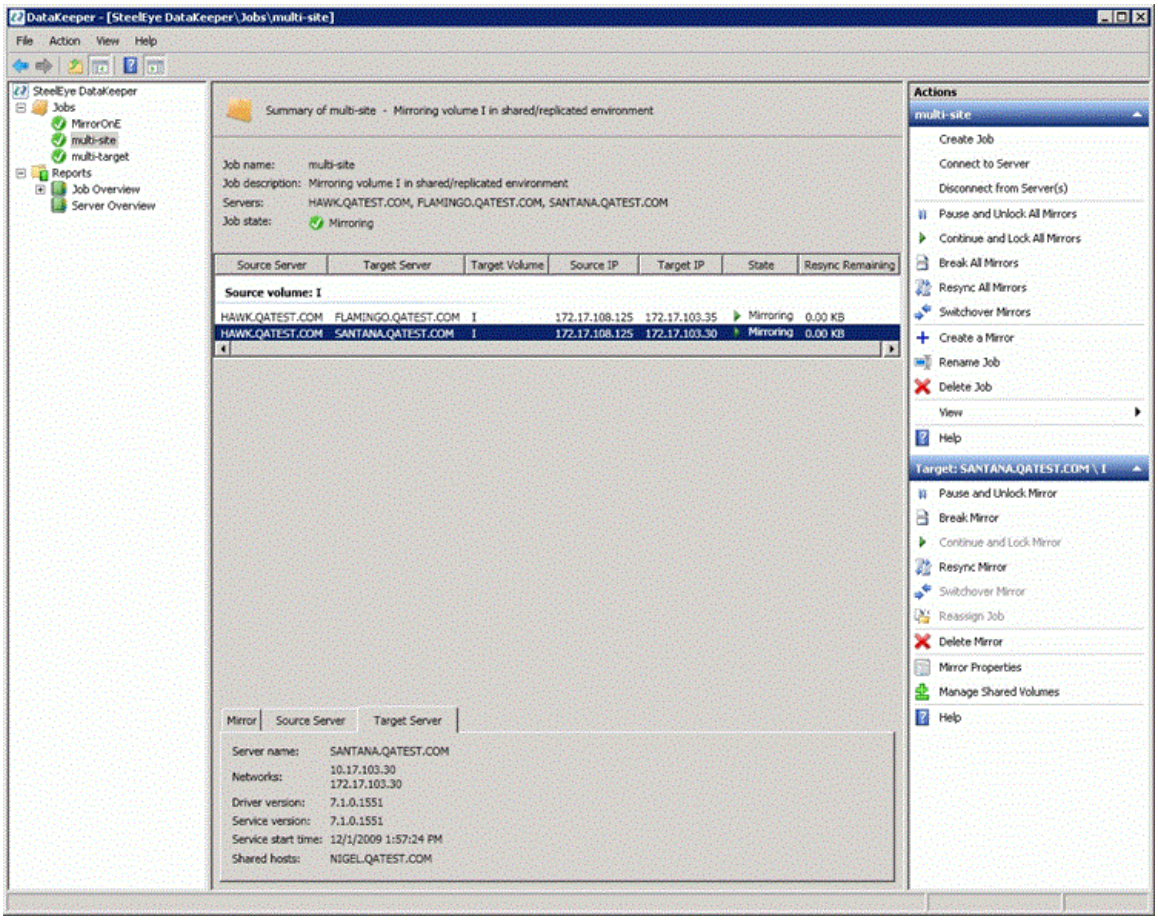
Some of the features include the following:

- Synchronous or Asynchronous block level volume replication.
- Built-in WAN optimization enabling SteelEye DataKeeper to fully utilize a high speed/high latency network connection without the need for WAN accelerators.
- Efficient compression algorithms optimizing use of available bandwidth.
- Intuitive MMC 3.0 GUI.

User Interface

SteelEye DataKeeper User Interface

The SteelEye DataKeeper User Interface uses a standard MMC snap-in interface.



- The left pane displays the Console Tree view. This includes the **Jobs** and **Reports**. Currently, there are two reports available - **Job Overview** and **Server Overview**. The **Job Overview** report provides a summary of all the jobs on the connected servers. The **Server Overview** report provides a summary of all the mirrors on the connected servers.
- The middle pane is the **Summary** view. This includes information about the selected item.
- The right column is the **Actions** view. This pane appears when activated through the **View** menu. The options available from this pane are the same options available from the **Action** menu. This column is divided into two sections. The **Actions** in the top section apply to the job and every mirror within the job. The **Actions** in the bottom section apply only to the selected mirror.
- At the bottom of the main window, three tabs appear: **Mirror**, **Source Server** and **Target Server**.

These tabs provide information on the mirror that has been selected.

- The icon shows the state of the mirror, which provides more information than the icons and states provided in the Failover cluster UI.

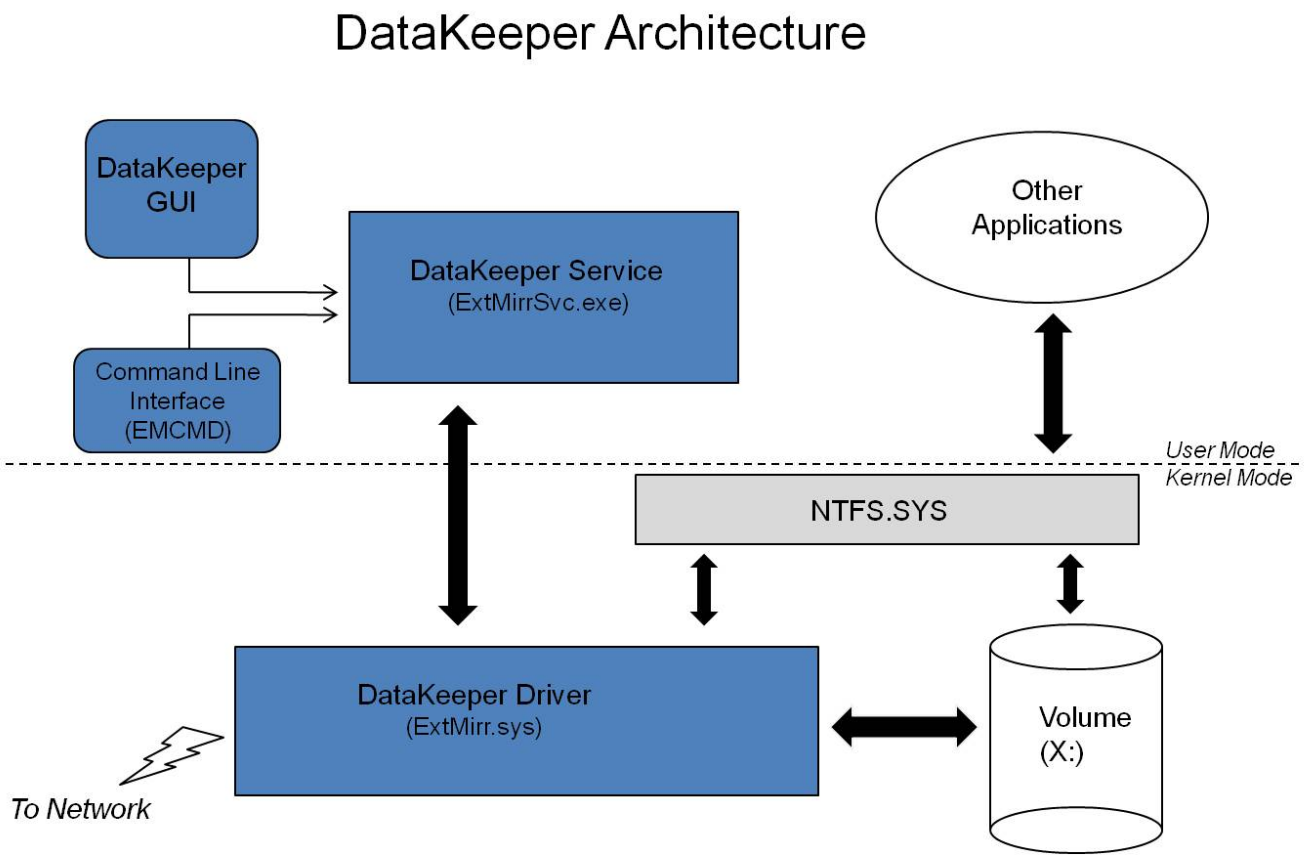
DataKeeper Components

SteelEye DataKeeper for Windows is comprised of the following components:

- **DataKeeper Driver (ExtMirr.sys)** - The DataKeeper Driver is a kernel mode driver and is responsible for all mirroring activity between the mirror endpoints.
- **DataKeeper Service (ExtMirrSvc.exe)** - The DataKeeper Service links the DataKeeper GUI and Command Line Interface to the DataKeeper Driver. All commands to manipulate the mirror are relayed through the DataKeeper Service to the DataKeeper Driver.

  Important: Stopping the DataKeeper Service does not stop mirroring. Sending the driver a PAUSE mirror, BREAK mirror or DELETE mirror command is the only way to interrupt mirroring.
- **DataKeeper Service Log On ID and Password Selection -** The DataKeeper Service Log On ID and Password Selection allows you to select the type of account to be used to start the service. Domain and Server account IDs with administrator privileges allow improved disaster recovery when network disruptions occur.
- **Command Line Interface (EMCMD.exe)** – There is an entire suite of EMCMD command options that can be used to operate DataKeeper.
- **DataKeeper GUI (Datakeeper.msc)** - The DataKeeper GUI is an MMC 3.0 (Microsoft Management Console) based user interface which allows you to control mirroring activity and obtain mirror status.
- **Packaging files, SteelEye Protection Suite scripts, help files, etc.**

The following diagram displays how the DataKeeper components interface with the NTFS file system and each other to perform data replication.



DataKeeper Service Log On ID and Password Selection

During a new DataKeeper installation setup, the user will be prompted for a DataKeeper Service Log On ID and Password.

The DataKeeper Service uses authenticated connections to perform volume switchovers and make mirror role changes across multiple servers. The Log On ID account chosen to run the DataKeeper Service will determine how much authority and permission is available to establish connections between servers and perform volume switchovers, especially when server or network disruptions occur.

Several types of Service Log On ID accounts are available as follows:

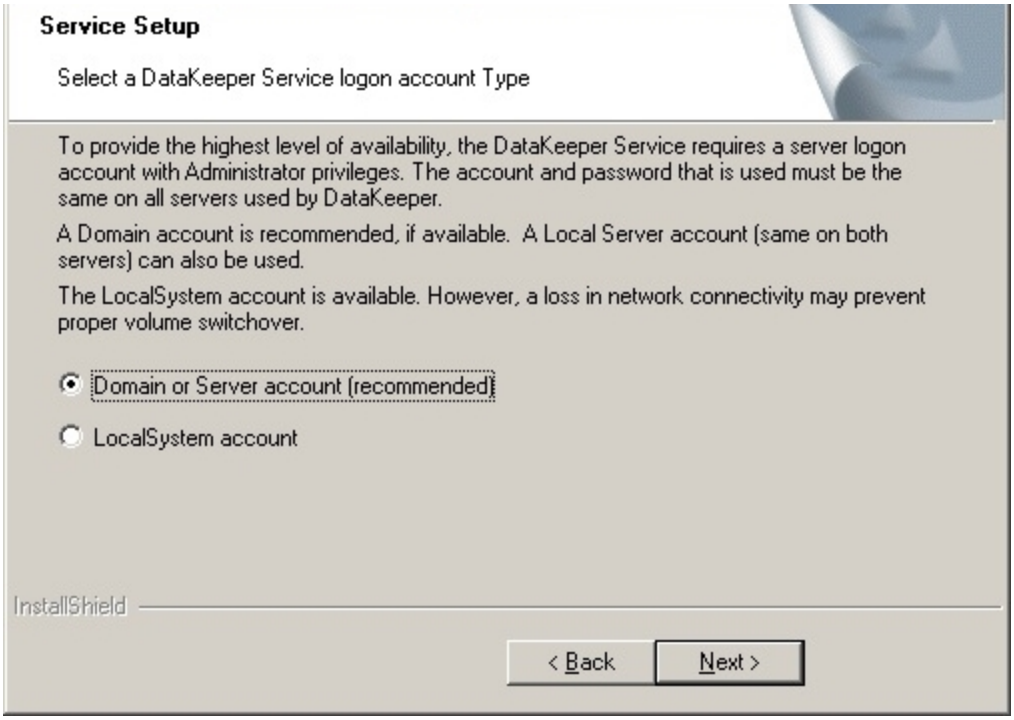
- A **Domain Account** with administrator privileges, valid on all connected servers in the domain (*recommended*)
- A **Server Account** with administrator privileges, valid on all connected servers
- The **Local System Account** (*not recommended*)

  Note: For Workgroups, use the **Server Account** option and use the server name \ administrator on each system as the Service Account for DataKeeper. **You should also log on to all servers using this same Log On ID and Password** (see related Known Issue).

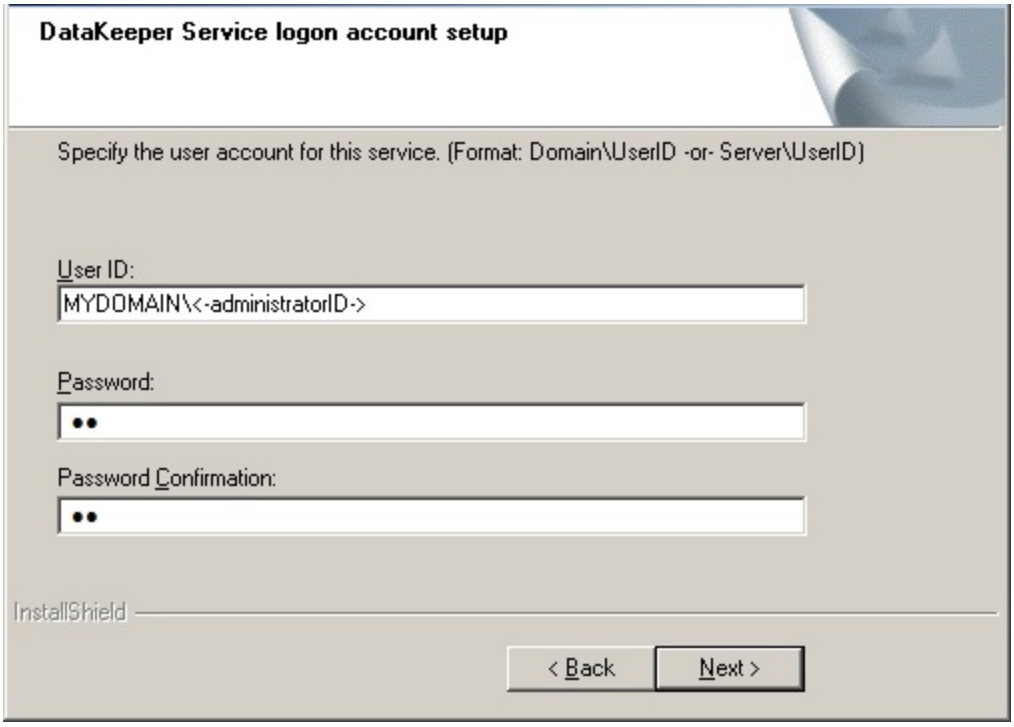
Note: The domain or server account used must be added to the Local System Administrators Group. The account must have administrator privileges on each server in which DataKeeper is installed.

Please note that the Local System account cannot be authenticated properly in a domain when network connectivity with Active Directory is lost. In that situation, connections between servers cannot be established with the Local System account causing DataKeeper volume switchover commands, via the network, to be rejected. IT organizations requiring fault tolerance during a disaster recovery, including network disruptions, should not use the Local System account.

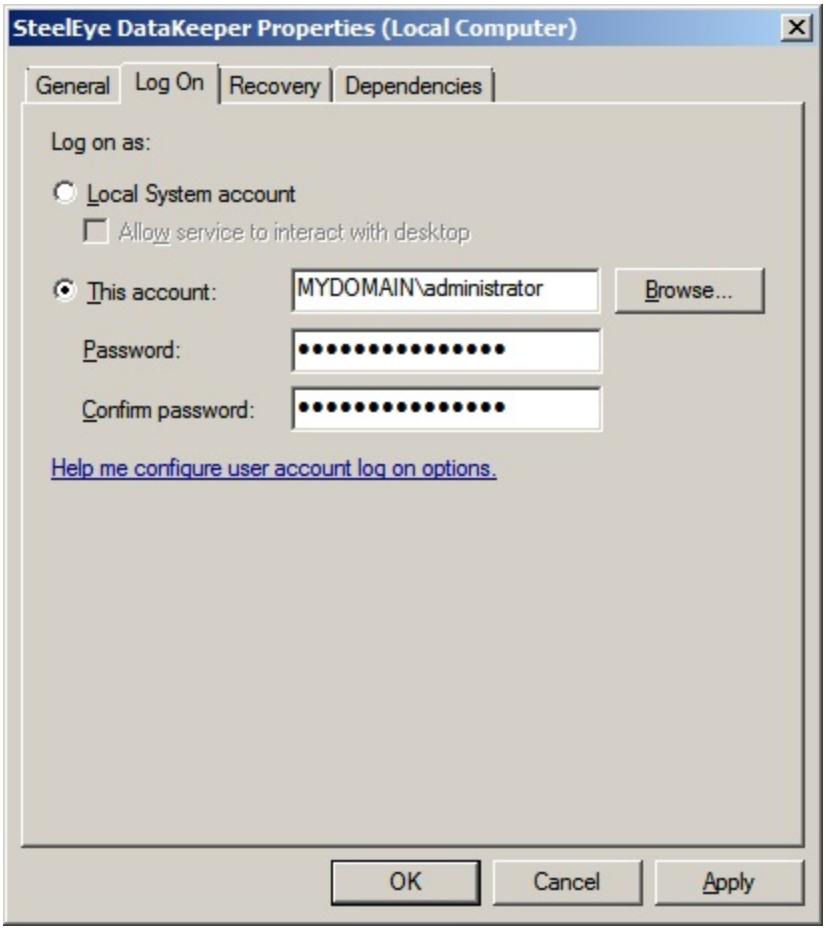
DataKeeper Installation – Service Logon ID Type Selection:



If a Domain or Server account is selected above, the DataKeeper Service Log On ID and Password Entry Form is displayed to enter that information.



If the DataKeeper Service has previously been configured with a Service Log On ID and Password, the setup program will omit the Service ID and Password selection dialogs. However, at any time, an administrator can modify the DataKeeper Service Log On ID and Password using the Windows Service Applet. Be sure to restart the DataKeeper Service after changing the Log On ID and/or Password.



The following table outlines these requirements:

Environment	DataKeeper Service Requirements	DataKeeper UI Requirements
Same Domain or Trusted Domain Environment	• Run the DK Service on all systems as the same account with the same credentials • Okay to use the default = Local System Account	• Log in as a domain admin and run the DK GUI • Or use “run as” Administrator option to run DK GUI

Environment	DataKeeper Service Requirements	DataKeeper UI Requirements
Mixed Environment Servers in a Mixture of Domain and WorkGroup or Servers in Separate Domains	• Create a local account on each system with same account name and password • Add this local account to the Administrator Group • Run the DK Service on all systems with the local account	• Log in using the local account you created to run the DK Service • Run the DK GUI **You should also log on to all servers using this same Log On ID and Password** (see related Known Issue).

Understanding Replication

How SteelEye DataKeeper Works

At the highest level, DataKeeper provides the ability to mirror a volume on one system (source) to a different volume on another system (target) across any network. When the mirror is created, all data on the source volume is initially replicated to the target volume, overwriting it. When this initial synchronization (also referred to as a full resync of the data) of the volumes is complete, the target volume is an exact replica of the source volume in terms of size and data content. Once the mirror is established, DataKeeper intercepts all writes to the source volume and replicates that data across the network to the target volume.

Replication is performed at the block level in one of two ways:

- Synchronous replication
- Asynchronous replication

In most cases, asynchronous mirroring is recommended because of the performance impact of synchronous mirroring.

SteelEye DataKeeper Intent Log

SteelEye DataKeeper uses an intent log (also referred to as a bitmap file) to track changes made to the source volume. This log, stored on the mirror source system, is a persistent record of write requests which have not yet been committed to both servers.

The intent log gives SteelEye DataKeeper the ability to survive a source system failure without requiring a full mirror resync after the recovery of the source server.

There is some performance overhead associated with the intent log, since each write to the volume must also be reflected in the intent log file. To minimize this impact, it is recommended that the intent logs be stored on a physical disk that is not involved in heavy read or write activity.

Non-Shared Volumes

By default, this intent log feature is enabled, and the intent log files are stored in a subdirectory called "Bitmaps" under the directory where SteelEye DataKeeper was installed.

To create the intent log file in a directory other than the default location, set the BitmapBaseDir registry entry to a directory where SteelEye DataKeeper will create the file. See "Relocation of Intent Log" for more information.

To disable the intent log feature, clear the BitmapBaseDir registry entry (set it to an empty string) on all current and potential mirror endpoint servers. **Disabling the intent log requires a reboot on each of these systems in order for this setting to take effect.** Keep in mind that if this feature is disabled, a full resync will be performed in the event of a source system failure.

Shared Volumes

When replicating shared volumes, the intent log files are stored in a subdirectory called "ReplicationBitmaps" on the replicated volume itself. This is necessary to allow switchover to the other shared source servers without resulting in a full resync of the data.

SIOS does not recommend relocating intent logs from their default locations.

Configuration Issue

When configuring a BitmapBaseDir registry entry, make sure that the folder and drive letter specified exist. If configured with a drive letter that does not exist, the following message will be received upon system boot up:

```
Global bitmap volume {drive letter}: has not been detected yet.
Mirror source threads may hang if this volume does not exist. Check
to make sure that the BitmapBaseDir registry entry specifies a valid
volume for storage of bitmaps.
```

Relocation of Intent Log

To relocate the Intent Log (bitmap file), please perform the following on all servers involved:

Note: LEAVE THE MIRROR IN THE MIRRORING STATE! Do not pause it and then move the bitmap file.

1. If you have more than one DataKeeper mirror, move all mirrors to a single system so that it is source for all mirrors.
2. On all systems, create the directory for the new location of the bitmap files (i.e. *R:\Bitmaps*). **Important**: If you choose to relocate the bitmap file from the default location (*%EXTMIRRBASE%\Bitmaps*), you must first create the new directory before changing the location in the registry and rebooting the system.

3. Modify the BitmapBaseDir registry value on all systems other than the mirror source system to reflect the new location. This includes mirror targets and any systems that share the volume with the mirror source or share with any of the targets.

   Edit Registry via `regedit`:

   ```
   HKEY_LOCAL_
   MACHINE\SYSTEM\CurrentControlSet\Services\ExtMirr\Parameters
   ```

   Modify the "`BitmapBaseDir`" parameter, change to the new location (i.e. *`R:\Bitmaps`*)

4. Reboot each of the non-source systems. If this volume is part of a Windows cluster, be sure that you do not shut down too many nodes simultaneously or you may lose the cluster quorum and cause the cluster to shut down on the remaining nodes.

5. Switch any volumes on the source system over to another system (target or shared source). Repeat Steps 2 and 3 on the system that was previously source.

6. After rebooting the original source system, all volume resources can be switched back to that system.

SteelEye DataKeeper Resynchronization

SteelEye DataKeeper performs resynchronization through the use of a bitmap file (intent log). It allocates memory that is used to keep track of "dirty" or "clean" blocks. When a full resync begins, SteelEye DataKeeper initializes the bit for each block that is in use by the file system to 1 ("dirty"), indicating that it needs to be sent to the target system. A full resync occurs at the initial creation of a mirror and during the resync operation after a mirror is broken. It then starts at the beginning of the bitmap, finds the first block whose bit is set to 1 or dirty, reads the corresponding block from the local hard disk, and sends it to the remote system. After this has completed successfully, it sets the block to 0 ("clean"). SteelEye DataKeeper then finds the next dirty bit and repeats this process.

As new writes come in during a resync, the corresponding blocks are set to 1 or dirty.

Once resync gets to the end of the bitmap, it looks to see if there are still any dirty blocks. It does this through a counter that is incremented when one is made dirty and decremented when cleaned. If any blocks are dirty, it resets its pointer to the beginning of the bitmap and starts again, only sending the dirty blocks to the remote system.

This process continues for multiple passes until all blocks are clean. When this happens, the mirror will go from the **Resynchronizing** state to the **Mirroring** state, and at that point, every write is mirrored (the bitmap is no longer necessary at that point).

You can follow the resynchronization process by viewing the resynchronization control counters in Performance Monitor.

This same resynchronization mechanism is used when you CONTINUE a PAUSED mirror.

Warning: If the target system is rebooted/shut down via the DK GUI when mirrors are paused and unlocked, a full resync will occur. To prevent the full resync in this case, be sure to perform a "Continue and Lock" prior to rebooting or shutting down the target system.

Initial Creation of a Mirror

When the mirror is created, DataKeeper must perform an initial synchronization of the data from the source volume to the target volume. This is referred to as a full resync. However, prior to this initial full resync of the data, DataKeeper first performs a process called "**whitespace elimination**" where all blocks of currently unused space on the source volume are eliminated from the initial synchronization and those blocks do not have to be replicated to the target volume.

Example: Whitespace Elimination

Source Volume Capacity	80 GB
Source Volume Free Space	35 GB
Amount of data to be resynced from source volume to target volume during initial creation of the mirror.	55 GB

Synchronous and Asynchronous Mirroring

SteelEye DataKeeper employs both asynchronous and synchronous mirroring schemes. Understanding the advantages and disadvantages between synchronous and asynchronous mirroring is essential to the correct operation of SteelEye DataKeeper.

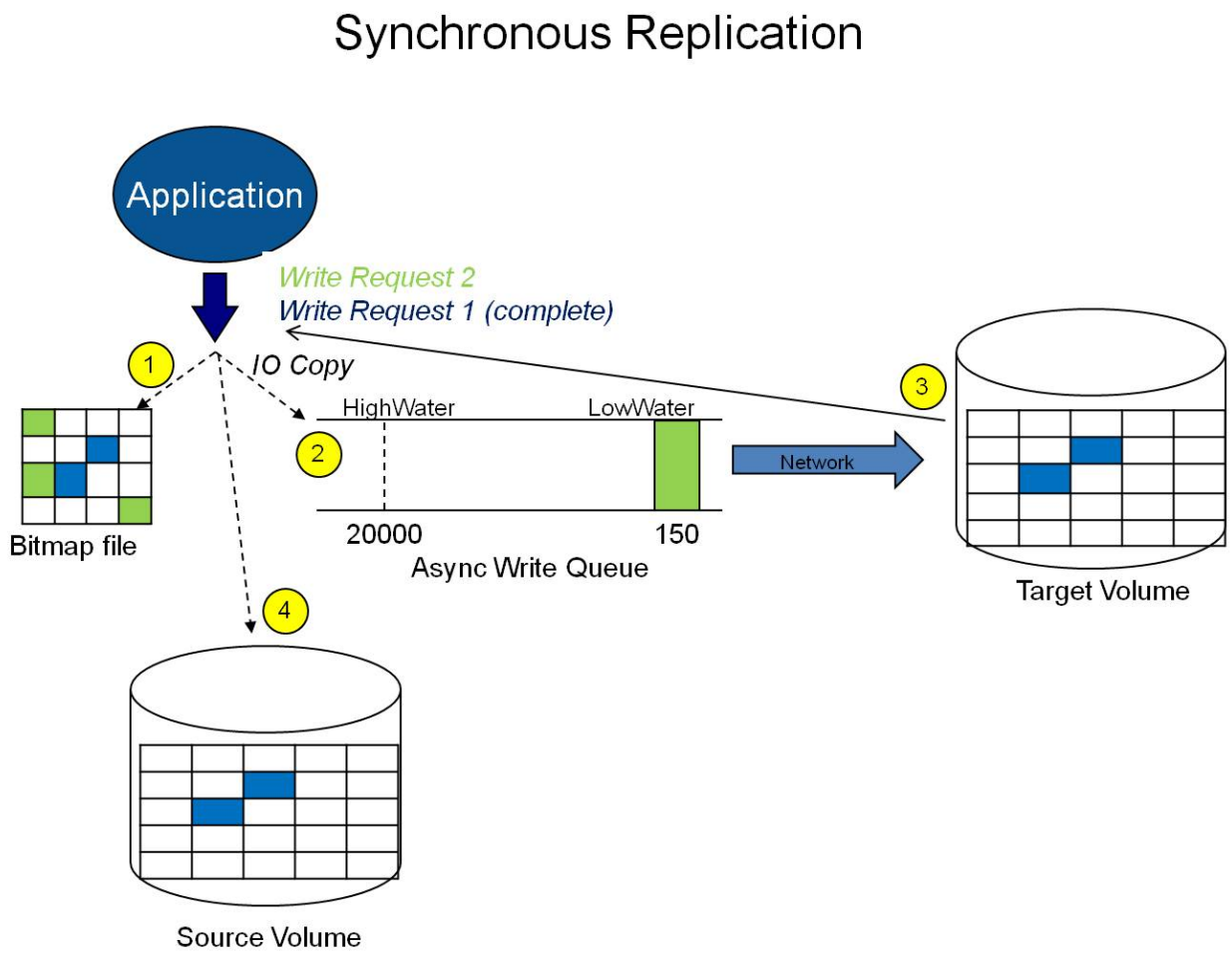
Synchronous Mirroring

With synchronous mirroring, each write is intercepted and transmitted to the target system to be written on the target volume at the same time that the write is committed to the underlying storage device on the source system. Once both the local and target writes are complete, the write request is acknowledged as complete and control is returned to the application that initiated the write. Persistent bitmap file on the source system is updated.

The following sequence of events describes what happens when a write request is made to the source volume of a synchronous mirror.

1. The following occur in parallel.
    a. Write request is intercepted and transmitted to the target system.
    b. Target system executes the write request on the target volume and sends the status of the write back to the source system.
    c. When the target system returns a successful status, the source system executes the write to the source volume and returns to the caller.
2. Should an error occur during network transmission or while the target system executes its target

volume write, the write process on the target is terminated. The source system will complete the write request on its source volume, and the state of the mirror changes from ***Mirroring*** to ***Paused***.



In this diagram, Write Request 1 has already completed. Both the target and the source volumes have been updated.

Write Request 2 has been sent from the application and the write is about to be written to the target volume. Once written to the target volume, DataKeeper will send an acknowledgment that the write was successful on the target volume, and in parallel, the write is committed to the source volume.

At this point, the write request is complete and control is returned to the application that initiated the write.

While synchronous mirroring insures that there will be no data loss in the event of a source system failure, synchronous mirroring can have a significant impact on the application's performance, especially in WAN or slow network configurations, because the application must wait for the write to occur on the source and across the network on the target.

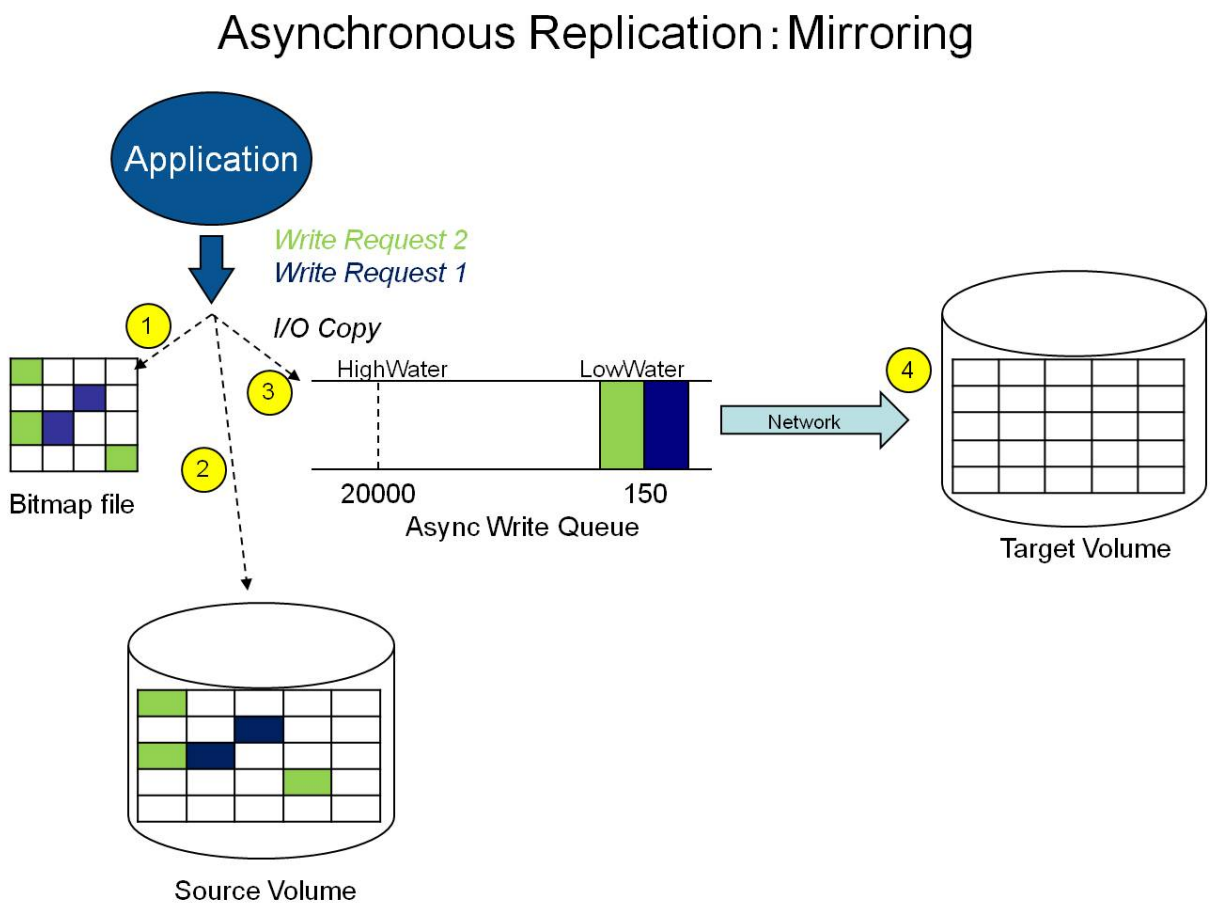
Asynchronous Mirroring

In most cases, SIOS recommends using asynchronous mirroring. With asynchronous mirroring, each write is intercepted and a copy of the data is made. That copy is queued to be transmitted to the target system as soon as the network will allow it. Meanwhile, the original write request is committed to the underlying storage device and control is immediately returned to the application that initiated the write. (**Note**: Certain database applications may send flush commands causing DataKeeper to perform in a synchronous manner. To prevent performance from being impacted in such cases, the registry entry "DontFlushAsyncQueue" may be set.)

At any given time, there may be write transactions waiting in the queue to be sent to the target machine. But it is important to understand that these writes reach the target volume in time order, so the integrity of the data on the target volume is always a valid snapshot of the source volume at some point in time. Should the source system fail, it is possible that the target system did not receive all of the writes that were queued up, but the data that has made it to the target volume is valid and usable.

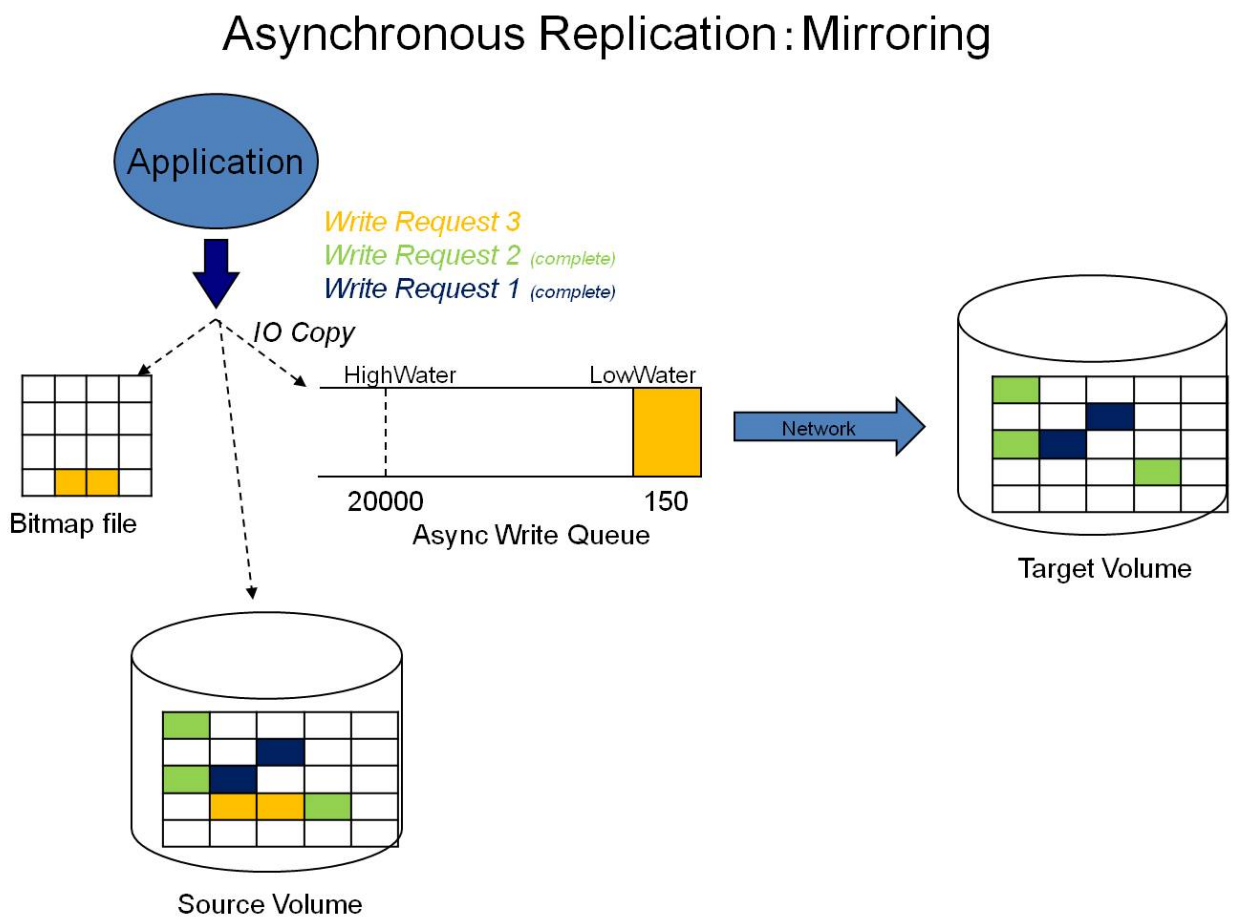
The following sequence of events describes what happens when a write request is made to the source volume of a synchronous mirror.

1. Persistent bitmap file on the source system is updated.
2. Source system adds a copy of the write to the Asynchronous Write Queue.
3. Source system executes the write request to its source volume and returns to the caller.
4. Writes that are in the queue are sent to the target system. The target system executes the write request on its target volume and then sends the status of the write back to the primary.
5. Should an error occur during network transmission or while the target system executes its target volume write, the write process on the secondary is terminated. The state of the mirror then changes from ***Mirroring*** to ***Paused***.



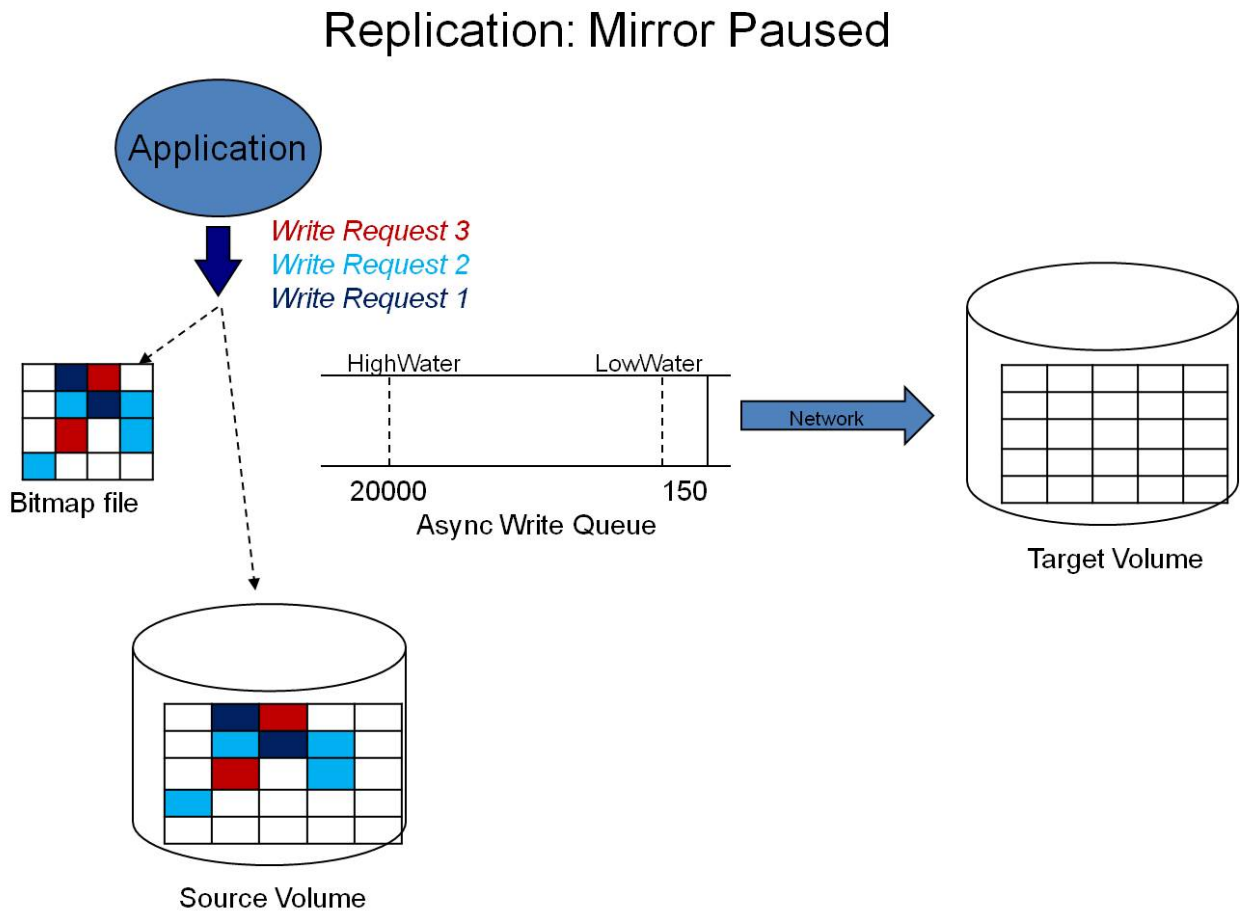
In the diagram above, the two write requests have been written to the source volume and are in the queue to be sent to the target system. However, control has already returned back to the application who initiated the writes.

In the diagram below, the third write request has been initiated while the first two writes have successfully been written to both the source and target volumes. While in the mirroring state, write requests are sent to the target volume in time order. Thus, the target volume is always an exact replica of the source volume at some point in time.



Mirror PAUSED

In the event of an interruption to the normal mirroring process as described above, the mirror changes from the **MIRRORING** state to a **PAUSED** state. All changes to the source volume are tracked in the persistent bitmap file only and nothing is sent to the target system.



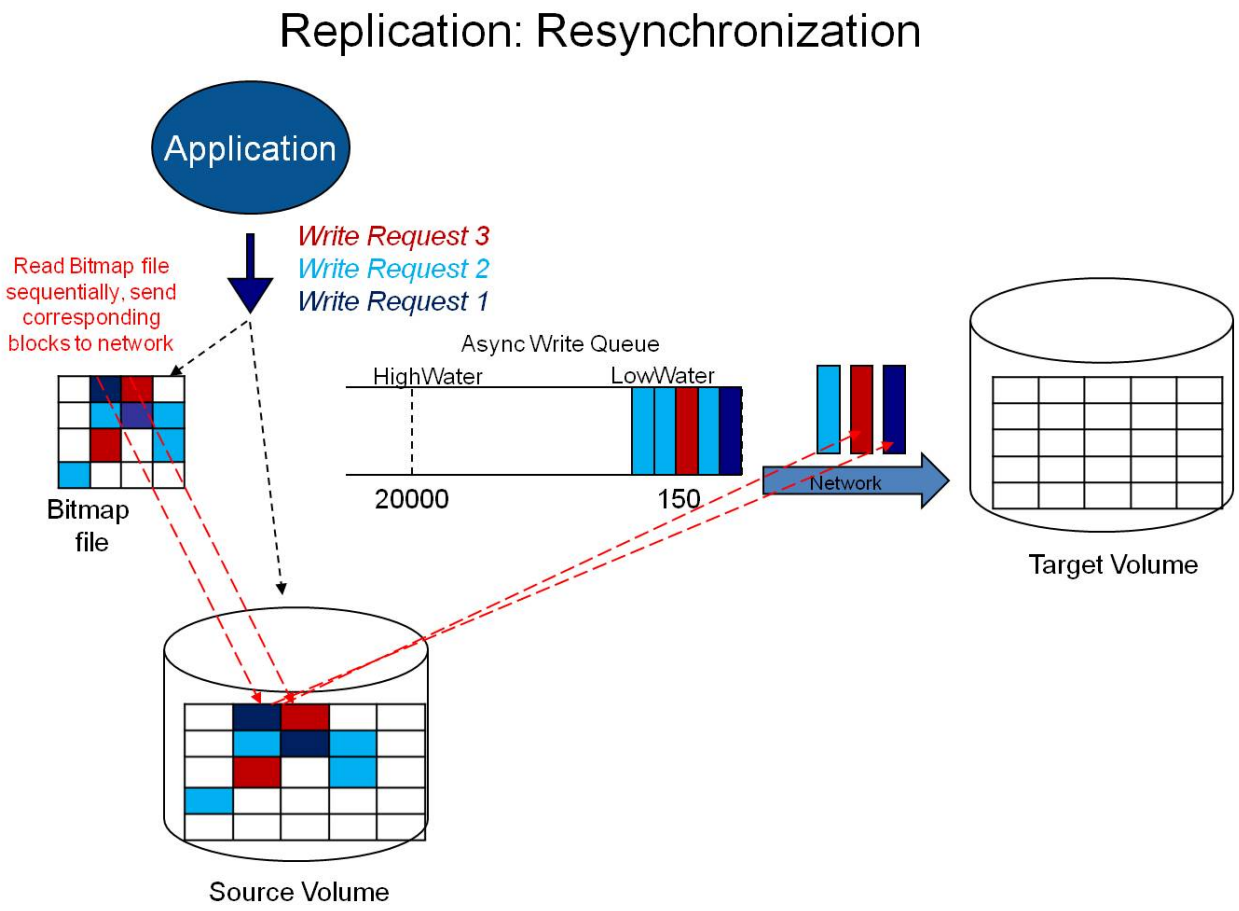
Mirror RESYNCING

When the interruption of either an Asynchronous or Synchronous mirror is resolved, it is necessary to resynchronize the source and target volumes and the mirror enters into a **RESYNC** state.

DataKeeper reads sequentially through the persistent bitmap file to determine what blocks have changed on the source volume while the mirror was **PAUSED** and then resynchronizes only those blocks to the target volume. This procedure is known as a partial resync of the data.

The user may notice a **Resync Pending** state in the GUI, which is a transitory state and will change to the **Resync** state.

During resynchronization, all writes are treated as Asynchronous, even if the mirror is a Synchronous mirror. The appropriate bits in the bitmap are marked dirty and are later sent to the target during the process of partial resync as described above.



Read and Write Operations

After the volume mirror is created and the two drives on the primary and secondary servers are synchronized, the following events occur:

- The system locks out all user access to the target volume; reads and writes are not allowed to the target volume. The source volume is accessible for both reads and writes.
- Both mirrored and non-mirrored volume read operations arriving at the driver on the primary server are passed on and allowed to complete normally without intervention. Reads of a mirrored volume on the secondary system are not allowed, i.e., the secondary has not assumed the role of a failed primary.
- Whenever the primary server receives a write request, the system first determines whether the request is for a mirrored volume. If not, the write is allowed to complete normally without any further intervention. If the write request is for a mirrored volume, the request is handled depending on the mirroring type:
    - If the type is synchronous, then the write request is sent to both the source and target volumes at the same time. Should an error occur during network transmission or while the target system executes its write, the write process on the target is terminated. The source will then complete

the write request, and the state of the mirror changes from **Mirroring** to **Paused**. The write operation is not acknowledged as complete until the source disk write completes and notification from the target is received (success or failure).

- If the type is asynchronous, then the primary executes the write request to its source volume, puts a copy of the write on the asynchronous write queue and returns to the caller. Writes that are in the queue are sent to the target volume. The secondary system executes the write request on the target volume and then sends the status of the write back to the primary. Should an error occur during network transmission or while the secondary executes its mirrored volume write, the write process on the secondary is terminated. The state of the mirror then changes from **Mirroring** to **Paused**.

To ensure uninterrupted system operation, SteelEye DataKeeper momentarily pauses the mirror and automatically continues it (i.e., performs a partial resync) in the following cases:

- In Asynchronous mirroring, when the asynchronous write queue length reaches the WriteQueueHighWater mark due to a massive number of writes to the volume in a short period of time (e.g., database creation). The user can monitor the mirroring behavior using the SteelEye DataKeeper Performance Monitor counters and adjust the WriteQueueHighWater mark if necessary. See Registry Entries for more details.
- When transmission of a write to the target system times out or fails due to resource shortage (e.g., source system resource starvation due to a flood of writes/network transmissions in a short period of time).

Volume Considerations

SteelEye DataKeeper primary and secondary systems have three types of volumes: system, non-mirrored and mirrored. During mirroring operations, system and non-mirrored volumes are not affected and the user has full access to all applications and data on these volumes.

What Volumes Cannot be Mirrored

The SteelEye DataKeeper service filters out the following types of disk partitions:

- Windows system volume
- Volume(s) that contain the Windows pagefile
- Non-NTFS formatted volumes (e.g. FAT, FAT32, Raw FS)
- Non-fixed drive types (e.g. CD-ROMs, diskettes)
- Target volumes that are smaller than the source volume

Volume Size Considerations

The source and target systems are not required to have drives of the same physical size. When the mirror is established, the target volume must be the same size, or larger than the source volume.

There is no limit on the size of volumes that can participate in a SteelEye DataKeeper mirror. However, you should be aware that on initial mirror creation, all data that is in use by the file system on the source volume must be sent to the target. For instance, on a 20 GB volume with 2 GB used and 18 GB free, 2 GB of data must be synchronized to the target. The speed of the network connection between the two systems, along with the amount of data to be synchronized, dictates how long the initial mirror creation will take.

Specifying Network Cards for Mirroring

SteelEye DataKeeper allows the administrator to specify which IP addresses should be used as mirror end-points. This allows the replicated data to be transmitted across a specific network which permits the user to segment mirrored traffic away from the client network if desired.

Dedicated LAN for Replication

While it is not required, a dedicated (private) network between the two servers will provide performance benefits and not adversely affect the client network.

Performance Monitor Counters

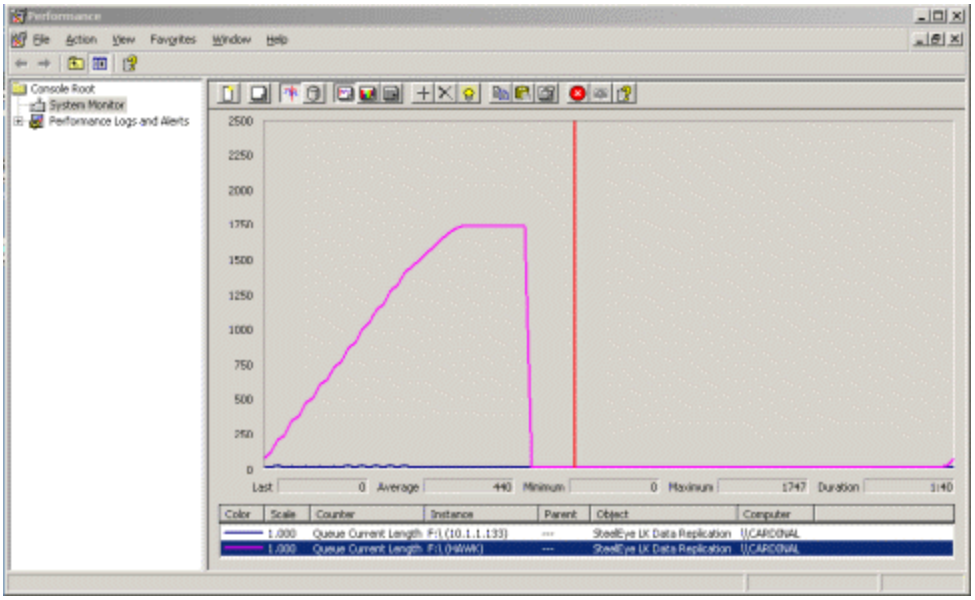
SteelEye DataKeeper provides counters that extend Performance Monitor with statistics about the status of mirroring on volumes. The counters are installed during the full installation of SteelEye DataKeeper software.

To access the counters, do the following:

1. On a **Microsoft Windows 2003** system, start the **Windows Performance Monitor** through the **Start** menu in the **Administrative Tools** program group by selecting **Performance**.
   On a **Microsoft Windows 2008** system, start the **Windows Performance Monitor** through the **Start** menu in the **Reliability and Performance** group.
   On a **Microsoft Windows 2012** system, start the **Windows Performance Monitor** through the **Performance Monitor** option in the **Administrative tools** .
   On all versions of Windows, you can start performance monitor through entering **`perfmon.msc`** using the command line.
2. Select **System Monitor** from the **Console Root** pane.
3. Click the **+** button in the chart pane to open the **Add Counters** dialog box.
4. Select the **SteelEye Data Replication** object.

On a system with a mirror in the source role, there will be one instance available for each target of that mirror.

SteelEye DataKeeper provides 14 counters that allow the monitoring of various operations related to the product. These counters allow the monitoring of such things as status, queuing statistics and general mirror status.



Mirror State Counters

Mirror Elapsed Time

Default Value: 0

Range: 0 - MAX_ULONG

This value represents the amount of time, in seconds, that the volume has been in Mirror state. This value will be 0 for volumes that are not currently involved in a mirror, volumes that are currently undergoing mirror creation (and synchronization), and volumes for which a mirror has been broken or deleted.

Mirror State

Default: 0

Range: 0 - 5

This value represents the current mirroring state of a volume. The following values are defined:

0 None - The volume is not currently involved in a mirror.

1 Mirroring - The volume is currently mirroring to a target.

2 Resynchronizing - The volume is currently being synchronized with its target.

3 Broken - The mirror exists but the source and target volumes are not in sync. New writes to the volume are not tracked.

4 Paused - The mirror exists but the source and target volumes are not in sync. The source server keeps track of any new writes.

5 Resync Pending - The source volume is waiting to be resynchronized.

Mirror Type

Default: 0

Range: 0-2

This value represents the type of mirroring this volume is engaged in. The following values are defined for this release:

0 None - The volume is not currently involved in a mirror.

1 Synchronous - Data is written to the target volume first.

2 Asynchronous - Data is written to the source volume first and queued to be sent to the target.

Network Number of Reconnects

Default: 0

Range: 0 - MAX_ULONG

This value is the number of network reconnections that have been made while the volume has been mirrored. A network reconnection occurs when communication is lost with the target.

Write Queue Counters

Queue Current Length

Default Value: 0

Range: 0 - <Queue High Water>

This value represents the current length, in terms of number of writes, of the SteelEye DataKeeper Asynchronous write queue for volumes currently being mirrored. Writes to the target system are pushed onto a queue and committed to the target at a later time, allowing writes to complete locally first.

Queue High Water

Default: 20000

Range: 0x4e20

This counter displays the Asynchronous write queue high water mark as set in the registry. During massive I/O traffic, if the Queue Current Length (above) reaches this value, the SteelEye DataKeeper driver will momentarily pause the mirror, drain the queue down to the Queue Low Water mark (below) and automatically start a partial resync.

Queue Low Water

Default: 150

Range: 0x96

This counter displays the Asynchronous write queue low water mark as set in the registry. During massive I/O traffic, if the queue length reaches the WriteQueueHighWater mark (above), the SteelEye DataKeeper driver will momentarily pause the mirror. When the queue length goes down to the Queue Low Water mark, the SteelEye DataKeeper driver will automatically start a partial resync.

Resynchronization Control Counters

Resync Current Block

Default: 0

Range: 0 - <Resync Total Blocks>

During the synchronization process, this value represents the current block that is being sent to the target. At other times (i.e. when mirror state is not EmMirrorStateResync), this value will be 0.

During synchronization, a given block may be sent to the target multiple times if writes are ongoing to the volume. This is based on the number of resync passes that are required.

Resync Dirty Blocks

Default Value: 0

Range: 0 - <Resync Total Blocks>

This value is the number of total blocks that are dirty during mirror resynchronization. "Dirty" blocks are those that must be sent to the target machine before synchronization is complete. This value will be 0 for all states other than EmMirrorStateResync.

When a mirror synchronization is begun, this value will be initially equal to the value of Resync Total Blocks. Please note that during a mirror synchronization, Resync Dirty Blocks may actually increase if a large number of incoming writes are made to the volume.

Resync Elapsed Time

Default Value: 0

Range: 0 - MAX_ULONG

While the mirror is being synchronized, this value represents the elapsed time in seconds that the synchronization has been occurring. After a mirror is successfully resynchronized, the value represents the total amount of time the previous synchronization operation took since the last system boot. The value will be 0 for volumes that either never have been synchronized or volumes that were not synchronized during the last boot.

Resync New Writes

Default: 0

Range: 0 - MAX_ULONG

This value represents the number of writes that have occurred on the volume since a synchronization operation has begun. This value will directly affect the number of dirty blocks, the number of passes required to synchronize the mirror and the amount of time the synchronization takes to complete.

Resync Pass

Default Value: 10

Range: 0 - MaxResyncPasses (Registry)

This value is the number of passes that have currently been made through the volume during the resynchronization process to update the target. The number of passes required to complete the synchronization process will increase based on the amount of writing that is being performed during synchronization. While writing to the source volume is allowed during synchronization, heavy writes will cause the synchronization to take longer, thus resulting in a much longer time until it is finished.

Resync Total Blocks

Default Value: 0

Range: 0 - MAX_ULONG

This value represents the number of 64k blocks used for resynchronization of the mirrored volume. The value is approximately equal to the file system size of the volume divided by 64K. Please note that the file system size is less than the partition size of the volume that is shown in the Windows Disk Management program. To see the file system size, type CHKDSK X: (where X is the drive letter).

Resync Phase

Default Value: 0

Range: 0 - 3

The mirror resync goes through a number of phases. Once the state is set to "resync", the phase is initialized to RESYNC_INITIAL_PHASE. If a full resync is to be performed, then it is transitioned to RESYNC_FULL_PHASE. If a partial resync is to occur, then this state is recorded as RESYNC_UPDATE_PHASE. The valid states are as follows:

RESYNC_NO_PHASE 0

RESYNC_INITIAL_PHASE 1

RESYNC_FULL_PHASE 2

RESYNC_UPDATE_PHASE 3

Chapter 2: Installation

For installation and licensing information, refer to the SteelEye Protection Suite for Windows Installation Guide on the SIOS Technical Doumentation site.

- http://docs.us.sios.com/#DK

Chapter 3: Configuration

Requirements/Considerations

The topics in this section identify several prerequisites to be aware of before implementing your DataKeeper configuration.

Sector Size

Beginning with DataKeeper Version 7.2.1, disks with sector size not equal to 512 bytes are supported. However, DataKeeper requires that the mirror source volume be configured on disk(s) whose sector size is the same as the disk(s) where the mirror target is configured. NTFS Metadata includes the disk sector size. DataKeeper replicates the entire NTFS file system from source to target, so the sector sizes must match.

Note: For DataKeeper Version 7.2 and prior, only disk devices whose sector size is the standard 512 bytes are supported.

Network Bandwidth

Because DataKeeper can replicate data across any available network, special consideration must be given to the question, "Is there sufficient bandwidth to successfully replicate the volume and keep the mirror in the **mirroring** state as the source volume is updated throughout the day?"

Keeping the mirror in the **mirroring** state is critical, because a switchover of the volume is not allowed unless the mirror is in the **mirroring** state.

Determine Network Bandwidth Requirements

Prior to installing SteelEye DataKeeper, you should determine the network bandwidth requirements for replicating your data. Use the method below to measure the rate of change for the data that you plan to replicate. This value indicates the amount of network bandwidth that will be required to replicate that data.

After determining the network bandwidth requirements, ensure that your network is configured to perform optimally. If your network bandwidth requirements are above your current available network capacity, you must consider one or more of the following options:

- Enable compression in DataKeeper, or in the network hardware, if possible
- Create a local, non-replicated storage repository for temporary data and swap files if you are replicating Hyper-V virtual machines

- Reduce the amount of data being replicated
- Increase your network capacity

If the network capacity is not sufficient to keep up with the rate of change that occurs on your disks, DataKeeper mirrors will remain in a resynchronizing state for considerable periods of time. During resynchronization, data on the target volume is not guaranteed to be consistent.

Measuring Rate of Change

Use Performance Monitor (perfmon) to measure the rate of change that occurs on your volumes that are to be replicated. The best way to do this is to create a log of disk write activity for some period of time (one day, for instance) to determine what the peak disk write periods are.

To track disk write activity,

- use perfmon to create a user-defined data collector set on Windows 2008 or a performance counter log on Windows 2003.
- add the counter "Disk Write Bytes/sec" for each volume - the volume counters can be found in the logical disks group.
- start the log and let it run for the predetermined amount of time, then stop and open the log.

An alternative to creating a log of disk writes is to use perfmon to track disk write bytes/sec interactively, in the Performance Monitor tool, and to observe the maximum and average values there.

SteelEye DataKeeper handles short bursts of write activity by adding that data to its async queue. However, make sure that over any extended period of time, the disk write activity for all replicated volumes combined remains, on average, below the amount of change that DataKeeper and your network can transmit.

SteelEye DataKeeper can handle the following average rates of change, approximately:

Network Bandwidth	Rate of Change
1.5 Mbps (T1)	182,000 Bytes/sec (1.45 Mbps)
10 Mbps	1,175,000 Bytes/sec (9.4 Mbps)
45 Mbps (T3)	5,250,000 Bytes/sec (41.75 Mbps)
100 Mbps	12,000,000 Bytes/sec (96 Mbps)
1000 Mbps (Gigabit)	65,000,000 Bytes/sec (520 Mbps)

Network Adapter Settings

DataKeeper requires that **"File and Printer Sharing for Microsoft Networks"** be enabled on the network interfaces to make a NAMED PIPE connection and be able to run DataKeeper's command line tool (EMCMD).

To test if you can make a Named Pipe connection, try to map a network drive on the TARGET system. If that fails, you have a Named Pipe issue.

DataKeeper also requires that **NetBIOS over TCP/IP** and **SMB** protocols be enabled. If the GUI does not operate correctly, make sure the following network configurations are enabled:

- Enable **NetBIOS over TCP/IP** and **SMB** protocols as in the following example:

```
My Computer->Manage->System Tools->Device Manager-
>View->Show Hidden Devices->Non-Plug and Play Drivers-
>NetBIOS over Tcpip (Enable)
```

- Enable **NetBIOS over TCP/IP** on each network adapter carrying mirror traffic as in the following example:

```
Start->Settings->Network and Dial-up Connections-><Your
Network Adapter>->Properties->Internet Protocol(TCP/IP)
->Properties->Advanced…button->WINS tab->Enable NetBIOS
over TCP/IP radio button (Checked)
```

- Enable the Microsoft "**Client for Microsoft Networks**" component on each system where the DataKeeper Administrator GUI will be used. This must be on the same adapter with **NetBIOS over TCP/IP** enabled (above). For example:

```
Start->Settings->Network and Dial-up Connections-><Your
Network Adapter>->Properties->Client for Microsoft
Networks(checked)
```

- Enable the Microsoft "**File and Printer Sharing for Microsoft Networks**" component on each system which the DataKeeper Administrator GUI will connect to locally and remotely. This must be on the same adapter with **NetBIOS over TCP/IP** enabled (above). For example:

```
Start->Settings->Network and Dial-up Connections-><Your
Network Adapter>->Properties->File and Printer Sharing
for Microsoft
```

DataKeeper Service Log On ID and Password Selection

During a new DataKeeper installation setup, the user will be prompted for a DataKeeper Service Log On ID and Password.

The DataKeeper Service uses authenticated connections to perform volume switchovers and make mirror role changes across multiple servers. The Log On ID account chosen to run the DataKeeper Service will determine how much authority and permission is available to establish connections between servers and perform volume switchovers, especially when server or network disruptions occur.

Several types of Service Log On ID accounts are available as follows:

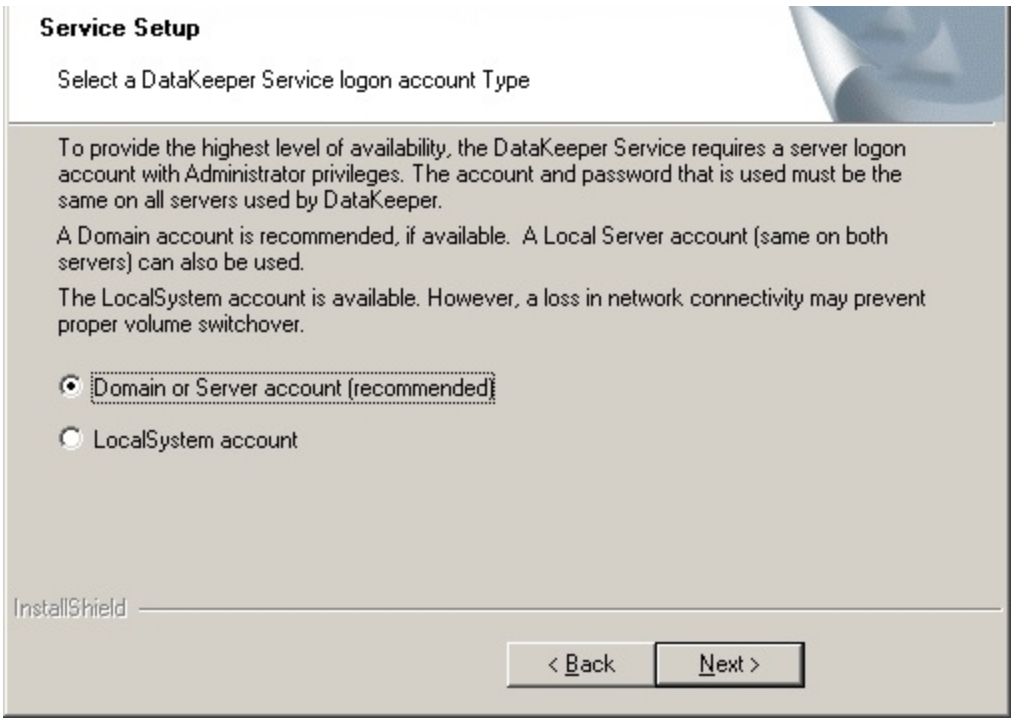
- A **Domain Account** with administrator privileges, valid on all connected servers in the domain (*recommended*)
- A **Server Account** with administrator privileges, valid on all connected servers
- The **Local System Account** (*not recommended*)

Note: For Workgroups, use the **Server Account** option and use the server name \ administrator on each system as the Service Account for DataKeeper. **You should also log on to all servers using this same Log On ID and Password** (see related Known Issue).

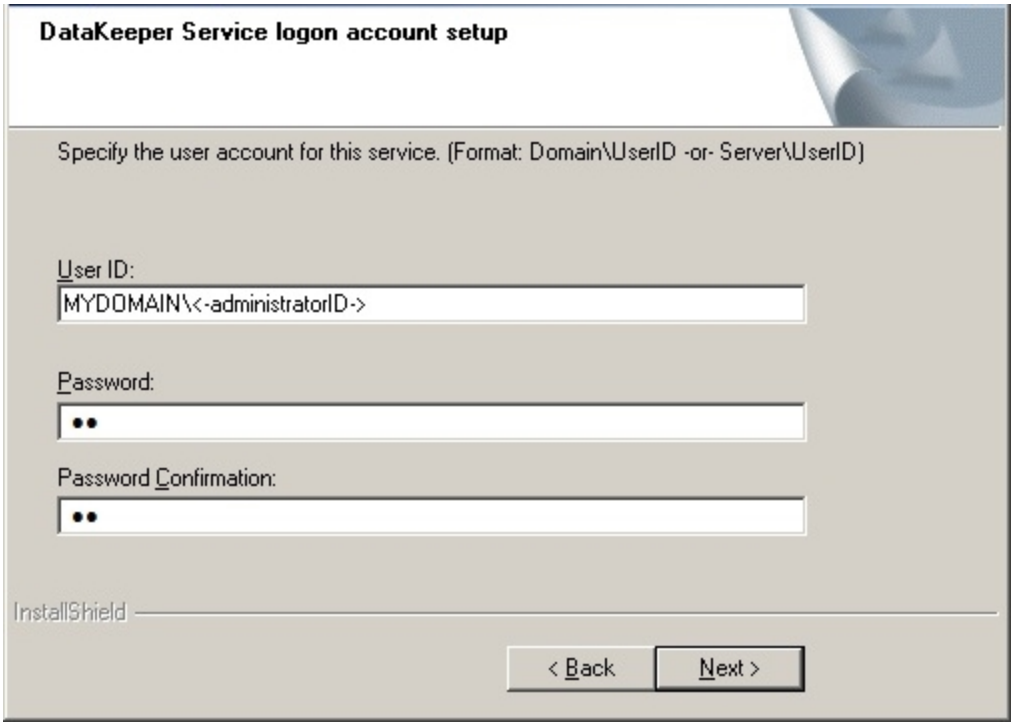
Note: The domain or server account used must be added to the Local System Administrators Group. The account must have administrator privileges on each server in which DataKeeper is installed.

Please note that the Local System account cannot be authenticated properly in a domain when network connectivity with Active Directory is lost. In that situation, connections between servers cannot be established with the Local System account causing DataKeeper volume switchover commands, via the network, to be rejected. IT organizations requiring fault tolerance during a disaster recovery, including network disruptions, should not use the Local System account.

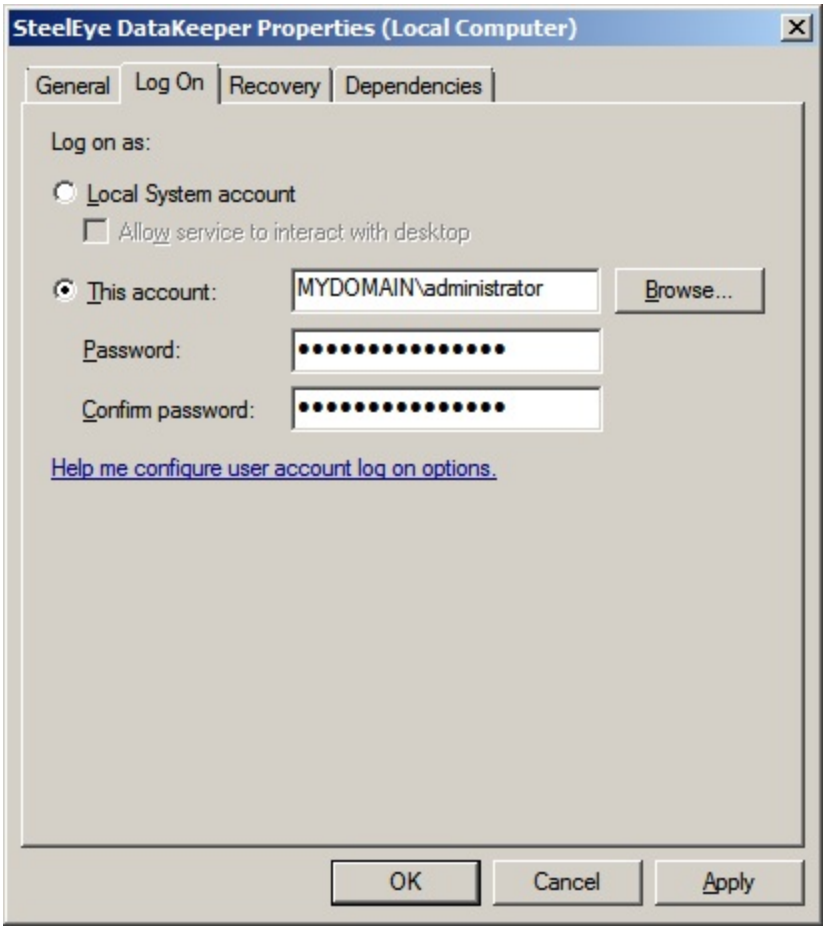
DataKeeper Installation – Service Logon ID Type Selection:



If a Domain or Server account is selected above, the DataKeeper Service Log On ID and Password Entry Form is displayed to enter that information.



If the DataKeeper Service has previously been configured with a Service Log On ID and Password, the setup program will omit the Service ID and Password selection dialogs. However, at any time, an administrator can modify the DataKeeper Service Log On ID and Password using the Windows Service Applet. Be sure to restart the DataKeeper Service after changing the Log On ID and/or Password.



The following table outlines these requirements:

Environment	DataKeeper Service Requirements	DataKeeper UI Requirements
Same Domain or Trusted Domain Environment	• Run the DK Service on all systems as the same account with the same credentials • Okay to use the default = Local System Account	• Log in as a domain admin and run the DK GUI • Or use “run as” Administrator option to run DK GUI

Environment	DataKeeper Service Requirements	DataKeeper UI Requirements
Mixed Environment Servers in a Mixture of Domain and WorkGroup or Servers in Separate Domains	• Create a local account on each system with same account name and password • Add this local account to the Administrator Group • Run the DK Service on all systems with the local account	• Log in using the local account you created to run the DK Service • Run the DK GUI **You should also log on to all servers using this same Log On ID and Password** (see related Known Issue).

Firewall Configurations

SteelEye DataKeeper cannot function properly if the firewall settings for the source and target machines are not configured correctly. This means you will need to configure a rule for inbound and outbound connections on each server running SteelEye DataKeeper as well as any network firewalls that replication traffic must traverse.

During installation of SteelEye DataKeeper, you will be prompted to allow the installer to configure your firewall rules needed by DataKeeper as well as to configure other system settings that are required by DataKeeper on Windows 2008. This is not necessary for Windows 2003. If you choose to allow the installer to make these changes, you will not need to configure your firewall manually. If you choose not to allow the installer to make these changes, you will need to configure your system manually as described in this section.

The ports that are required to be open for replication are as follows: 137, 138, 139, 445, 9999, plus ports in the 10000 to 10025 range, depending upon which volume letters you plan on replicating. The chart below shows the additional ports which must be open depending upon which drive letters you plan on replicating.

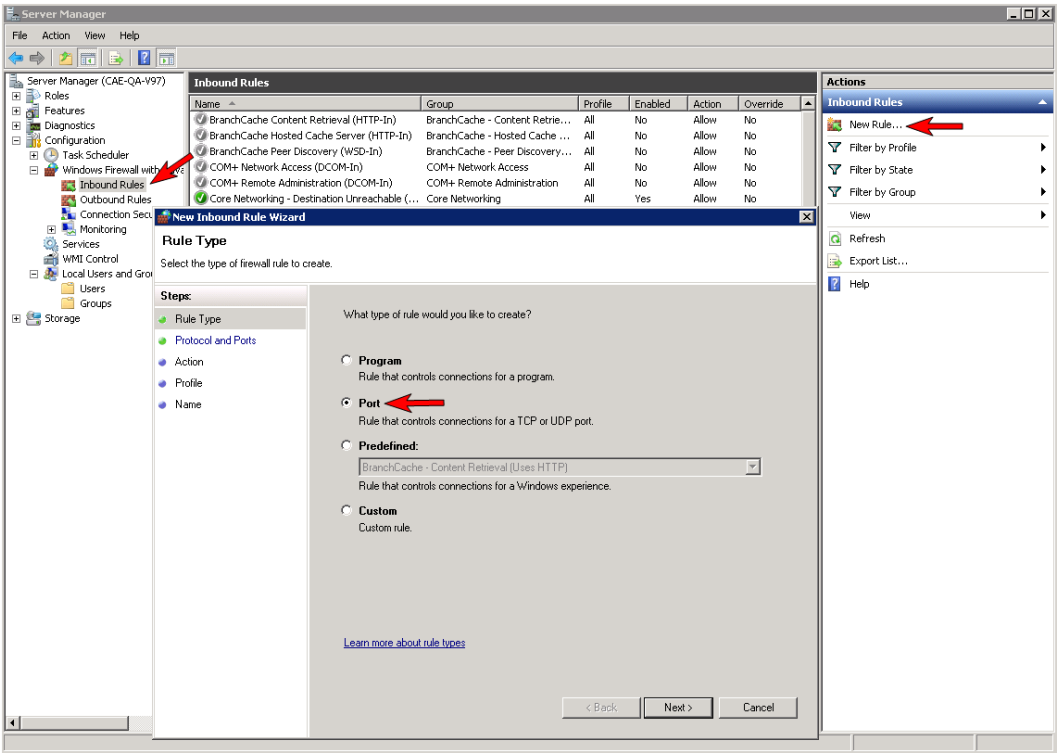
Port #	Volume Letter	Port #	Volume Letter
10000	A	10013	N
10001	B	10014	O
10002	C	10015	P
10003	D	10016	Q
10004	E	10017	R
10005	F	10018	S
10006	G	10019	T
10007	H	10020	U
10008	I	10021	V

Port #	Volume Letter	Port #	Volume Letter
10009	J	10022	W
10010	K	10023	X
10011	L	10024	Y
10012	M	10025	Z

Configuring Microsoft's Windows Firewall with Advanced Security - Example

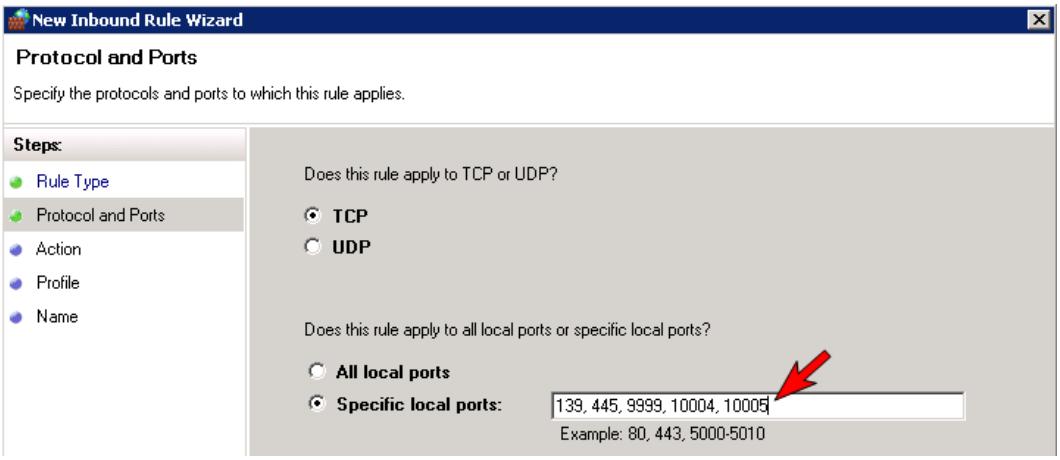
The exact steps required to configure the firewall for each cluster is as varied as each possible cluster configuration, but the following procedure and screen shots will give you one example to follow when using SteelEye DataKeeper to replicate the E: and F: volumes. Note the Port # and Volume Letter table listings in the previous section.

1. Open Microsoft's **Windows Server Manager** and select **Inbound Rules** to create a rule for the TCP protocol as well as the UDP protocol.
2. Select **New Rule** from the **Actions** panel in the right column of the window. Select **Port** as the type of rule to be created. Select **Next**.

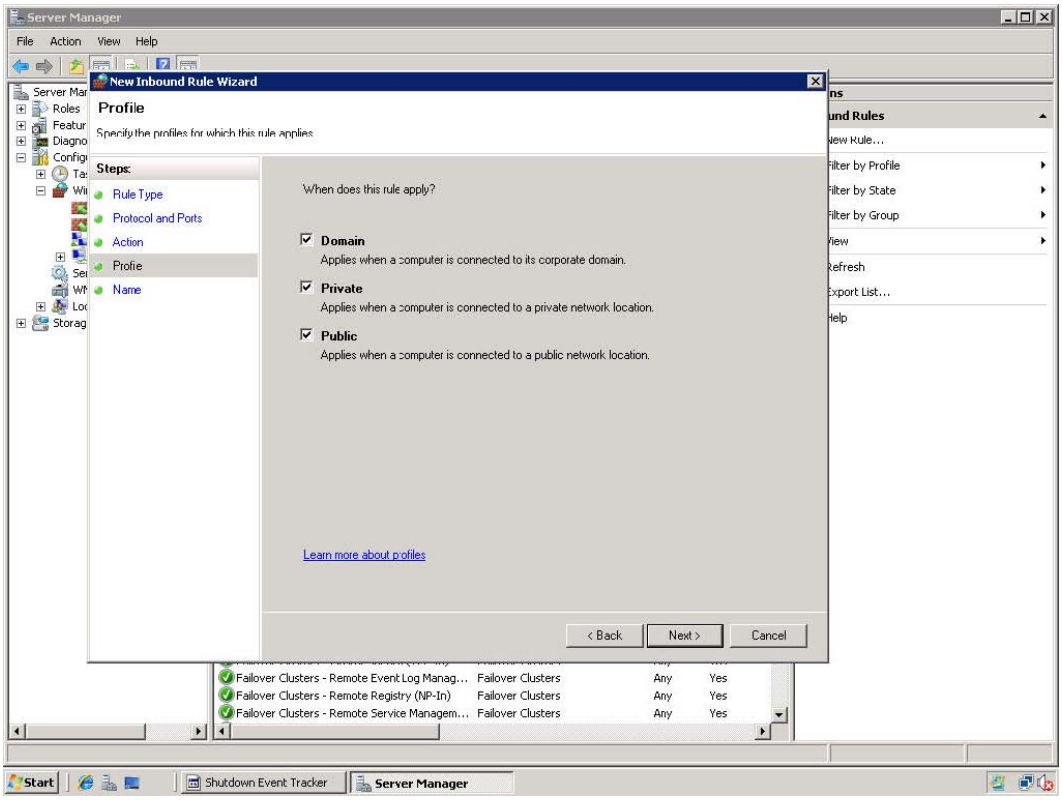


3. Select **TCP** for the type of protocol impacted by this rule. Select the **Specific local ports** button and

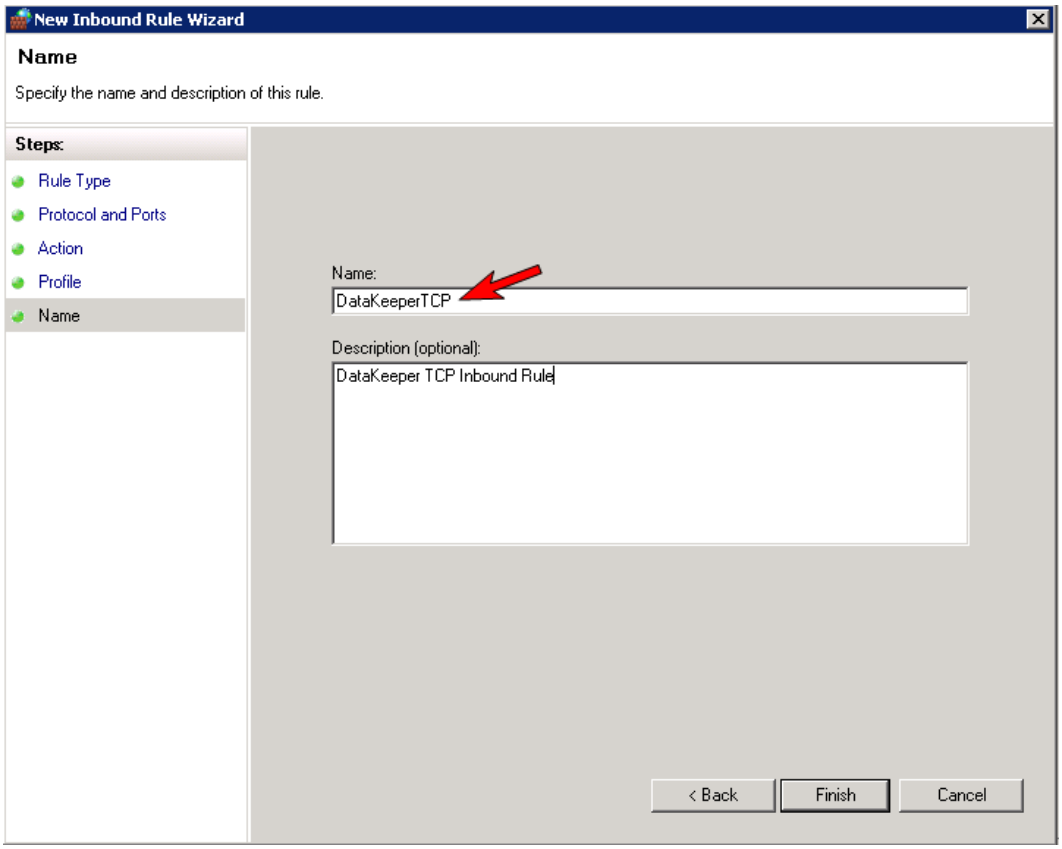
enter the following ports: **139**, **445**, **9999**, **10004** (for the E drive) and **10005** (for the F drive). Select **Next**.



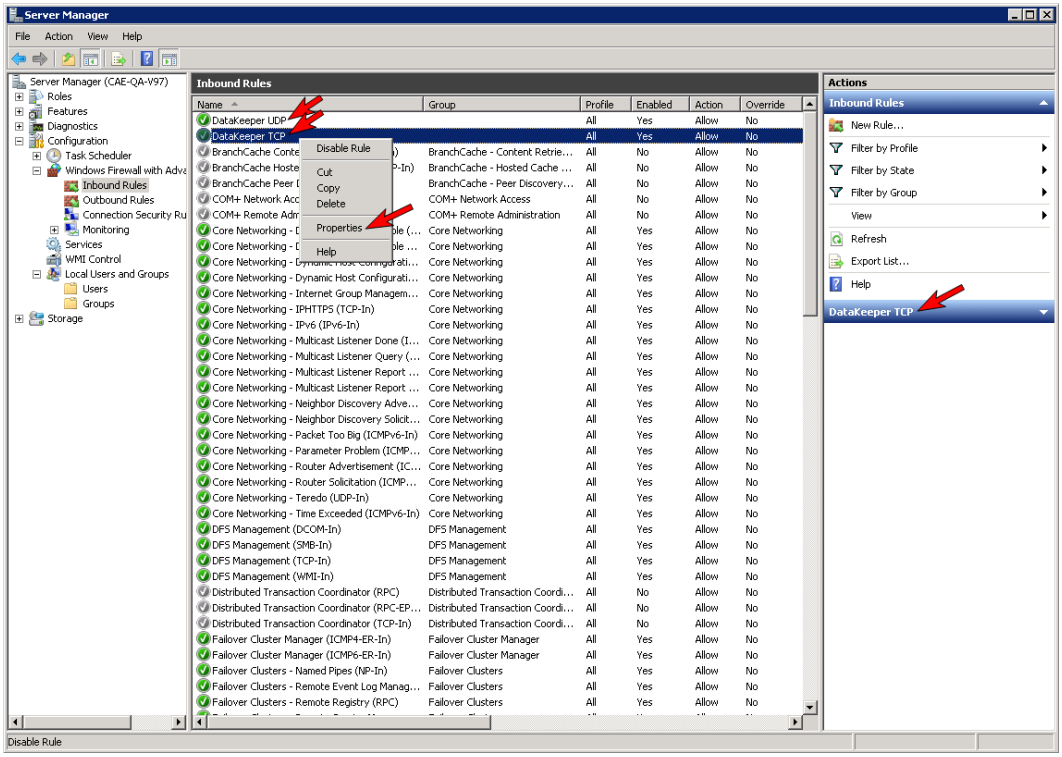
4. For the action, select **Allow the Connection**. Select **Next**.
5. For the profile, select **Domain**, **Private** and **Public** for the conditions when this rule applies. Select **Next**.



6. Enter a **Name** and **Description** for the new **Inbound Rule** and select **Finish**.



7. Select **New Rule** again to create the rule for **UDP protocol**. Select **Port** as the type of rule to be created. Select **Next**.
8. Select **UDP** for the type of protocol impacted by this rule. Select the **Specific local ports** button and enter the following ports in the Specific local ports field: **137, 138**. Select **Next**.
9. For the action, select **Allow the Connection**. Select **Next**.
10. For the profile, select **Domain**, **Private** and **Public** for the conditions when this rule applies. Select **Next**.
11. Enter a **Name** and **Description** for the new **Inbound Rule** and select **Finish**.
12. Your new DataKeeper rules will appear in the **Inbound Rules list** and the **Action** panel column. You can select the DataKeeper rule in the center panel and click the right mouse button to view the rule **Properties**.



High-Speed Storage Best Practices

Configure Bitmaps

If the DataKeeper default bitmap location (`%ExtMirrBase%\Bitmaps`) is not located on high-speed storage, you should move the bitmaps to a high-speed storage device in order to eliminate I/O bottlenecks with bitmap access. To do this, allocate a small disk partition, located on the high-speed storage drive, on which to place the bitmap files. Create the folder in which the bitmaps will be placed, and then Relocate the bitmaps (intent logs) to this location.

Disk Partition Size

The disk partition size must be big enough to contain all bitmap files for every mirror that will exist on your system. Each bit in the DataKeeper bitmap represents 64 KB of space on the volume, so to determine the bitmap size for a bitmap file, use the following formula:

```
<volume size in bytes> / 65536 / 8
```

Example:

For a 765 GB volume, convert the 765 GB to bytes

```
765 * 1,073,741,824 = 821,412,495,360
```

```
bytes
```

Divide the result by 64K (65,536 bytes) to get the number of blocks/bits

```
821,412,495,360 / 65,536 = 12,533,760
blocks/bits
```

Divide the resulting number of blocks/bits by 8 to get the bitmap file size in bytes

```
12,533,760 / 8 = 1,566,720
```

So a mirror of a 765 GB volume would require 1,566,720 bytes for its bitmap file, or approximately 1.5 MB.

A simple rule of thumb to use is that each GB of disk space requires 2 KB of bitmap file space.

Remember to reserve room for all mirror targets (if you have multiple target systems, each one needs a bitmap file). Also remember to reserve room for all mirrored volumes.

Increase the `WriteQueueLowWater` Tunable

In order to increase throughput during a mirror Resynchronization operation, you may need to increase the `WriteQueueLowWater` value for the mirror. If your mirror is currently in the Mirroring state, you can safely change this value without pausing the mirror.

If your mirror is currently resyncing, you can pause the mirror and change the `WriteQueueLowWater` value to any value less than the mirror's `WriteQueueHighWater` value. The default `WriteQueueLowWater` value is 150. Raising it to 2000 commonly improves resynchronization speed. Each environment is different, so you may find that a certain `WriteQueueLowWater` setting gives you the best resynchronization throughput. To change the value, follow these steps:

1. Start on the mirror source server. If resyncing, pause the mirror.
2. Via your registry, go to the following:

   ```
   HKEY_LOCAL_
   MACHINE\SYSTEM\CurrentControlSet\services\ExtMirr\Paramet
   ers\Volumes\{GUID-for-VOLUME}\Targets\<IP-Address>
   ```

3. Right-click on the `WriteQueueLowWater` entry and Select **Modify**.
4. Change the **Base** value to **Decimal**.
5. Change **Value** data from 150 to 2000.

- Update these registry changes in order to pick up and use these new values via the `READREGISTRY` command.
- Make the above changes on all other targets/systems in the cluster.

Handling Unmanaged Shutdown Issues

Unmanaged shutdowns due to power loss or other circumstances force a consistency check during the reboot. This may take several minutes or more to complete and can cause the drive not to reattach and can cause a dangling mirror. Use the ioAdministrator console to re-attach the drives or reboot the system again and make sure the check runs. For further information, refer to the ***ioXtreme User Guide for Windows***.

Other Recommendations/Suggestions

- Check the Network Interface configuration settings. Increasing the Receive and Transmit buffers on the interfaces often improves replication performance. Other settings that may also affect your performance include: Flow Control, Jumbo Frames and TCP Offload. In certain cases, disabling Flow Control and TCP Offload can result in better replication performance. Enabling larger ethernet frames can also improve throughput.
- Check the location of the NICs on the bus (the slot that they're physically plugged into) as this can also affect the speed.
- Use Iometer, an I/O subsystem measurement and characterization tool available free on the internet, to test network throughput. Iometer can be set up in a client/server configuration and can test network throughput directly. Another alternative is to set up a file share using the replication IP address, and then copy large amounts of data over that share while monitoring the network throughput using `Perfmon` (Network Interface / Bytes Sent Per Second) or the Task Manager "Networking" tab.
- Make sure you have the latest drivers and firmware for the network adapters.

WAN Considerations

Replicating data across the network to a remote server located miles away from the source server is the most common use of DataKeeper. Typically, this configuration relies on a WAN of some sort to provide the underlying network that DataKeeper uses to replicate the data. If the bandwidth of the WAN is limited, there are a number of additional factors to consider including:

- Initial Synchronization of Data Across the LAN/WAN
- Compression
- Bandwidth Throttle

Initial Synchronization of Data Across the LAN or WAN

When replicating large amounts of data over a WAN link, it is desirable to avoid full resynchronizations which can consume large amounts of network bandwidth and time. DataKeeper avoids almost all full resyncs by using its bitmap technology. However, the initial synchronization of the data, which occurs when the mirror is first created, cannot be avoided.

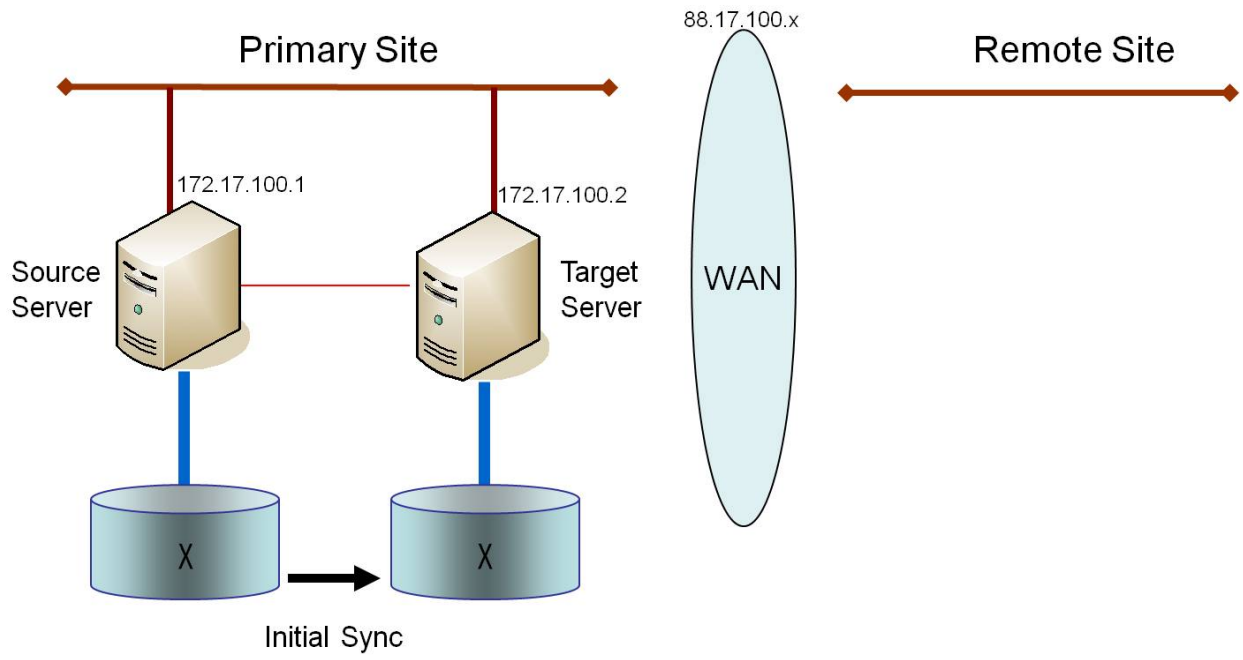
In WAN configurations, one way to avoid the initial full synchronization of data across the WAN is to configure both systems on a LAN, create the mirror and allow the initial full synchronization to occur across the LAN. Once the initial synchronization is complete, update the IP addresses for the source and target, which will place the mirror in the **Paused** state. Move the target system to its new location. Once the target system is in place, power it on and verify all network settings, including the IP address that was updated. On the source system, run the **CHANGEMIRRORENDPOINTS*** command. The mirror will be **CONTINUED** and only a partial resync (the changes that have occurred on the source volume since the mirror was **PAUSED**) of the data is necessary to bring the TARGET volume in sync with the SOURCE.

***Note: This command supports changing the endpoints of a mirrored volume that is configured on 3 nodes or fewer. For configurations greater than three nodes, create mirrors with the final endpoint at the local site and use route adds to get the mirrors created and resynced before moving the server to the final location/address/DR site.**

Example

In the example below, a mirror is created locally in the primary site, and then the target will be moved to remote site. The source server is assigned the IP address 172.17.100.1, and the target server is assigned the IP address 172.17.100.2. The WAN network IP is 88.17.100.x,

- Using the DataKeeper UI, create a mirror on Volume X from 172.17.100.1 to 172.17.100.2. **Note**: Connecting to the target by name is recommended so DNS name resolution later will automatically resolve to the new IP address.



Once the initial sync of the data is complete,

- Update the IP address for the network adapter for the source to 88.17.100.1 and update the IP address for the network adapter on the target to 88.17.200.2. This will place the mirror on the source side into the PAUSED state.

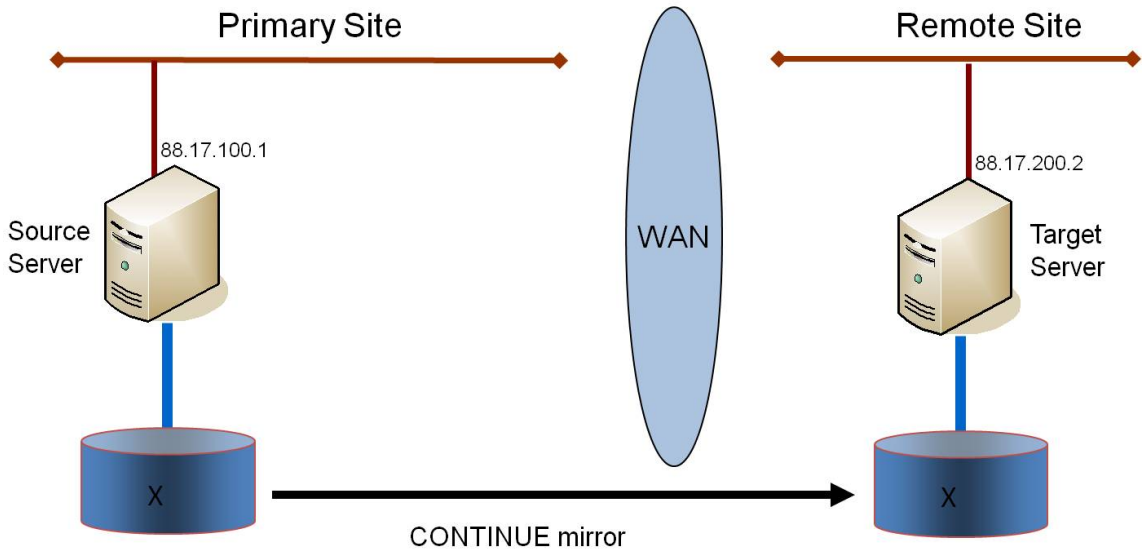
- Ship the target machine to its new location.
- Power on the target machine and verify all network settings, including the IP address updated above.
- On the source system, open a DOS command window and change directory to the DataKeeper directory by executing the following command:

```
cd %EXTMIRRBASE%
```

- Run the following command to update the existing mirror endpoints to the new IP addresses:

```
EMCMD 172.17.100.1 CHANGEMIRRORENDPOINTS X 172.17.100.2 88.17.100.1
88.17.200.2
```

- DataKeeper will resync the changes that have occurred on the source server while the target server was unreachable.
- When this partial resync is complete, the mirror will change to the **MIRRORING** state.



Verifying Data on the Target Volume

By design, DataKeeper locks the target volume. This prevents the file system from writing to the target volume while the replication is occurring. However, DataKeeper does provide a mechanism to unlock the target volume and allow read/write access to it while the mirror is still in place. There are two methods to do this:

1. Pause the mirror and unlock the target volume via the Pause and Unlock mirror option in the DataKeeper UI.
2. Use the DataKeeper command line interface (EMCMD) to pause the mirror (PAUSEMIRROR) and unlock the target volume (UNLOCKVOLUME).

Once unlocked, the target volume is completely accessible. When finished inspecting the target volume, be sure to continue the mirror to re-lock the target volume and allow DataKeeper to resync any changes that occurred on the source volume while the mirror was paused. Any writes made to the target volume while it was unlocked will be lost when the mirror is continued.

WARNING: If a reboot is performed on the target system while the target volume is unlocked, a full resync will occur when the target system comes back up.

Compression

DataKeeper allows the user to choose the compression level associated with each mirror. Enabling compression can result in improved replication performance, especially across slower networks. A compression level setting of 3-5 represents a good balance between CPU usage and network efficiency based on the system, network and workload.

Note: The compression level of a mirror can be changed after the mirror is created. See the topic "Changing the Compression Level of an Existing Mirror".

Bandwidth Throttle

DataKeeper attempts to utilize all of the available network bandwidth. If DataKeeper is sharing the available bandwidth with other applications, you may wish to limit the amount of bandwidth DataKeeper is allowed to use. DataKeeper includes a feature called **Bandwidth Throttle** that will do this. The feature is enabled via a registry setting.

Note: For additional information on both **Compression** and **Bandwidth Throttle**, see the topics below.

- Registry Entries
- Changing the Compression Level of an Existing Mirror

Chapter 4: Administration

The topics in this section provide detailed instructions for performing DataKeeper administration tasks.

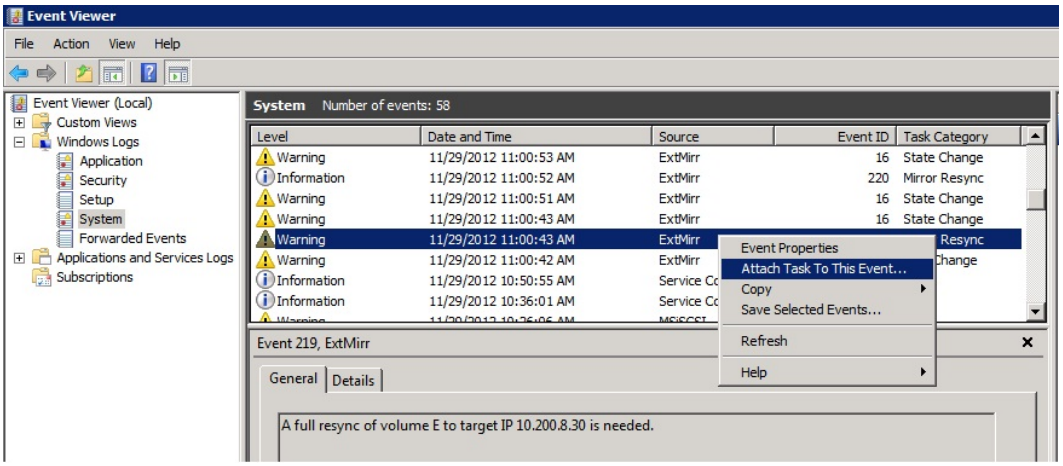
DataKeeper Event Log Notification

The **Event Log notification** is a mechanism by which one or more users may receive email notices when certain events occur. The **Windows Event Log** can be set up to provide notifications of certain DataKeeper events that get logged.

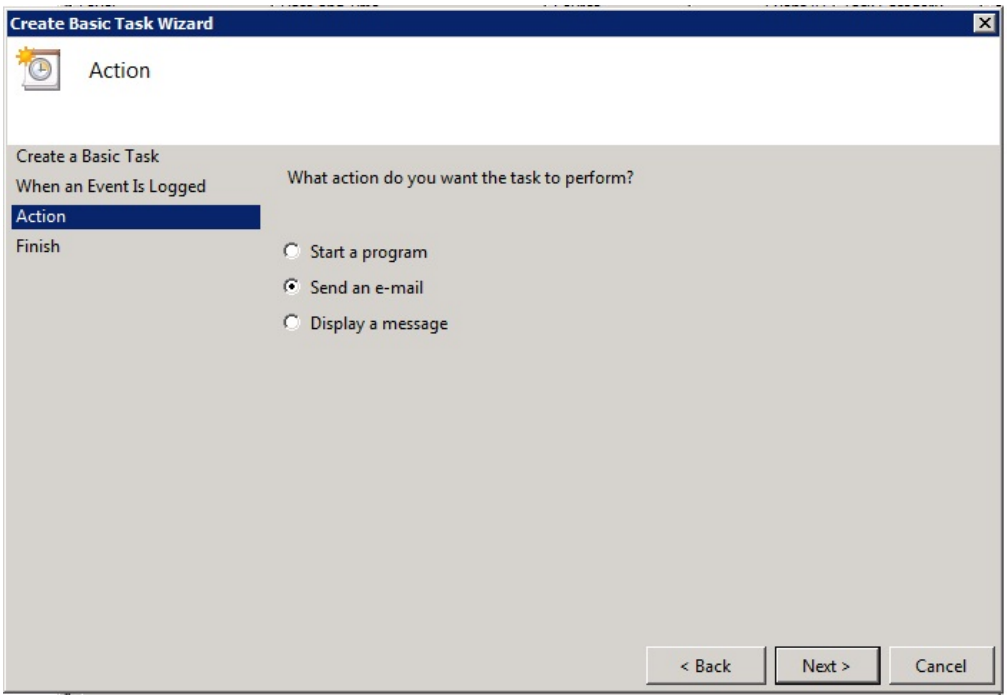
Note: This option is only available for **Windows Server 2008 R2**.

To set up the **Windows Event Log email task** for DataKeeper events, perform the following steps:

1. Open **Event Viewer**, go to the **System** or **Application** log and highlight the event in which you want to be notified.
2. Right-click the event and select **Attach Task To This Event…**



3. Follow the ***Task Wizard*** directions, choosing the **Send an e-mail** option when prompted and filling in the appropriate information.



4. When you click **Finish** at the end of the ***Task Wizard***, the new task will be created and added to your Windows schedule.

   Note: These email tasks will need to be set up on each node that will generate email notification.

Primary Server Shutdown

On a graceful shutdown of the source server, all pending writes to the target are completed. This ensures that all data is present on the target system.

On an unexpected source server failure, the Intent Log feature eliminates the need to do a full resync after the recovery of the source server. If the Intent Log feature is disabled or if SteelEye DataKeeper detected a problem accessing the volume's Intent Log file, then a full resync will occur after the source server is restored to service.

Secondary Server Failures

In the event there is a failure affecting the secondary (target) system, the affected mirror is marked **Paused**. It is necessary to correct the condition that caused the secondary to fail and then resync the volumes. There are no write attempts made to the target after the secondary server fails.

When the secondary server comes back online after a failure, the source side of the mirror will automatically reconnect to the target side of the mirror. A partial resync follows.

Extensive Write Considerations

SteelEye DataKeeper allows users to access the source during the creation and resync process. Extensive writes during the create or resync process increase the amount of time required to complete the operation.

The user can also increase the MaxResyncPasses registry value to allow the resynchronization process to finish even when the source volume is being accessed continuously.

CHKDSK Considerations

If you must run `CHKDSK` on a volume that is being mirrored by SteelEye DataKeeper, it is recommended that you first **pause** the mirror. After running `CHKDSK`, **continue** the mirror. A partial resync occurs (updating those writes generated by the `CHKDSK`) and mirroring will continue.

Failure to first **pause** the mirror may result in the mirror automatically entering the **Paused** state and performing a **Resync** while `CHKDSK` is in operation. While this will not cause any obvious problems, it will slow the `CHKDSK` down and result in unnecessary state changes in SteelEye DataKeeper.

SteelEye DataKeeper automatically ensures that volumes participating in a mirror, as either source or target, are not automatically checked at system startup. This ensures that the data on the mirrored volumes remains consistent.

Note: The bitmap file (for non-shared volumes) is located on the C drive which is defined by BitmapBaseDir as the default location. Running CHKDSK on the C drive of the **Source** system will cause an error due to the active bitmap file. Therefore, a switchover must be performed so that this Source becomes Target and the bitmap file becomes inactive. The CHKDSK can then be executed on this system as the new target (original source).

DKSUPPORT

`DKSUPPORT` .cmd, found in the *<DataKeeper Installation Path>\SUPPORT* directory, is used to collect important configuration information and event log files and put them in a zip file. SIOS Support Engineers will commonly request this zip file as part of the support process. To run this utility, double-click the file `DKSUPPORT` from the explorer window or follow the procedures below to run it from a command prompt.

• Open a command prompt

• Type "`cd %extmirrbase%`"

• You will now be placed in the DataKeeper directory or c:\Program Files (x86) \SteelEye\DataKeeper

• From the aforementioned directory type "`cd support`"

• From within the support directory, execute the following command "`dksupport.cmd`"

• Run this command on all systems that are participating in DataKeeper mirroring

The zip file will be created in the same Support directory, and can either be emailed to support@us.sios.com or File transferred (FTP) to support engineering

Note: This command may take some time to execute.

Event Log Considerations

It is important that SteelEye DataKeeper be able to write to the Event Log. You should ensure that the Event Log does not become full. One way to accomplish this is to set the Event Log to overwrite events as needed:

1. Open the **Event Log**.
2. Right-click on **System Log** and select **Properties**.
3. Under **Log Size**, select **Overwrite Events as Needed**.

Using Disk Management

When using the Windows Disk Management utility to access SteelEye DataKeeper volumes, please note the following:

- Using Disk Management to delete partitions that are being mirrored is not supported. Deleting a partition that is part of a SteelEye DataKeeper mirror will yield unexpected results.
- Using Disk Management to change the drive letter assigned to a partition that is a part of a SteelEye DataKeeper mirror is not supported and will yield unexpected results.
- The Windows Disk Management utility will take longer to start on the target node based on the number of drives. Because the Windows operating system has error condition retries built in when a volume is locked, the speed with which it starts on the "locked" target node is affected.

Registry Entries

The following registry entries are associated with the SteelEye DataKeeper service or driver and can be viewed using `Regedt32`. The first section contains entries that may be modified; the second section contains entries that are for viewing only and should not be modified.

Registry Entries that MAY be Modified

HKEY_LOCAL_MACHINE\SYSTEM

\CurrentControlSet

\Services

\ExtMirr

\Parameters

\Volumes

\{Volume GUID}

\Targets

\{Target IP}

The SteelEye DataKeeper driver uses the Parameters key and those below it. The values within the Parameters key (denoted with *) are global for all volumes on the system. The values under each of the Target IP registry keys (denoted with †) are specific to a mirror only. Values denoted with both * and † appear under both keys. (The target-specific value overrides the global value in this case.)

BandwidthThrottle †

Location: *HKEY_LOCAL_MACHINE\SYSTEM\<CurrentControlSet>\Services\ExtMirr\Parameters\Volumes\{Volume GUID}\Targets\{Target IP}\BandwidthThrottle*		
Name	Type	Default Data
BandwidthThrottle	REG_DWORD	0
Specifies the maximum amount of network bandwidth (in kilobits per second) that a particular mirror is allowed to use. A value of 0 means unlimited.		

BitmapBaseDir*

Location: *HKEY_LOCAL_MACHINE\SYSTEM\<CurrentControlSet>\Services\ExtMirr\Parameters\BitmapBaseDir*		
Name	Type	Default Data
BitmapBaseDir	REG_SZ	*C:\%EXTMIRRBASE%\Bitmaps* (usually *C:|Program Files\SteelEye\DataKeeper\Bitmaps* but may be different when upgrading a system or if you install SteelEye DataKeeper to a different path)
Specifies a directory where SteelEye DataKeeper stores its Intent Log files. (**Note**: The drive letter must be in uppercase.) To disable the intent log feature, clear this registry entry (set it to an empty string) on all current and potential mirror endpoint servers. **Disabling the intent log requires a reboot on each of these systems in order for this setting to take effect.**		

CompressionLevel †

Location: *HKEY_LOCAL_MACHINE\SYSTEM\<CurrentControlSet>\Services\ExtMirr\Parameters\Volumes\{Volume GUID}\Targets\{Target IP}\CompressionLevel*		
Name	Type	Default Data
CompressionLevel	REG_DWORD	0

Specifies the compression level for the given mirror. Valid values are 0 to 9. Level 0 is "no compression". Values from 1 to 9 specify increasingly CPU-intensive levels of compression. Compression level 1 is a "fast" compression - it does not require as much CPU time to compress the data, but results in larger (less compressed) network packets. Level 9 is the maximum amount of compression - it results in the smallest network packets but requires the most CPU time. The level can be set to somewhere in between, to balance CPU usage and network efficiency based on your system, network and workload.

DontFlushAsyncQueue *

Location: *HKEY_LOCAL_MACHINE\SYSTEM\<CurrentControlSet>\Services\ExtMirr\Parameters\DontFlushAsyncQueue*		
Name	Type	Default Data
DontFlushAsyncQueue	REG_SZ	**empty** <drive letter> [<drive letter>]
Allows the user to specify a volume or volumes that should not flush their async queues when the driver receives a flush request. This value should contain the drive letter(s) of the volume(s) to which this applies. Drive letters may be adjacent to each other (i.e. XY), or space separated (i.e. X Y), with no colons. After updating this registry value, execute the READREGISTRY command so that DataKeeper immediately starts using the new value. **(Note: When setting DontFlushAsyncQueue, data and database logs should be on the same partition.)**		

PingInterval *

Location: *HKEY_LOCAL_MACHINE\SYSTEM\<CurrentControlSet>\Services\ExtMirr\Parameters\PingInterval*		
Name	Type	Default Data
PingInterval	REG_DWORD	3000 (0xBB8)
Specifies the interval in milliseconds between pings. Use a higher value for Wide Area Networks (WANs) or unreliable networks. Along with the **MaxPingMisses**, you may customize them to adjust mirroring to the network performance.		

MaxResyncPasses *

Location: *HKEY_LOCAL_MACHINE\SYSTEM\<CurrentControlSet>\Services\ExtMirr\Parameters\MaxResyncPasses*		
Name	Type	Default Data
MaxResyncPasses	REG_DWORD	200 (0xc8)

Specifies the maximum number of resync passes before SteelEye DataKeeper will give up trying to resynchronize the mirror while there is traffic on the source volume. In every pass, SteelEye DataKeeper marks the volume blocks that were written to during the pass. In the next pass, it will send to the target only the marked blocks.

TargetPortBase *

Location: *HKEY_LOCAL_MACHINE\SYSTEM\<CurrentControlSet>\Services\ExtMirr\Parameters\TargetPortBase*		
Name	Type	Default Data
TargetPortBase	REG_DWORD	10000
Specifies the base TCP port number for target volume connections. This number may need to be adjusted if the default port is used by another service or is blocked by a firewall. The actual port that the target listens on is calculated as follows: Port = **TargetPortBase** + (Volume Letter - A:) For example: **TargetPortBase** = 10000 Volume Letter = H Port = 10000 + (H: -A:) = 10007		

TargetPortIncr *

Location: *HKEY_LOCAL_MACHINE\SYSTEM\<CurrentControlSet>\Services\ExtMirr\Parameters\TargetPortIncr*		
Name	Type	Default Data
TargetPortIncr	REG_DWORD	256
Specifies the increment to the base TCP port number. This is used only when a TCP port is found to be in use. For example, if the target is attempting to listen on port 10005 and that port is in use, it will retry listening on port 10005 + **TargetPortIncr**.		

TargetDispatchPort * †

Locations:

On Target System:
HKEY_LOCAL_MACHINE\SYSTEM\<CurrentControlSet>\Services\ExtMirr\Parameters\TargetDispatchPort

On Source System Creating Mirror to Above Target:
HKEY_LOCAL_MACHINE\SYSTEM\<CurrentControlSet>\Services\ExtMirr\Parameters\TargetDispatchPort

AND

HKEY_LOCAL_MACHINE\SYSTEM\<CurrentControlSet>\Services\ExtMirr\Parameters\Volumes\{Volume GUID}\Targets\(Target IP}\TargetDispatchPort

Name	Type	Default Data
TargetDispatchPort	REG_DWORD	9999

There are two places where this should be set if you are changing the dispatch port from 9999. On the target system, place it in the *ExtMirr\Parameters* key. The new setting will apply to all existing and new targets on that server. **A target reboot is required when the target Parameters key has been changed for this setting to take effect**. On any source system that will be creating the mirror to this target, place it in the *ExtMirr\Parameters* key and also in the *ExtMirr\Parameters\Targets\{TargetIP}* key if the mirror already exists. **Note**: Make sure the ports are the SAME on both the source and the target.

A firewall port must also be opened manually on all source and target servers for the new dispatch port to work.

WriteQueueHighWater * †

Note: Do not change this value while the mirror is actively being resynced; you must pause the mirror prior to changing this value.

Locations:

For New Mirrors*:*
HKEY_LOCAL_MACHINE\SYSTEM\<CurrentControlSet>\Services\ExtMirr\Parameters\WriteQueueHighWater

AND

For Existing Mirrors:
HKEY_LOCAL_MACHINE\SYSTEM\<CurrentControlSet>\Services\ExtMirr\Parameters\Volumes\{Volume GUID}\Targets\{Target IP}\WriteQueueHighWater

Note: If editing this entry under **Parameters**, all NEW mirrors created will inherit this value. If editing this entry under **Target**, the value pertains to that one Target only. ***Any Target values override Parameter values***.

Name	Type	Default Data
WriteQueueHighWater	REG_DWORD	20000 (0x4e20)

Specifies the high water mark of the asynchronous write queue. During intensive I/O traffic, if the queue length reaches this value, the SteelEye DataKeeper driver momentarily pauses the mirror, drains the queue and automatically starts a partial resync. This value represents the number of write requests in the queue, not the number of bytes. After updating this registry value, execute the READREGISTRY command so that DataKeeper immediately starts using the new value.

Note: This value depends on the available memory in the system. You can monitor the mirroring behavior using the SteelEye DataKeeper Performance Monitor counters and set this value accordingly.

WriteQueueLowWater*†

Note: Do not change this value while the mirror is actively being resynced; you must pause the mirror prior to changing this value. Do not set this value higher than WriteQueueHighWater.

Locations:

For New Mirrors*:*
HKEY_LOCAL_MACHINE\SYSTEM\<CurrentControlSet>\Services\ExtMirr\Parameters\WriteQueueLowWater

AND

For Existing Mirrors:
HKEY_LOCAL_MACHINE\SYSTEM\<CurrentControlSet>\Services\ExtMirr\Parameters\Volumes\{Volume GUID}\Targets\{Target IP}\WriteQueueLowWater

Note: If editing this entry under **Parameters**, all NEW mirrors created will inherit this value. If editing this entry under **Target**, the value pertains to that one Target only. ***Any Target values override Parameter values***.

Name	Type	Default Data
WriteQueueLowWater	REG_DWORD	150 (0x96)
This value is used to specify the maximum number of resync block requests that can be queued up at the same time. Changing this value may change the speed of mirror resynchronizations - a higher value often leads to a faster resync at the cost of increased memory usage during resync. The value specified in **WriteQueueLowWater** is half the number of resync block requests. After updating this registry value, execute the READREGISTRY command so that DataKeeper immediately starts using the new value.		

SnapshotLocation *

Location: *HKEY_LOCAL_MACHINE\SYSTEM\<CurrentControlSet>\Services\ExtMirr\Parameters\Volumes\{Volume GUID}\SnapshotLocation*		
Name	Type	Default Data
SnapshotLocation	REG_SZ	<drive letter>
Specifies the folder where the target snapshot file for this volume will be stored.		

TargetSnapshotBlocksize *

Location: *HKEY_LOCAL_MACHINE\SYSTEM\<CurrentControlSet>\Services\ExtMirr\Parameters\Volumes\{Volume GUID}\TargetSnapshotBlocksize*		
Name	Type	Default Data
TargetSnapshotBlocksize	REG_DWORD	None

DataKeeper target snapshot uses a default block size of 64KB for all entries that are written to the snapshot file. This block size can be modified by creating this `TargetSnapshotBlocksize` registry key.

The value should always be set to a multiple of the disk sector size, which is usually 512 bytes. Certain workloads and write patterns can benefit from changing the block size. For example, a volume that is written in a sequential stream of data (e.g. SQL Server log files) can benefit from a larger block size. A large block size results in fewer reads from the target volume when consecutive blocks are written. But a volume that is written in a random pattern may benefit from a smaller value or the default 64KB. A smaller block size will result in less snapshot file usage for random write requests.

Registry Entries that SHOULD NOT be Modified

The following registry entries are listed for informational purposes only. They should **NOT** be modified.

HKEY_LOCAL_MACHINE\SYSTEM

\CurrentControlSet

\Services

\ExtMirrSvc

This key is the base key for the service. All values directly under this key are used by the operating system to load the service. These should NOT be modified or else the service may not load correctly. The service does not use these values internally. More information on these specific keys can be obtained in the *`regentry.hlp`* file in the Windows Resource Kit.

ErrorControl

Location: *HKEY_LOCAL_MACHINE\SYSTEM\<CurrentControlSet>\Services\ExtMirrSvc\ErrorControl*		
Name	Type	Default Data
ErrorControl	REG_DWORD	1 **(Do NOT Modify)**
This value specifies what the system should do in the event the service fails to load. The default value of **1** tells the system to ignore the failure and continue booting the system. Changing this value may prevent the system from starting.		

DisplayName

Location: *HKEY_LOCAL_MACHINE\SYSTEM\<CurrentControlSet>\Services\ExtMirrSvc\DisplayName*		
Name	Type	Default Data
DisplayName	REG_SZ	SteelEye DataKeeper **(Do NOT Modify)**
This value specifies the name of the service to be displayed in the *Control Panel**Services* window.		

ImagePath

Location: *HKEY_LOCAL_MACHINE\SYSTEM\<CurrentControlSet>\Services\ExtMirrSvc\ImagePath*		
Name	Type	Default Data
ImagePath	REG_EXPAND_SZ	C:\<DK_Install_path> **(Do NOT Modify)**
This value specifies the path of the service executable.		

Start

Location: *HKEY_LOCAL_MACHINE\SYSTEM\<CurrentControlSet>\Services\ExtMirr\Parameters\Start*		
Name	Type	Default Data
Start	REG_DWORD	2 **(Do NOT Modify)**
This value specifies when the service loads. For the SteelEye DataKeeper service, this value must be set to 2, allowing the service to start automatically during system startup. Setting this value to anything else may result in a system crash or cause disk corruption.		

Type

Location: *HKEY_LOCAL_MACHINE\SYSTEM\<CurrentControlSet>\Services\ExtMirrSvc\Type*		
Name	Type	Default Data
Type	REG_DWORD	16 (0x10) **(Do NOT Modify)**

HKEY_LOCAL_MACHINE\SYSTEM

\CurrentControlSet

\Services

\ExtMirr

This key is the base key for the driver. All values directly under this key are used by the operating system to load the driver. These should not be modified or else the driver may not load correctly. The driver does not use these values internally. More information on these specific keys can be obtained in the `regentry.hlp` file in the Windows Resource Kit.

ErrorControl

Location: *HKEY_LOCAL_MACHINE\SYSTEM\<CurrentControlSet>\Services\ExtMirr\ErrorControl*		
Name	Type	Default Data
ErrorControl	REG_DWORD	1 **(Do NOT Modify)**
This value specifies what the system should do in the event the driver fails to load. The default value of 1 tells the system to ignore the failure and continue booting the system. Changing this value may prevent the system from starting.		

Group

Location: *HKEY_LOCAL_MACHINE\SYSTEM\<CurrentControlSet>\Services\ExtMirr\Group*		
Name	Type	Default Data
Group	REG_SZ	Filter **(Do NOT Modify)**
This value specifies the name of the group in which the SteelEye DataKeeper driver is a part of. This value should always be Filter. Changing this value could result in unpredictable results, including disk corruption.		

Start

Location: *HKEY_LOCAL_MACHINE\SYSTEM\<CurrentControlSet>\Services\ExtMirr\Start*		
Name	Type	Default Data
Start	REG_DWORD	0 **(Do NOT Modify)**
This value specifies when the driver loads. For the SteelEye DataKeeper driver, this value must be set to 0, allowing the driver to start during the initial phase of system boot. Setting this value to anything else may result in a system crash or cause disk corruption.		

Tag

Location: *HKEY_LOCAL_MACHINE\SYSTEM\<CurrentControlSet>\Services\ExtMirr\Tag*

Name	Type	Default Data
Tag	REG_DWORD	0x4 **(Do NOT Modify)**
This value specifies the order in which a driver loads in its group. For the SteelEye DataKeeper driver, this value should be 0x4 specifying that the driver will load at the same time as DiskPerf.Sys, which is right above FtDisk.Sys (NT's Fault Tolerant disk driver) and below the file systems. Changing this value may cause disk corruption.		

Type

Location: *HKEY_LOCAL_MACHINE\SYSTEM\<CurrentControlSet>\Services\ExtMirr\Type*		
Name	Type	Default Data
Type	REG_DWORD	0x1 **(Do NOT Modify)**
This value specifies the type of executable this key defines. For the SteelEye DataKeeper driver, this value should be 0x1 specifying that it is a kernel mode driver. Changing this value will have unpredictable results.		

HKEY_LOCAL_MACHINE\SYSTEM

\CurrentControlSet

\Services

\ExtMirr

\Parameters

The SteelEye DataKeeper driver uses this key and those below it. The values below this are used internally by the driver. The values directly under the Parameters key represent values that are global for all volumes on the system.

BuildDate

Location: *HKEY_LOCAL_MACHINE\SYSTEM\<CurrentControlSet>\Services\ExtMirr\Parameters\BuildDate*		
Name	Type	Default Data
BuildDate	REG_SZ	<None> **(Do NOT Modify)**
Specifies the date that the driver was built.		

BuildTime

Location: *HKEY_LOCAL_MACHINE\SYSTEM\<CurrentControlSet>\Services\ExtMirr\Parameters\BuildTime*		
Name	Type	Default Data
BuildTime	REG_SZ	<None> **(Do NOT Modify)**
Specifies the time that the driver was built.		

LastStartTime

Location: *HKEY_LOCAL_MACHINE\SYSTEM\<CurrentControlSet>\Services\ExtMirr\Parameters\LastStartTime*		
Name	Type	Default Data
LastStartTime	REG_DWORD	0 to MAX_DWORD **(Do NOT Modify)**
This value specifies the time, represented as seconds since January 1, 1970 in Greenwich Mean Time (GMT), since the system was started with the SteelEye DataKeeper driver running. This value is written to the registry during driver initialization and never read by the driver. This value is currently for informational purposes only.		

Version

Location: *HKEY_LOCAL_MACHINE\SYSTEM\<CurrentControlSet>\Services\ExtMirr\Parameters\Version*		
Name	Type	Default Data
Version	REG_SZ	<None> **(Do NOT Modify)**
Specifies a text string containing the version number of the last SteelEye DataKeeper driver to have booted on this system. **Note**: Any changes in the following values will take effect after the next system reboot.		

HKEY_LOCAL_MACHINE\SYSTEM

\CurrentControlSet

\Services

\ExtMirr

\Parameters

\Volumes

\{Volume GUID}

Keys under the ***Parameters\Volumes*** key represent disk volumes that have been mirrored (either Source or Target). The key name represents the GUID that Windows assigns to the volume in the Disk Management program.

BitmapFileValidOnFailover

Location: *HKEY_LOCAL_MACHINE\SYSTEM\<CurrentControlSet>\Services\ExtMirr\Parameters\Volumes\{Volume GUID}\BitmapFileValidOnFailover*		
Name	Type	Default Data
BitmapFileValidOnFailover	REG_BINARY	1 **(Do NOT Modify)**
Specifies whether a valid bitmap file was found on a failover. Used internally by the driver.		

Failover

Location: *HKEY_LOCAL_MACHINE\SYSTEM\<CurrentControlSet>\Services\ExtMirr\Parameters\Volumes\{Volume GUID}\Failover*		
Name	Type	Default Data
Failover	REG_BINARY	1 **(Do NOT Modify)**
Specifies whether the mirror is becoming a target due to failover. Used internally by the driver.		

MirrorRole

Location: *HKEY_LOCAL_MACHINE\SYSTEM\<CurrentControlSet>\Services\ExtMirr\Parameters\Volumes\{Volume GUID}\MirrorRole*		
Name	Type	Default Data
MirrorRole	REG_DWORD	0 (None), 1 (Source), 2 (Target) **(Do NOT Modify)**
Specifies the mirroring role of the volume. Used internally by the driver.		

SnapshotDevice

Location: *HKEY_LOCAL_MACHINE\SYSTEM\<CurrentControlSet>\Services\ExtMirr\Parameters\Volumes\{Volume GUID}\SnapshotDevice*		
Name	Type	Default Data

SnapshotDevice	REG_SZ	\\.\PHYSICALDRIVE<x> **(Do NOT Modify)**
Specifies the virtual disk attached for target snapshot. Used internally by the driver.		

VolumeAttributes

Location: *HKEY_LOCAL_MACHINE\SYSTEM\<CurrentControlSet>\Services\ExtMirr\Parameters\Volumes\{Volume GUID}\VolumeAttributes*		
Name	Type	Default Data
VolumeAttributes	REG_DWORD	0 **(Do NOT Modify)**
Specifies a bitmap of the volume attributes set by the SteelEye DataKeeper Service. Used internally by the service and the driver. BIT 0: All Net Alert BIT 1: Broken State Alert BIT 2: Resync Done Alert BIT 3: FailOver Alert BIT 4: Net Failure Alert BIT 5: LifeKeeper Configured BIT 6: Auto Resync Disabled		

HKEY_LOCAL_MACHINE\SYSTEM

\CurrentControlSet

\Services

\ExtMirr

\Parameters

\Volumes

\{Volume GUID}

\Targets

\{Target IP}

Note: The following fields are present under the <Target Name> subdirectory on the source and under the <Targets> subdirectory on the target.

Below is a list of registry values that define the configuration for each volume:

BitmapFileEnabled

Location: *HKEY_LOCAL_MACHINE\SYSTEM\<CurrentControlSet>\Services\ExtMirr\Parameters\Volumes\{Volume GUID}\Targets\{TargetIP}\BitmapFileEnabled*		
Name	Type	Default Data
BitmapFileEnabled	REG_BINARY	1 **(Do NOT Modify)**
Specifies whether a bitmap file will be created for a mirror. The bitmap file makes it possible for a mirror to recover from a primary system failure without a full resync.		

BitmapFileValid

Location: *HKEY_LOCAL_MACHINE\SYSTEM\<CurrentControlSet>\Services\ExtMirr\Parameters\Volumes\{Volume GUID}\Targets\{TargetIP}\BitmapFileValid*		
Name	Type	Default Data
BitmapFileValid	REG_BINARY	1 **(Do NOT Modify)**
Specifies whether the bitmap file is valid.		

Enabled

Location: *HKEY_LOCAL_MACHINE\SYSTEM\<CurrentControlSet>\Services\ExtMirr\Parameters\Volumes\{Volume GUID}\Targets\{TargetIP}\Enabled*		
Name	Type	Default Data
Enabled	REG_BINARY	1 **(Do NOT Modify)**
Indicates the mirror exists.		

TargetDriveLetter

Location: *HKEY_LOCAL_MACHINE\SYSTEM\<CurrentControlSet>\Services\ExtMirr\Parameters\Volumes\{Volume GUID}\Targets\{TargetIP}\TargetDriveLetter*		
Name	Type	Default Data
TargetDriveLetter	REG_BINARY	None **(Do NOT Modify)**

Specifies the drive letter of the volume on the target side at the time of a mirror creation or continue. This value is the Unicode representation of the drive letter.

This value is written by the driver during a mirror creation or continue operation and is present for informational purposes only. The driver does not read this value.

Note: It is possible for drive letters to change while the system is running. This can be done by using the Disk Management utility and other methods. This value is only accurate as of the last mirror create or continuation.

WARNING: THIS VALUE SHOULD NOT BE CHANGED BY THE USER.

SourceDriveLetter

Location: *HKEY_LOCAL_MACHINE\SYSTEM\<CurrentControlSet>\Services\ExtMirr\Parameters\Volumes\{Volume GUID}\Targets\{TargetIP}\SourceDriveLetter*

Name	Type	Default Data
SourceDriveLetter	REG_BINARY	None **(Do NOT Modify)**

Specifies the drive letter of the volume on the source side at the time of a mirror creation or continue. This value is the Unicode representation of the drive letter.

This value is written by the driver during a mirror creation or continue operation and is present for informational purposes only. The driver does not read this value.

Note: It is possible for drive letters to change while the system is running. This can be done by using the Disk Management program and other methods. This value is only accurate as of the last mirror create or continuation.

WARNING: THIS VALUE SHOULD NOT BE CHANGED BY THE USER.

MirrorState

Location: *HKEY_LOCAL_MACHINE\SYSTEM\<CurrentControlSet>\Services\ExtMirr\Parameters\Volumes\{Volume GUID}\Targets\{TargetIP}\MirrorState*

Name	Type	Data

MirrorState	REG_DWORD	**Range:** 0 (None), 1 (Mirror), 2 (Resync), 3 (Broken), 4 (Mirror Paused), 5 (Resync Pending) **Default:** 0 (None) **(Do NOT Modify)**
Indicates the current mirroring state of a volume. **WARNING: THIS VALUE SHOULD NOT BE CHANGED BY THE USER.**		

MirrorType

Location: *HKEY_LOCAL_MACHINE\SYSTEM\<CurrentControlSet>\Services\ExtMirr\Parameters\Volumes\{Volume GUID}\Targets\{TargetIP}\MirrorType*		
Name	Type	Data
MirrorType	REG_DWORD	**Range:** 0 (None), 1 (Synchronous), 2 (Asynchronous) **Default:** 0 (None) **(Do NOT Modify)**
Indicates the type of mirroring this volume is engaged in. **WARNING: THIS VALUE SHOULD NOT BE CHANGED BY THE USER.**		

CleanShutdown

Location: *HKEY_LOCAL_MACHINE\SYSTEM\<CurrentControlSet>\Services\ExtMirr\Parameters\Volumes\{Volume GUID}\Targets\{TargetIP}\CleanShutdown*		
Name	Type	Default Data
CleanShutdown	REG_DWORD	1 **(Do NOT Modify)**
Indicates whether reboot was intentional or the result of a failure. The driver uses this entry to determine whether to force a full resync or to do a partial resync. A graceful shutdown with no failed writes results in a partial resync. **Note**: This entry applies to source side only.		

BreakUserRequested

Location: ***HKEY_LOCAL_MACHINE\SYSTEM\<CurrentControlSet>\Services\ExtMirr\Parameters\Volumes\{Volume GUID}\Targets\{TargetIP}\BreakUserRequested***		
Name	Type	Default Data
BreakUserRequested	REG_BINARY	None **(Do NOT Modify)**
Determines whether the mirror was broken or paused because of an error or because the user requested the break/pause. If this entry indicates a break error, the system attempts to recover from the break/pause. **Note**: This entry is used internally by the SteelEye DataKeeper driver.		

RemoteName

Location: ***HKEY_LOCAL_MACHINE\SYSTEM\<CurrentControlSet>\Services\ExtMirr\Parameters\Volumes\{Volume GUID}\Targets\{TargetIP}\RemoteName***		
Name	Type	Default Data
RemoteName	REG_SZ	None **(Do NOT Modify)**
Indicates the name of the system (string value) that we are mirroring with. This value on the target indicates the source; this value on the source indicates the target.		

Chapter 4: Using EMCMD with SteelEye DataKeeper

The EMCMD utility that ships with SteelEye DataKeeper provides users with a command line method to manipulate the mirror. Because these scripts run in situations where the "normal" validation rules may not apply, EMCMD does not perform the same kinds of sanity checks that the user would experience using the SteelEye DataKeeper User Interface. EMCMD simply passes commands to the SteelEye DataKeeper Replication service allowing the service to make any decisions. It is this lack of checks that also makes this a useful diagnostic and support tool - though it is potentially dangerous for someone not very experienced with the inner workings of SteelEye DataKeeper.

The following sections detail the operation of the EMCMD SteelEye DataKeeper Command Line. You must be in the EM directory or the directory must be in your path to issue these commands.

Note: The following style conventions will be utilized throughout.

<system>	Use the system's NetBIOS name, IP address or fully qualified domain name to attach to a given system. You can also use a period (.) to attach to the local system where emcmd is being executed.
<drive>	Refers to the drive letter that is being referenced. EMCMD parses out everything after the first character, therefore, any ":" (colon) would be extraneous.

Mirror State Definitions

The following numbers are used by the system to internally describe the various states. They are used by EMCMD, and they are also the state numbers found in event log entries.

-1: Invalid State

0: No Mirror

1: Mirroring

2: Mirror is resyncing

3: Mirror is broken

4: Mirror is paused

5: Resync is pending

BREAKMIRROR

EMCMD <system> BREAKMIRROR <volume letter> [<targetsystem>]

This command forces the mirror into a **Broken** state. Breaking the mirror will cause a full resync to occur when the mirror is continued or resynced. The parameters are:

<system>	This is the source system of the mirror to break. Running the BREAKMIRROR command on the target has no effect.
<volume letter>	The drive letter of the mirror that you want to break.
<target system>	This is the IP address of the target system of the mirror to break. This optional parameter may be used if multiple targets are associated with the mirror. If not specified, the mirror will be broken to all targets.

CHANGEMIRRORENDPOINTS

EMCMD <system> CHANGEMIRRORENDPOINTS <volume letter> <old target IP> <new source IP> <new target IP>

This command is used to move a DataKeeper protected volume to another network location.

Note: This command supports changing the endpoints of a mirrored volume that is configured on 3 nodes or fewer. If your configuration consists of more than three nodes, the mirrors must be deleted and recreated.

Refer to the examples below.

See "WAN Considerations" and "Initial Synchronization of Data Across the LAN/WAN" in the "Configuration" section.

<system>	This is the system that has the new source IP address available for the mirror.
<volume letter>	The drive letter of the mirror to be changed.
<old target IP>	The previous IP address of the target system.
<new source IP>	The new IP address of the source system.
<new target IP>	The new IP address of the target system.

Notes:

- A job may contain multiple volumes and multiple mirrors. The CHANGEMIRRORENDPOINTS command will modify endpoints on one mirror each time it is used. For a 1x1 mirror (1 source, 1 target), only one command is required. For a 2x1 mirror (2 nodes with a shared volume with one target node) or a 1x1x1 (1 source, two target nodes), two commands are required to change the necessary mirror endpoints.
- If an existing mirror whose endpoints are being changed is currently an active mirror, it must be put into the **Paused**, **Broken** or **Resync Pending** state before the endpoints can be changed.

	**CAUTION**: Using the Break command will cause a **full resync**. It is recommended that the mirror be Paused instead.

- Before making changes, it will be helpful to display **Job Information** for the volume. For example, `emcmd . getJobInfoForVol D .`
- While making endpoint changes, the **Job** icon in the DataKeeper GUI may turn red. However, it will return to green after the `ContinueMirror` command is performed.

In the following examples, we move mirrors from the 172.17.103 subnet to the 192.168.1 subnet. The basic steps are as follows:

1. **Display job information** for the volume
2. **Pause the Mirror** using the EMCMD command line
3. **Change the IP address** on the system(s) (if necessary)

IMPORTANT: If you haven't already done so, prior to performing the **CHANGEMIRRORENDPOINTS** command, **update the IP addresses** for the source and target. This will automatically place the mirror into the **Paused** state.

4. **Run EMCMD CHANGEMIRRORENDPOINTS** to change to the new IP address

1x1 Mirror CHANGEMIRRORENDPOINTS Command Example

For a 1x1 mirror (source and target only), one command is required.

```
emcmd SYS1.MYDOM.LOCAL getJobInfoForVol D

        ID = caa97f9f-ac6a-4b56-8f25-20db9e2808a8

        Name = Mirr Vol D

        Description = Mirror Volume D

        MirrorEndPoints =
        SYS3.MYDOM.LOCAL;D;172.17.103.223;SYS1.MYDOM.LOCAL;E
        ;172.17.103.221;A

emcmd SYS1.MYDOM.LOCAL PauseMirror D

emcmd SYS1.MYDOM.LOCAL ChangeMirrorEndPoints D 172.17.103.223
192.168.1.221 192.168.1.223

emcmd SYS1.MYDOM.LOCAL getJobInfoForVol D

. . .

        MirrorEndPoints =
        SYS3.MYDOM.LOCAL;D;192.168.1.223;SYS1.MYDOM.LOCAL;D;
        192.168.1.221;A

emcmd SYS1.MYDOM.LOCAL ContinueMirror D
```

2x1 Mirror CHANGEMIRRORENDPOINTS Command Example

For a 2x1 mirror that includes a shared source volume and a target volume, two commands are required.

```
emcmd SYS1.MYDOM.LOCAL getJobInfoForVol E

        ID = caa97f93e-ac6a-4b56-8f25-20db9e2808a8

        Name = Mirr Vol E

        Description = Mirror Volume E

        MirrorEndPoints =
        SYS1.MYDOM.LOCAL;E;0.0.0.0;SYS2.MYDOM.LOCAL;E
        ;0.0.0.0;D

        MirrorEndPoints =
        SYS3.MYDOM.LOCAL;E;172.17.103.223;SYS2.MYDOM.LOCAL;E
        ;172.17.103.222;A

        MirrorEndPoints =
        SYS3.MYDOM.LOCAL;E;172.17.103.223;SYS1.MYDOM.LOCAL;E
        ;172.17.103.221;A

emcmd SYS1.MYDOM.LOCAL PauseMirror E

emcmd SYS1.MYDOM.LOCAL ChangeMirrorEndPoints E 172.17.103.223
192.168.1.221 192.168.1.223

emcmd SYS2.MYDOM.LOCAL ChangeMirrorEndPoints E 172.17.103.223
192.168.1.222 192.168.1.223

emcmd SYS1.MYDOM.LOCAL getJobInfoForVol E

. . .

        MirrorEndPoints =
        SYS1.MYDOM.LOCAL;E;0.0.0.0;SYS2.MYDOM.LOCAL;E;0.0.0.
        0;D

        MirrorEndPoints =
        SYS3.MYDOM.LOCAL;E;192.168.1.223;SYS2.MYDOM.LOCAL;E;
        192.168.1.222;A

        MirrorEndPoints =
        SYS3.MYDOM.LOCAL;E;192.168.1.223;SYS1.MYDOM.LOCAL;E;
        192.168.1.221;A

emcmd SYS1.MYDOM.LOCAL ContinueMirror E
```

1x1x1 Mirror CHANGEMIRRORENDPOINTS Command Example

For a 1x1x1 mirror that includes 2 Target volumes, 2 commands are required.

```
emcmd SYS1.MYDOM.LOCAL getJobInfoForVol J

        ID = caa97f93j-ac6a-4b56-8f25-20db9j2808a8

        Name = Mirr Vol J

        Description = Mirror Volume J

        MirrorEndPoints =
        SYS1.MYDOM.LOCAL;J;172.17.103.221;SYS3.MYDOM.LOCAL;J
        ;172.17.103.223;A

        MirrorEndPoints =
        SYS3.MYDOM.LOCAL;J;172.17.103.223;SYS2.MYDOM.LOCAL;J
        ;172.17.103.222;A

        MirrorEndPoints =
        SYS1.MYDOM.LOCAL;J;172.17.103.221;SYS2.MYDOM.LOCAL;J
        ;172.17.103.222;A
```

In this example the system "SYS3.MYDOM.LOCAL" will be moved to another site.

SYS1 and SYS2 will now use a new subnet (192.168.1.*) to communicate with SYS3.

However, SYS1 and SYS2 will continue using 172.17.103.* to communicate with each other.

```
emcmd SYS1.MYDOM.LOCAL PauseMirror J

emcmd SYS1.MYDOM.LOCAL ChangeMirrorEndPoints J 172.17.103.223
192.168.1.221 192.168.1.223

emcmd SYS2.MYDOM.LOCAL ChangeMirrorEndPoints J 172.17.103.223
192.168.1.222 192.168.1.223

emcmd SYS1.MYDOM.LOCAL getJobInfoForVol J

. . .

        MirrorEndPoints =
        SYS1.MYDOM.LOCAL;J;192.168.1.221;SYS3.MYDOM.LOCAL;J;
        192.168.1.223;A

        MirrorEndPoints =
        SYS3.MYDOM.LOCAL;J;192.168.1.223;SYS2.MYDOM.LOCAL;J;
        192.168.1.222;A

        MirrorEndPoints =
```

```
        SYS1.MYDOM.LOCAL;J;172.17.103.221;SYS2.MYDOM.LOCAL;J
        ;172.17.103.222;A

emcmd SYS1.MYDOM.LOCAL ContinueMirror J
```

CLEARASR_OK

EMCMD <system> CLEARASR_OK <volume> [<target_ip>]

This command clears the Automatic Split-brain Recovery OK flag, setting it to **FALSE**. If a target_ip is specified, the flag for that mirror is set to **FALSE**. If no target_ip is specified, the flags for all mirrors of the specified volume are set to **FALSE**. This command is only valid on a mirror **Source** system.

The Automatic Split-brain Recovery OK flag is for internal DataKeeper use only - it is not expected that a user would need to execute this command.

The parameters are:

<system>	**Source** system only
<volume letter>	Drive letter of the mirrored volume
[<target ip>]	IP address of the **Target** system

CLEARSNAPSHOTLOCATION

EMCMD <system> CLEARSNAPSHOTLOCATION <volume letter>

This command clears the snapshot location (directory path) for the given volume on the given system. Once this command executes successfully, snapshots will be disabled for the given volume.

The parameters are:

<system>	This is the system name/IP address of snapshot location.
<volume letter>	This is the drive letter of the volume to be snapshotted.

Sample output:

```
Status = 0
```

When the command is successful, it will return a status of 0. Otherwise, it will report a non-zero status.

CLEARSWITCHOVER

EMCMD <system> CLEARSWITCHOVER <volume letter>

This command should be run on a target system where a mirror has been previously deleted with the DELETELOCALMIRRORONLY command and now needs to be re-established. This command clears the SteelEye DataKeeper switchover flag that is set for a volume that has been deleted from the Target role using DELETELOCALMIRRORONLY. If you delete a target using DELETELOCALMIRRORONLY and do not run CLEARSWITCHOVER, you will not be able to re-establish a mirror target unless you reboot the system.

<system>	This is the target system where you just ran DELETELOCALMIRRORONLY
<volume letter>	The drive letter of the mirror.

CONTINUEMIRROR

EMCMD <system> CONTINUEMIRROR <volume letter> [<target system>]

This command forces a paused or broken mirror to resume mirroring. On successful completion of the resync (full or partial), the mirror state is changed to **Mirroring**. This command will not automatically relock the target volume if it is unlocked.

Note: If target volume is unlocked, it must be relocked prior to running this command.

The parameters are:

<system>	This is the source system of the mirror to resume mirroring.
<volume letter>	The drive letter of the mirror that you want to resume mirroring.
<target system>	This is the IP address of the target system of the mirror to resync. This optional parameter may be used if multiple targets are associated with the mirror. If not specified, a resync will be performed to all targets.

CREATEJOB

EMCMD <system> CREATEJOB <JobName> <Description><SysName1> <DrvLetter1> <IP1> <SysName2><DrvLetter2> <IP2> <MirrorType> ...

This command is for internal use only.

CREATEMIRROR

EMCMD <system> CREATEMIRROR <volume letter> <target system> <type> [options]

This command creates a mirror between two machines, using the same drive letter on each. The parameters are as follows:

< system >	This is the IP address of the source system (see Note below).
< volume letter>	This is the drive letter that is being mirrored. This will be both the source and target drive letter.
<target system >	This is the IP address of the target system (see Note below).
<type>	This is the type of mirror, where type is a single character: A - Create an Asynchronous Mirror S - Create a Synchronous Mirror
[options]	Optional arguments that specify behavior deviant from the norm. These can be OR'd together to create a set of options (*add decimal values - for example, for option 1 + option 4, place a 5 in the command*). They are: 1: Create the mirror without doing a full resync operation. 2: Do not wait for the target side of the mirror to be created before returning. 4: Create with boot-time restrictions in place - essentially treat the create as you would a mirror re-establishment as part of the boot process. This option will check to see if the remote system is already a source and fail the creation if it determines that it was a source.

	**NOTE**: Both source and target IP addresses must be of the same protocol. A mirror can only be created using two IPV4 or two IPV6 addresses. DataKeeper does not currently support mirror endpoints with different protocols. IPV6 configuration is NOT supported in a Windows 2003 environment.

IPv4 Example:

```
EMCMD 192.168.1.1 CREATEMIRROR E 192.168.1.2 A 5
```

IPv6 Example:

```
EMCMD 2001:5c0:110e:3304:a6ba:dbff:feb2:f7fd CREATEMIRROR F
2001:5c0:110e:3304:a6ba:dbff:feb2:afd7 A 5
```

DELETEJOB

EMCMD <system> DELETEJOB [<JobId>]

This command is for internal use only.

DELETELOCALMIRRORONLY

EMCMD <system> DELETELOCALMIRRORONLY <volume letter> [<target system>]

This command deletes the mirror only on the <system> it is issued on. It handles the case when a mirror ends up with a target and no source or source and no target. The parameters are:

<system>	This can be either the source or the target system.
<volume letter>	The drive letter of the mirror that you want to delete.
<target system>	This is the IP address of the target system of the mirror to delete. This optional parameter may be used if multiple targets are associated with the mirror.

DELETEMIRROR

EMCMD <system> DELETEMIRROR <volume letter> [<target system>]

This command deletes the mirror from both the source and the target if <system> is a source. If <system> is a target, it will delete the target side of the mirror only if the source system is down. The parameters are:

<system>	This can be either the source or the target system.
<volume letter>	The drive letter of the mirror that you want to delete.
<target system>	This is the IP address of the target system of the mirror to delete. This optional parameter may be used if multiple targets are associated with the mirror. If not specified, the mirror will be deleted for all targets.

DROPSNAPSHOT

EMCMD <system> DROPSNAPSHOT <volume letter> [<volume letter> ...]

This command will notify DataKeeper to lock the volume and clean up the snapshot files that it created. The parameters are:

<system>	This is the IP address of the system containing the snapshot.
<volume letter>	This is the drive letter of the snapshotted volume on the target server. If dropping multiple snapshots, the drive letters should be separated by spaces.

GETASR_OK

EMCMD <system> GETASR_OK <volume letter> <target_ip>

This command retrieves the Automatic Split-brain Recovery OK flag for the given mirror. This command is only valid on a mirror **Source** system.

The Automatic Split-brain Recovery OK flag is for internal DataKeeper use only - it is not expected that a user would need to execute this command.

The parameters are:

<system>	**Source** system only
<volume letter>	Drive letter of the mirrored volume
<target ip>	IP address of the **Target** system

Sample output:

```
ASR_OK: FALSE
```

When the command is successful, it will report FALSE or TRUE.

GETCOMPLETEVOLUMELIST

EMCMD <system> GETCOMPLETEVOLUMELIST

This command displays information on all volumes eligible to be mirrored or already in a mirror. Sample output:

Volume 1 information:

Volume Root	= F:
Volume Label	= New Volume
Volume File System	= NTFS
Volume Total Space	= 2151608320
Mirror Role	= 01
Number of targets	= 2

Target 0 information:	
Volume State	= 0001
Target System	= 10.1.1.133
Target Drive Letter	= F
Target 1 information:	
Volume State	= 0002
Target System	= 10.1.1.134
Target Drive Letter	= F

GETCONFIGURATION

EMCMD <system> GETCONFIGURATION <volume letter>

This command retrieves and displays the net alert settings (also referred to as "volume attributes") for the volume. The parameters are:

<system>	This can be either the source or the target systems.
<volume letter>	The drive letter of the volume you want information on.

Sample output:

** Calling GetConfiguration [Volume F] **

All Net Alert bit	IS NOT enabled
Net Alert	IS NOT enabled
Broken State Alert	IS NOT enabled
Resync Done Alert	IS NOT enabled
Failover Alert	IS NOT enabled
Net Failure Alert	IS NOT enabled
LK Config	IS NOT enabled
Auto Resync	IS NOT enabled
MS Failover Cluster Config	IS NOT enabled
Shared Volume	IS NOT enabled

GETEXTENDEDVOLUMEINFO

EMCMD <system> GETEXTENDEDVOLUMEINFO <volume letter>

This command returns extended volume information about the selected volume such as disk signature, physical disk offset and internal disk id. The parameters are:

<system>	This can be either the source or the target systems.
<volume letter>	The drive letter of the volume you want information on.

Sample output:

----------EXTENDED INFO ---

Physical Disk Signature = {217abb5a-0000-0000-0000-000000000000}

Physical Disk Offset = 32256

Internal Disk ID = 0xf2fa

GETJOBINFO

EMCMD <system> GETJOBINFO [<JobId>]

This command displays job information for a specific JobId or all defined jobs.

GETJOBINFOFORVOL

EMCMD <system> GETJOBINFOFORVOL <DrvLetter> [<FullSysname>|<IP>]

This command displays job information related to a specific volume on a specific system.

GETMIRRORTYPE

EMCMD <system> GETMIRRORTYPE <volume letter>

This command provides a numeric output of the type of mirror.

The parameters are:

<system>	This can be either the source or the target systems.
<volume letter>	The driver letter of the volume you want information on.

Output format:

c:> EMCMD . GETMIRRORTYPE F

Target system 10.1.1.133, Type 2

Target system 10.1.1.134, Type 2

Mirror Type:

-1: Invalid Type (EMCMD cannot get the requested information.)

0: No mirror

1: Synchronous Mirror

2: Asynchronous Mirror

GETMIRRORVOLINFO

EMCMD <system> GETMIRRORVOLINFO <volume letter>

This command provides a very terse output of the state of mirror. The command GETMIRRORVOLINFO can return multiple lines of output (one per target). It provides essentially the same information as the GETVOLUMEINFO command does.

The parameters are:

<system>	This can be either the source or the target systems.
<volume letter>	The drive letter of the volume you want information on.

Sample output:

c:> EMCMD . GETMIRRORVOLINFO F

F: 1 CARDINAL10.1.1.133 1

F: 1 CARDINAL10.1.1.134 1

Output format:

[Volume Letter} {Mirror Role} [Source System] [Target System] [Mirror State]

Mirror Role: 1 = source; 2 = target

Mirror State:

-1: Invalid State

0: No mirror

1: Mirroring

2: Mirror is resyncing

3: Mirror is broken

4: Mirror is paused

5: Resync is pending

GETREMOTEBITMAP

EMCMD <system> GETREMOTEBITMAP <volume letter> <targetsystem> <local file>

This command is for internal use only.

GETRESYNCSTATUS

EMCMD <system> GETRESYNCSTATUS <volume letter>

This command returns information indicating the overall status of a resync operation. The parameters are:

<system>	This can be either the source or the target systems.
<volume letter>	The drive letter of the volume you want to set the configuration on.

Sample output:

Resync Status for Volume F:

Target 0 (Target System 10.1.1.133)

ResyncPhase : 3

BitmapPass : 1

NumberOfBlocks : 32831

DirtyBlocks : 0

CurrentBlock : 0

NewWrites : 1803

ResyncStartTime: Fri Nov 05 13.57.51 2008

LastResyncTime : Fri Nov 05 13.57.51 2008

Target 1 (Target System 10.1.1.134)

ResyncPhase : 2

BitmapPass : 0

NumberofBlocks : 32831

DirtyBlocks : 2124

CurrentBlock : 29556

NewWrites : 0

ResyncStartTime: Fri Nov 05 15:09:47 2008

LastResyncTime: Fri Nov 05 15:09:47 2008

The **ResyncPhase** is used internally and has little meaning outside of the development environment. The values are: 0 (unknown), 1 (initial), 2 (update), and 3 (done).

The **BitmapPass** is the number of times we have passed through the bitmap indicating the number of dirty blocks. We count from zero. If we do a resync in one pass, then this never increments.

The **NumberOfBlocks** is the number of 64K data blocks on the volume.

The **DirtyBlocks** parameter is the number of blocks that the bitmap indicates need to be updated (and have not already been).

The **CurrentBlock** parameter indicates the current location in the bitmap.

The **NewWrites** parameter indicates the number of writes that have occurred on the volume since we have been resyncing.

The **ResyncStartTime** and **LastResyncTime** parameters describe the time that the resync was begun and the last time a resync write operation was sent across the network.

GETSERVICEINFO

EMCMD <system> GETSERVICEINFO

This command retrieves version and other information about the SteelEye DataKeeper service and driver that is running on the specified machine. The parameters are:

<system>	This can be either the source or the target systems.

Sample output:

Service Description: = SteelEye DataKeeper Service

Service Build Type: = Release

Service Version = 7.0

Service Build = 1

Driver Version = 7.0

Driver Build = 1

Volume Bit Map = 1000070h

Service Start Time = Fri Oct 06 11:20:45 2008

Last Modified Time = Fri Oct 06 15:11:53 2008

GETSNAPSHOTLOCATION

EMCMD <system> GETSNAPSHOTLOCATION <volume letter>

This command retrieves the currently configured snapshot location (directory path) for the given volume on the given system. It will return an empty result if the snapshot location is not configured on the given volume.

The parameters are:

<system>	This is the system name/IP address of the system containing volume to be snapshotted.
<volume letter>	This is the drive letter of the volume to be snapshotted.

Sample output:

```
C:\Temp
```

When the command is successful, it will report the snapshot directory path on stdout, which will be empty if snapshot location is not yet configured.

GETSOURCEMIRROREDVOLUMES

EMCMD <system> GETSOURCEMIRROREDVOLUMES

This command displays information about the volumes on the system that are currently the source in a mirror.

Sample output:

Status = 0

Source Volume = F:

Source Label = New Volume

Source #Targs = 2

Target 0

Target System = 10.1.1.133

Mirror State = 0001

Target 1

Target System = 10.1.1.134

Mirror State = 0001

GETTARGETMIRROREDVOLUMES

EMCMD <system> GETTARGETMIRROREDVOLUMES

This command displays information about the volumes on the system that are currently the target in a mirror.

Sample output:

** Calling GetTargetMirroredVolumes **

Returned 1 Target Volumes

Target Volume 1 information:

Volume Root = F:

Volume State = 1

Source = 10.1.1.132

Target = BLUEJAY

GETVOLUMEDRVSTATE

EMCMD <system> GETVOLUMEDRVSTATE <volume letter>

This command retrieves the current state of the SteelEye DataKeeper device driver.

The parameters are:

<system>	This can be either the source or the target systems.
<volume letter>	The drive letter of the volume you want to get the configuration on.

The output is a number indicating the state. The output is purposely terse as it is designed to be parsed in a DataKeeper recovery script. The output is one of the following mirror states:

-1: Invalid State

0: No mirror

1: Mirroring

2: Mirror is resyncing

3: Mirror is broken

4: Mirror is paused

5: Resync is pending

The output also provides the address of the mirror end point (source or target).

GETVOLUMEINFO

EMCMD <system> GETVOLUMEINFO <volume letter> <level>

This command returns information about the selected volume. The parameters are:

<system>	This can be either the source or the target systems.
<volume letter>	The drive letter of the volume you want information on.
<level>	A number between 1-3 indicating the amount of detail you want.

Sample output:

----------LEVEL 1 INFO----------

Volume Root = F:

Last Modified = Fri Nov 05 15:24:14 2008

Mirror Role = SOURCE

Label = New Volume

FileSystem = NTFS

Total Space = 2151608320

Num Targets = 2

Attributes : 20h

---------- -------LEVEL 2 INFO----------

>> Remote [0] = 10.1.1.133, F:

Mirror State = MIRROR

Mirror Type = ASYNCHRONOUSLY

>> Remote [1] = 10.1.1.133, F:

Mirror State = MIRROR

Mirror Type = ASYNCHRONOUSLY

----------LEVEL 3 INFO----------

>> Remote [0} = 10.1.1.133, F:

No Resync or CompVol Statistics to report

>> Remote [1] = 10.1.1.134, F:

No Resync or CompVol Statistics to report

ISBREAKUSERREQUESTED

EMCMD <system> ISBREAKUSERREQUESTED <volume letter>

This command checks whether a broken mirror is a result of a user request. The command may only be run on the local system. The parameters are:

<system>	This should be the local system.
<volume letter>	The drive letter of the volume that you want to check.

Output:

TRUE	The mirror was broken because of a user request.
FALSE	The mirror was broken by SteelEye DataKeeper (e.g., network failure, failure to write data on target side, etc). The volume is not in a BROKEN (3) state.

ISPOTENTIALMIRRORVOL

EMCMD <system> ISPOTENTIALMIRRORVOL <volume letter>

This command checks to determine if a volume is a candidate for mirroring. The command may only be run on the local system. The parameters are:

<system>	This should be the local system.
<volume letter>	The drive letter of the volume that you want to check.

Output:

TRUE - The volume is available for mirroring.

Otherwise, the output may be some combination of the following:

System Drive

RAW filesystem

FAT filesystem

ACTIVE partition

Contains PageFile

GetDriveType not DRIVE_FIXED

If the drive letter points to a newly created volume (i.e. SteelEye DataKeeper driver not attached yet), or a non-disk (network share, CD-ROM), the output will be:

Unable to open - SteelEye DataKeeper driver might not be attached (you may need to reboot) or this might not be a valid hard disk volume.

If there is an internal error getting volume information, you may see the message:

```
Error in GetVolumeInfo - <Hex error code>
```

You should never see this, but it is possible. It indicates a problem in the service or driver.

LOCKVOLUME

EMCMD <system> LOCKVOLUME <volume letter>

This command forces an exclusive lock on the volume specified. This call will fail if a process owns open handles into the volume. The parameters are:

<system>	This can be either the source or the target systems.
<volume letter>	The drive letter of the volume you want to lock.

MERGETARGETBITMAP

EMCMD <system> MERGETARGETBITMAP <volume letter> <target system>

This command is for internal use only.

PAUSEMIRROR

EMCMD <system> PAUSEMIRROR <volume letter> [<target system>]

This command forces the mirror into a **Paused** state. The parameters are:

<system>	This is the source system of the mirror to pause. Running the PAUSEMIRROR command on the target has no effect.

<volume letter>	The drive letter of the mirror that you want to pause.
<target system>	This is the IP address of the target system of the mirror to pause. This optional parameter may be used if multiple targets are associated with the mirror. If not specified, the mirror to all targets will be paused.

PREPARETOBECOMETARGET

EMCMD <system> PREPARETOBECOMETARGET <volume letter>

This command should only be used to recover from a Split-Brain condition. It should be run on the system where the mirror is to become a target and is only valid on a mirror source. This command causes the mirror to be deleted and the volume to be locked.

To complete split-brain recovery, run CONTINUEMIRROR on the system that remains as the mirror source.

Example Scenario

If volume F: is a mirror source on both SYSA and SYSB, you can use emcmd to resolve this split-brain situation. Choose one of the systems to remain a source - for example, SYSA. Make sure there are no files or modifications on SYSB that you want to save - if so, these need to be copied manually to SYSA. To re-establish the mirror, perform the following steps:

```
EMCMD SYSB PREPARETOBECOMETARGET F
```

The mirror of F: on SYSB will be deleted and the F: drive will be locked.

```
EMCMD SYSA CONTINUEMIRROR F
```

Mirroring of the F: drive from SYSA to SYSB will be established, a partial resync will occur (overwriting any changes that had been made on SYSB), and the mirror will reach the **Mirroring** state.

READREGISTRY

EMCMD <system>READREGISTRY <volume letter>

This command tells the SteelEye DataKeeper driver to re-read its registry settings. The parameters are:

<system>	This can be either the source system or the target system.
<volume letter>	The drive letter of the mirror for which you want to re-read settings.

This command causes the following registry settings to be re-read and any changes to take effect.

Source system (changes to these parameters take effect immediately):

BandwidthThrottle

CompressionLevel

WriteQueueHighWater

WriteQueueLowWater

DontFlushAsyncQueue

Target system (changes take effect the next time the source and target systems reconnect):

TargetPortBase

TargetPortIncr

RESTARTVOLUMEPIPE

EMCMD <system> RESTARTVOLUMEPIPE <volume letter>

This command is for internal use only.

RESYNCMIRROR

EMCMD <system> RESYNCMIRROR <volume letter> [<target system>]

This command forces the mirror to be fully resynced. The parameters are:

<system>	This is the source system name.
<volume letter>	This is the drive letter of the mirror that should be resynced.
<target system>	This is the IP address of the target system of the mirror to resync. This optional parameter may be used if multiple targets are associated with the mirror. If not specified, a resync to all targets will be performed.

SETASR_OK

EMCMD <system> SETASR_OK <volume> [<target_ip>]

This command sets the Automatic Split-brain Recovery OK flag to **TRUE**.

If a target_ip is specified, the flag is set to TRUE for that mirror. If no target_ip is specified, the flags for all mirrors of the specified volume that are in the **Mirroring** state - or in the **Paused** state, with **BreakUserRequested** = **FALSE** - are set to **TRUE**. This command is only valid on a mirror **Source** system.

The Automatic Split-brain Recovery OK flag is for internal DataKeeper use only - it is not expected that a user would need to execute this command.

The parameters are:

<system>	**Source** system only
<volume letter>	Drive letter of the mirrored volume
[<target ip>]	IP address of the **Target** system

SETCONFIGURATION

EMCMD <system> SETCONFIGURATION <volume letter><configuration mask>

This command sets the net alert settings (also referred to as "volume attributes") for the volume. The parameters are:

<system>	This can be either the source or the target system.
<volume letter>	The drive letter of the volume you want to set the configuration on.
<configuration mask>	This is a bitmask indicating the net alert settings. These bits are defined: 1 -- 0x01: All Net Alerts is enabled 2 -- 0x02: Broken State Alert is enabled 4 -- 0x04: Resync Done Alert is enabled 8 -- 0x08: Failover Alert is enabled 16 -- 0x10: Net Failure Alert is enabled 32 -- 0x20: LifeKeeper Config is enabled 64 -- 0x40: Auto Resync is enabled 128 -- 0x80: MS Failover Cluster Config is enabled 256 -- 0x100: Shared Volume is enabled

Example to clear all flags:

EMCMD . SETCONFIGURATION E 0

Multiple Configuration Example to enable Shared Volume and MS Failover Cluster Config (*add decimal values 256 + 128)***:**

EMCMD . SETCONFIGURATION E 384

SETSNAPSHOTLOCATION

EMCMD <system> SETSNAPSHOTLOCATION <volume letter> "<directory path>"

This command sets the snapshot location (directory path) for the given volume on the given system. The directory must be valid on the system in question, must be a local drive/path, must be an absolute path and cannot be left blank (see CLEARSNAPSHOTLOCATION). If no snapshot location is currently configured, executing this command will have the effect of enabling target snapshots on the given volume.

The parameters are:

<system>	This is the system name/IP address containing volume to be snapshotted.
<volume letter>	This is the drive letter of the volume to be snapshotted.
<directory path>	This is the absolute directory path, local to <system>, for the snapshot file location. Note that this value must be enclosed in quotes if the path contains a space character.

Sample output:

```
Status = 0
```

When the command is successful, it will return a status of 0. Otherwise, it will report a non-zero status.

STOPSERVICE

EMCMD <system> STOPSERVICE

This command stops the DataKeeper service.

SWITCHOVERVOLUME

EMCMD <system> SWITCHOVERVOLUME <volume letter> [-f]

This command attempts to make the given system become the source for the requested volume. **This command is for internal use only.**

< system >	This is the IP address of the system to become source.
< volume letter>	This is the drive letter of the requested volume.
[-f]	This option may be used for a ***fast*** (*unsafe*) switchover. This option should only be used if the status of the current source is known. Incorrect usage of this can result in a split-brain condition.

TAKESNAPSHOT

EMCMD <target system> TAKESNAPSHOT <volume letter> [<volume

letter> ...]

This command, run on the target system, will notify DataKeeper to establish a snapshot of the given volume (s) on the given system. If no snapshot location has been configured, the command will fail.

The parameters are:

<target system>	This is the target system name/IP address containing the volume to be snapshotted.
<volume letter>	This is the drive letter(s) of the volume(s) to be snapshotted on the target server. If multiple volumes are to be snapshotted, the drive letters should be separated by spaces.

Note: All target volumes must have the same source system.

UNLOCKVOLUME

EMCMD <system> UNLOCKVOLUME <volume letter>

This command forces the volume specified to unlock. The parameters are:

<system>	This can be either the source or the target systems.
<volume letter>	The drive letter of the volume you want to unlock.

UPDATEJOB

EMCMD <system> UPDATEJOB <JobId> <Name> <Descr> [<SysName1> <DrvLetter1> <IP1> <SysName2><DrvLetter2> <IP2> <MirrorType>]...

This command is for internal use only.

UPDATEVOLUMEINFO

EMCMD <system> UPDATEVOLUMEINFO <volume letter>

This command causes the SteelEye DataKeeper service to query the driver for the correct mirror state. This command is useful if the DataKeeper GUI displays information that appears to be incorrect or not up-to-date.

Note: The SteelEye DataKeeper service updates the volume information automatically based on new messages in the system Event Log.

The parameters are:

<system>	This can be either the source or the target system.
<volume letter>	The drive letter of the volume that you want to update its info.

Chapter 5: User Guide

The topics in this section are designed to be a reference for you as you get started using SteelEye DataKeeper, helping you identify the type of configuration you are interested in implementing and providing detailed instructions for effectively using your SteelEye DataKeeper software.

Getting Started

Choose Your Configuration

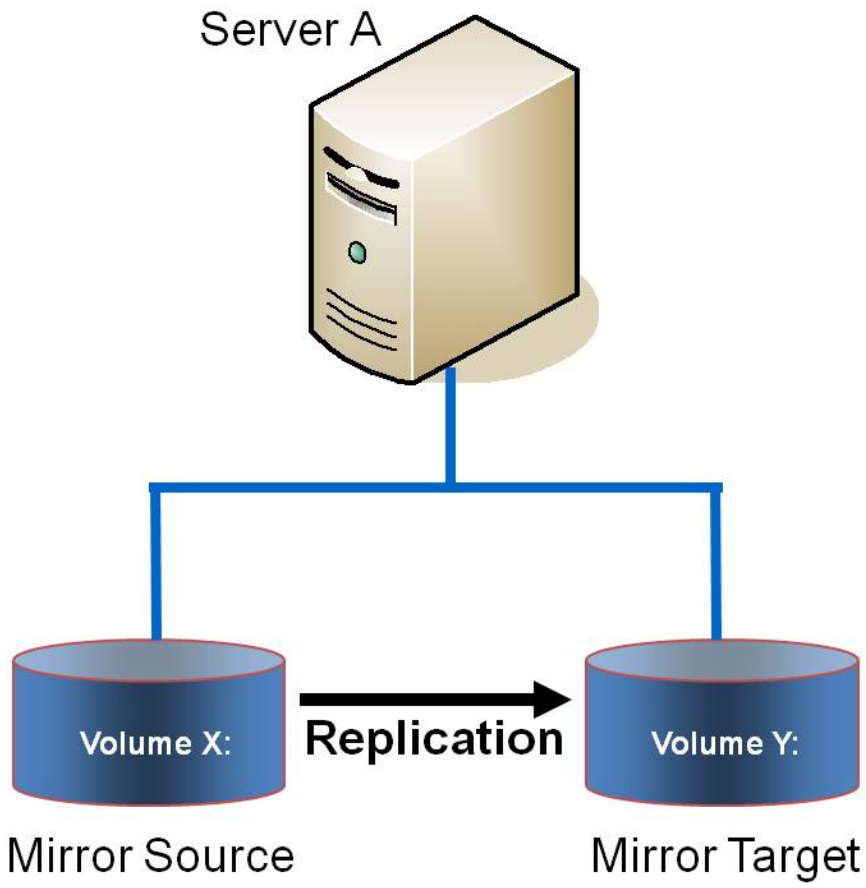
DataKeeper can be utilized in a number of different configurations to facilitate a number of different functions including:

- Provide a second physical copy of your data
- Eliminate the Single Point of Failure associated with traditional MSCS/WSFC clusters

Review the following replication configurations and their example USE CASES to familiarize yourself with just some of DataKeeper's capabilities. Then use the topics associated with the configuration you are interested in to obtain detailed information about that configuration.

Disk-to-Disk

This is a simple one server, two disks configuration, mirroring Volume X on Server A to Volume Y on Server A.



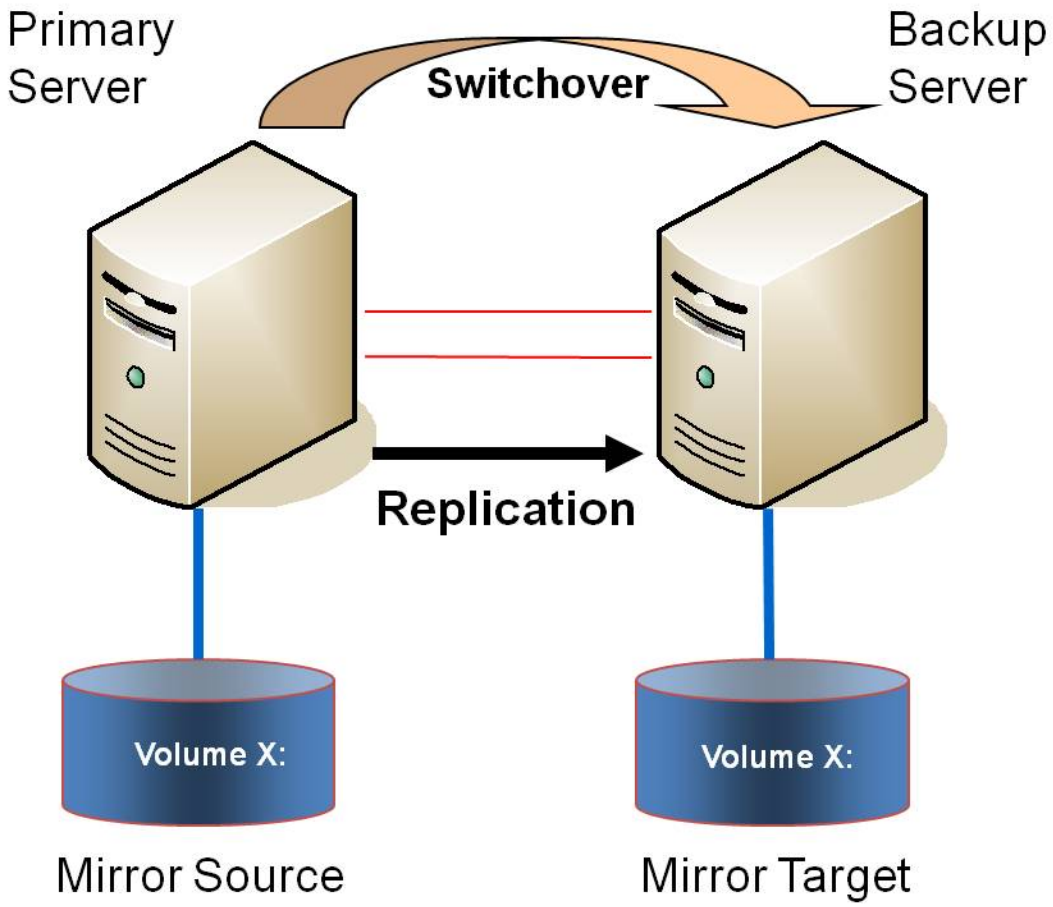
Example: USE CASE	Replicate data from one volume on a server to another volume on the same server. These disks can be different storage arrays, protecting against data loss should the primary SAN fail.

Additional topics of interest include:

- Creating Mirrors
- Managing Mirrors
- Extensive Write Considerations
- Frequently Asked Questions

One-to-One

This is a simple one source, one target configuration, mirroring Volume X across the network. In addition to providing a second physical copy of the data, DataKeeper also provides the ability to switch over the mirror which allows the data to become active on the backup server.



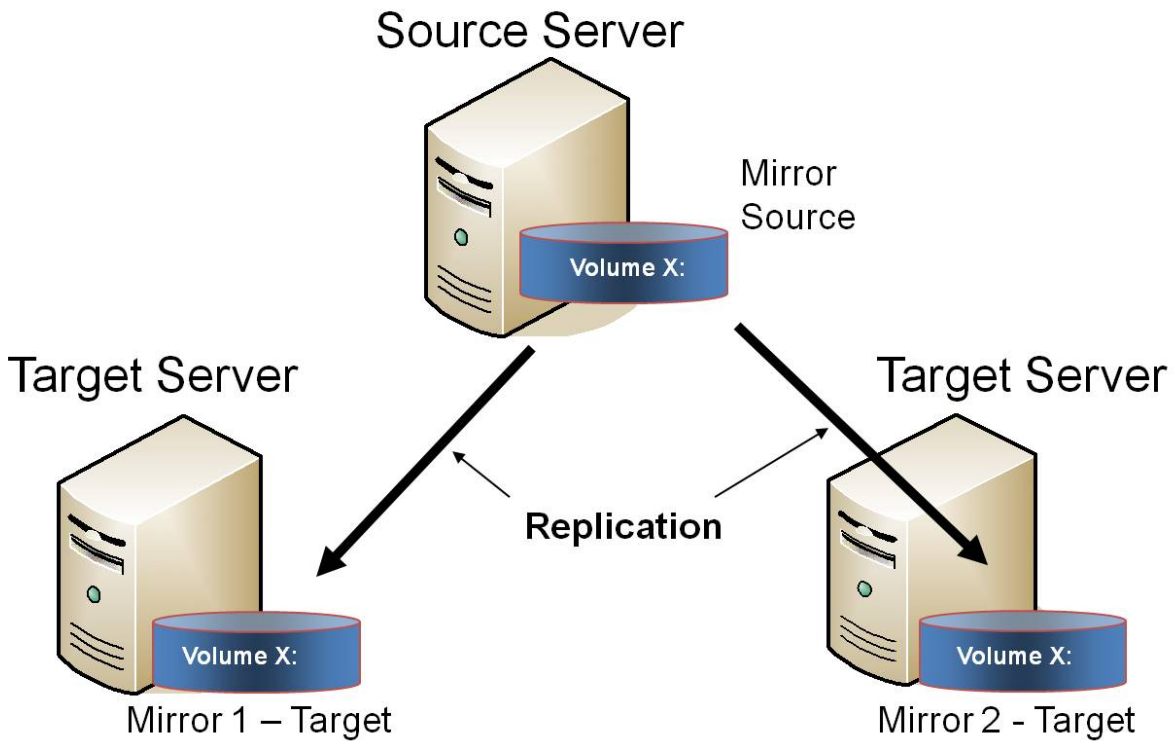
Example: USE CASE	Replicate data on one or more volumes from a server in one city to another server in another city.

Additional topics of interest include:

- Primary Server Shutdown
- Secondary Server Failures
- Using DataKeeper Standard Edition To Provide Disaster Recovery For Hyper-V Virtual Machines
- Frequently Asked Questions

One-to-Many (Multiple Targets)

This configuration involves one primary (source) system replicating one (or more) volume(s) to two different target systems across the network. This is referred to as a multiple target configuration.



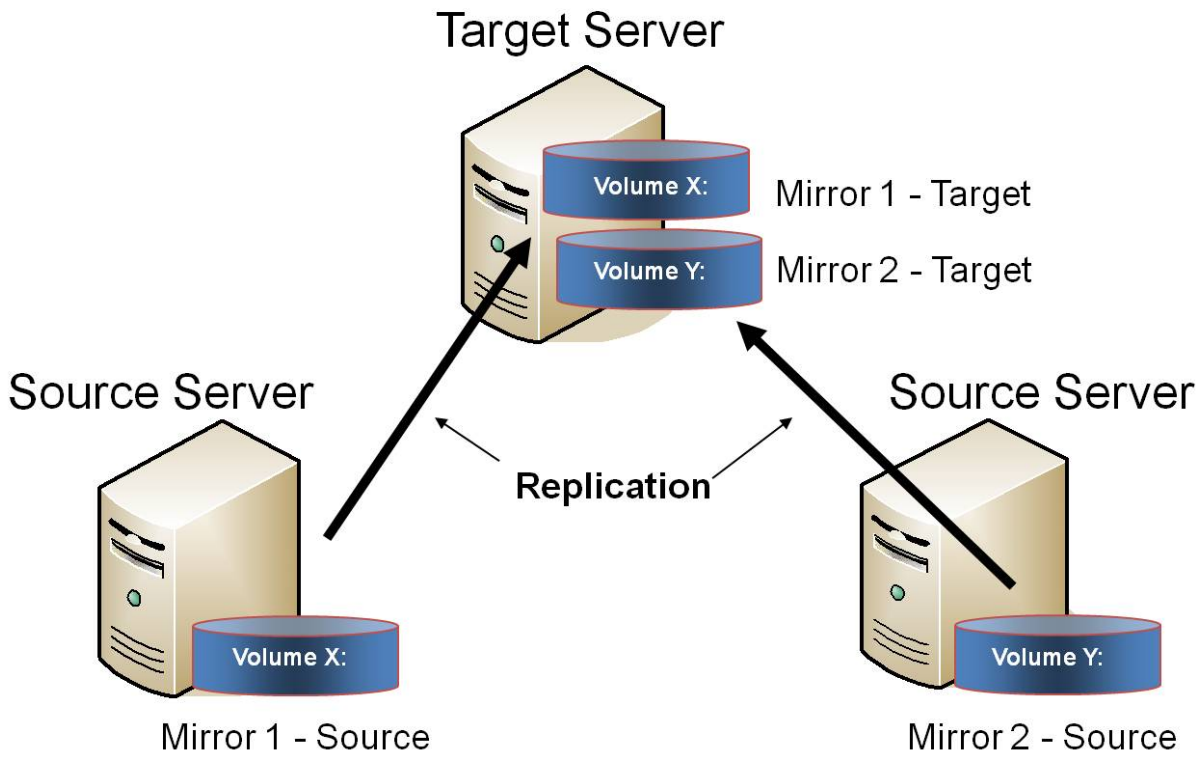
Note that there are two mirrors that are completely independent of each other. The mirrors might be using different networks, they may have different compression or bandwidth throttle settings and they may be in completely different states (e.g. Mirror 1 -- Mirroring, Mirror 2 -- Resyncing).

Example: USE CASE	Replicate data to one target server that resides locally in the same site with the primary server and replicate another copy of the data to a remote site for disaster recovery purposes should something happen to the first site.
Example: USE CASE	To periodically replicate or "push" data to multiple target systems from a single source system.

Additional topics of interest include:

Many-to-One

This configuration involves multiple source servers replicating one (or more) volumes to the same target system. In this configuration, each volume being replicated to the target server must have a unique drive letter.



Note: This is actually two One-to-One mirrors.

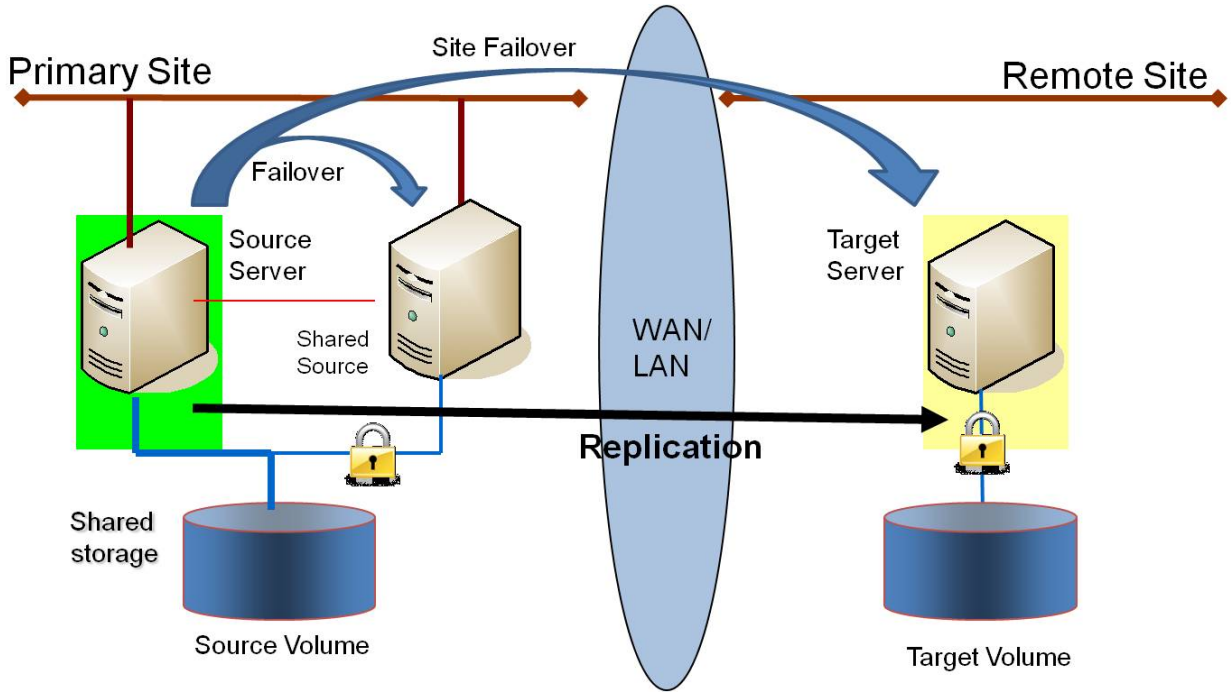
Example: USE CASE	Users may wish to replicate multiple branches back to a single data center for backup consolidation and disaster recovery purposes.

Additional topics of interest include:

- Primary Server Shutdown
- Secondary Server Failures
- Using DataKeeper Standard To Provide Disaster Recovery For Hyper-V Virtual Machines
- Frequently Asked Questions

N-Shared-Disk Replicated to One

This configuration allows you to replicate the shared volume(s) of the primary site to a remote system across the network.



This configuration is ideal for providing local failover within the Primary Site and disaster recovery protection should the entire Primary Site go down.

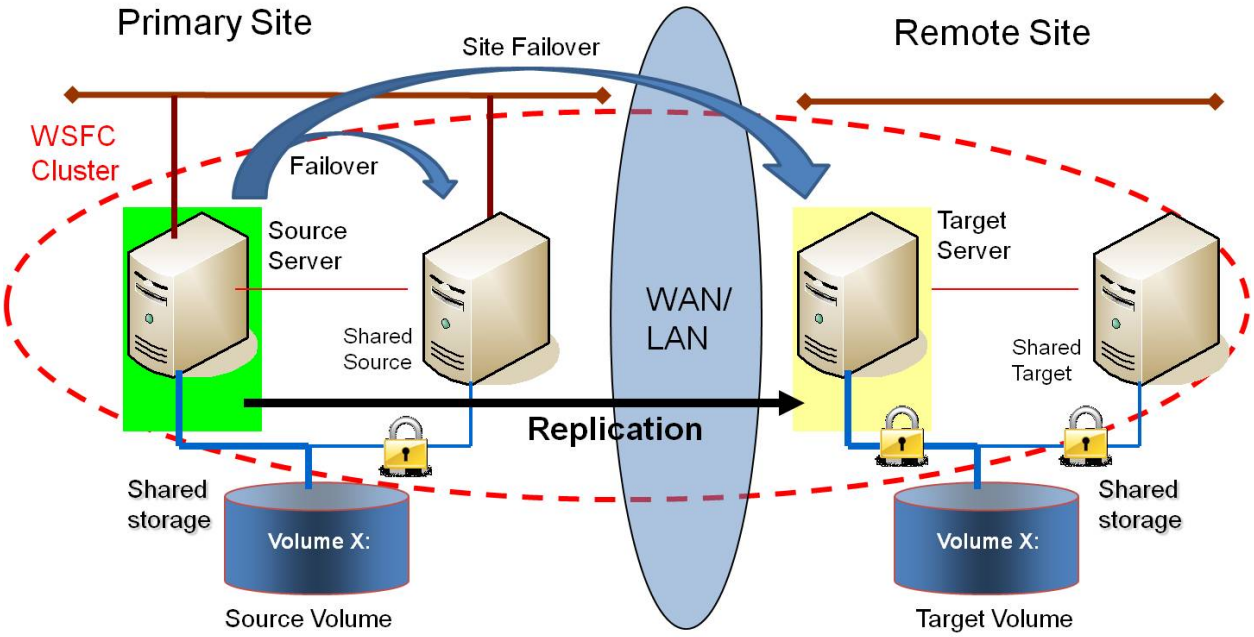
Example: USE CASE	Extend your MSCS or WSFC cluster to a DR site by replicating the shared volume to a remote target. In the event of a primary site outage, the remote server becomes the active server.

Additional topics of interest include:

- Creating Mirrors with Shared Volumes
- Managing Shared Volumes
- Adding a Shared System
- Removing a Shared System
- Frequently Asked Questions

N-Shared-Disk Replicated to N-Shared-Disk

This configuration replicates data between sites where each site utilizes shared storage.



Note that the number of systems in the Primary Site does not have to equal the number of systems in the Remote Site.

Also note that only the Source Server has access to the Source Volume. Shared Source systems and all systems on the target side cannot access the volume and are locked from the file system's perspective.

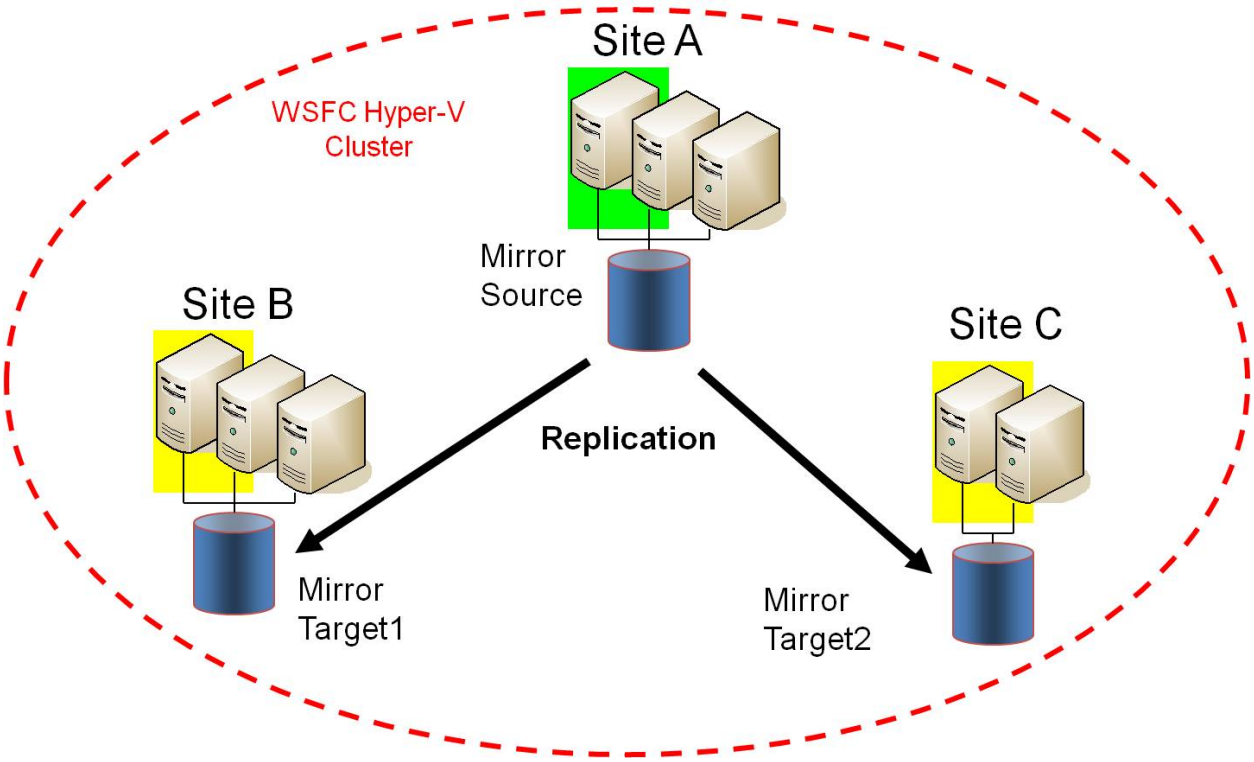
Example: USE CASE	Users who wish to provide the same level of availability in their DR site will deploy this configuration to ensure that regardless of what site is in service, the availability level stays the same.
Example: USE CASE	Where Hyper-V clusters are configured with virtual machines distributed across many cluster nodes, it is important to have a similar number of cluster nodes available in the disaster recovery sight to ensure that the resources are available to run all of the virtual machines in the event of a disaster.

Additional topics of interest include:

- Creating Mirrors with Shared Volumes
- Managing Shared Volumes
- Adding a Shared System
- Removing a Shared System
- Frequently Asked Questions

N-Shared-Disk Replicated to Multiple N-Shared-Disk Targets

This is a complex configuration which combines the aspects of replicating a shared storage environment to multiple shared targets.



Example: **USE CASE**	Users who wish to provide the same level of availability in their DR sites will deploy this configuration to ensure that regardless of what site is in service, the availability level stays the same.

Additional topics of interest include:

- Creating Mirrors with Shared Volumes
- Managing Shared Volumes
- Adding a Shared System
- Removing a Shared System
- Frequently Asked Questions

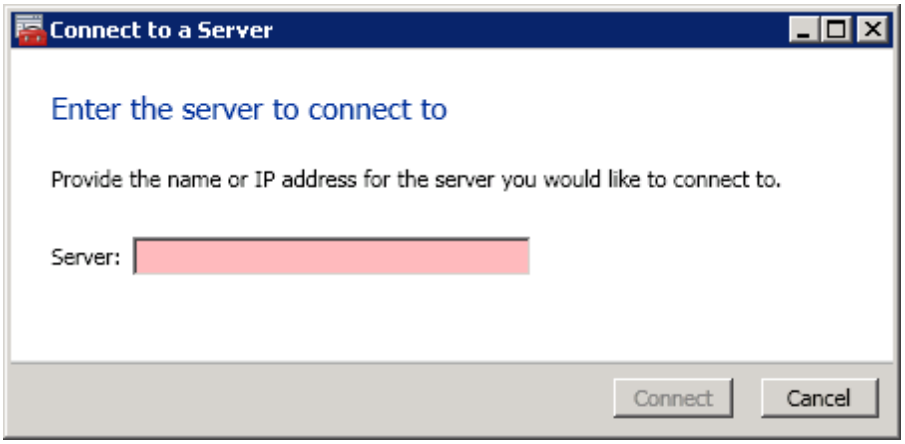
Setting Up SteelEye DataKeeper

Follow these steps to start using SteelEye DataKeeper:

1. Connect to the servers you wish to configure for replication. You can select **Connect to Server** from the **Action** pull down menu, right-click on the job folder in the left panel tree display and select **Connect to Server** or choose **Connect to Server** from the **Actions** pane.
2. Create a Job. From the right **Actions** pane, select **Create Job** or you can right-click on the job folder in the left panel tree and select **Create Job**.
3. Create a mirror for the new job.

Connecting to a Server

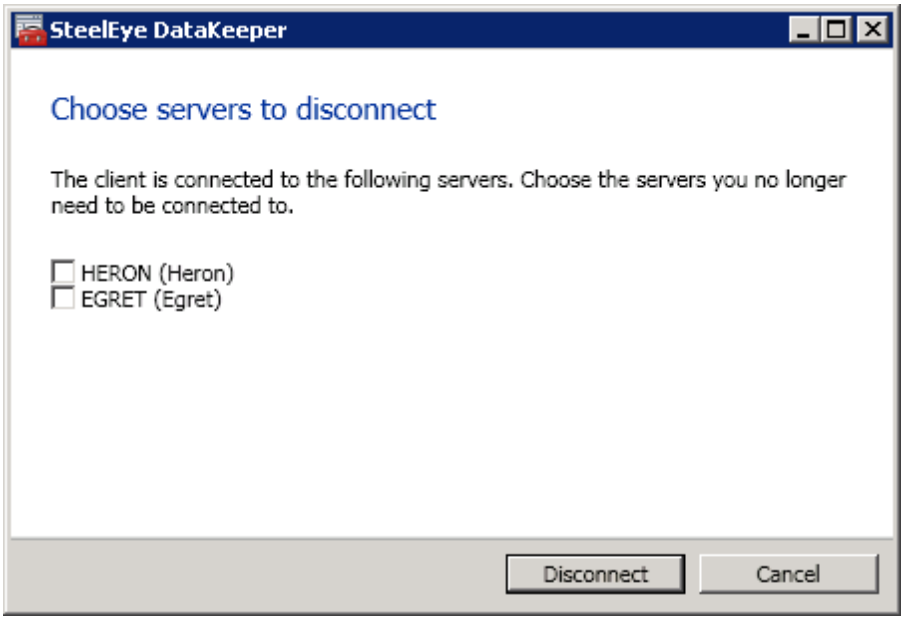
Use this dialog to connect to the server of your choice. You may enter the IP address, system NetBIOS name or the full system domain name for the server. Click **Connect** to select it.



Disconnecting from a Server

Use this dialog to disconnect from a server. You may use this option if you no longer wish to view the server in the Administration Window.

From the list of servers, select the server(s) that you wish to disconnect from and click **Disconnect**.



Creating a Job

1. If not already connected, connect to the server where you want to create a job.
2. From the right **Actions** pane, select **Create Job**. The **Job Wizard** will prompt you for a **Job Name** and **Description**.
3. Enter the appropriate information and select **Create Job** to finish.
4. You will immediately be prompted to Create a Mirror for this job.

Configuring Mirrors

Creating a Mirror

Before creating a mirror, ensure the following:

- You have created a job to hold the mirror.
- The volume on both the source and target systems must be of the **NTFS** file system type.

- The target volume must be greater than or equal to the size of the source volume.
- If the volume will be configured on a **Dynamic Disk**, create the dynamic volume first, then reboot the system before continuing with mirror creation (see the Mirroring with Dynamic Disks Known Issue for further information).
- See Volume Considerations for more information, including what volumes cannot be mirrored.
- You must be connected to both the source and target server before creating the mirror. Use the Connect to Server link in the **Actions** pane or in the **Mirror Create** dialog box.

Creating the Mirror

1. Select **Create a Mirror** from the right column **Actions** task pane. The **Choose a Source** dialog box appears.
2. Enter or choose the **Server Name** for the source volume. You can select the **Connect to Server** link below this field to connect to the server at this time.
3. Choose the **IP address** that is on the subnet you wish to use for the replication traffic.
4. Enter or choose the **Volume** to be used on the selected server. Select **Next**. The **Choose a Target** dialog box appears.
5. Enter or choose the server with the **Target Volume**. If necessary, you can select the **Connect to Server** link at this time.
6. Choose the **IP address** that is on the subnet you wish to use for the replication traffic.
7. Enter or choose the **Volume** to be used on the selected server. Press **Next** to continue. The **Configure Details** dialog box will display.
8. Use the slide bar to set the **data compression level** for data sent from the source to the target system. **Note**: Compression is only recommended to be used when replicating across WAN connections.
9. Select how (Asynchronously or Synchronously) the source volume data should be sent to the target volume.
10. If you wish to limit the amount of bandwidth used by replication, enter the **maximum bandwidth** for transmission; otherwise, leave the default setting. Select **Done**. The job with the new mirror will appear in the left tree pane and the main window displays.

    Note: After creating a mirror, its initial state may be displayed as **Resync Pending** in the **Summary** pane. When the initial mirror resynchronization completes, its state will automatically switch to the **Mirror** state.

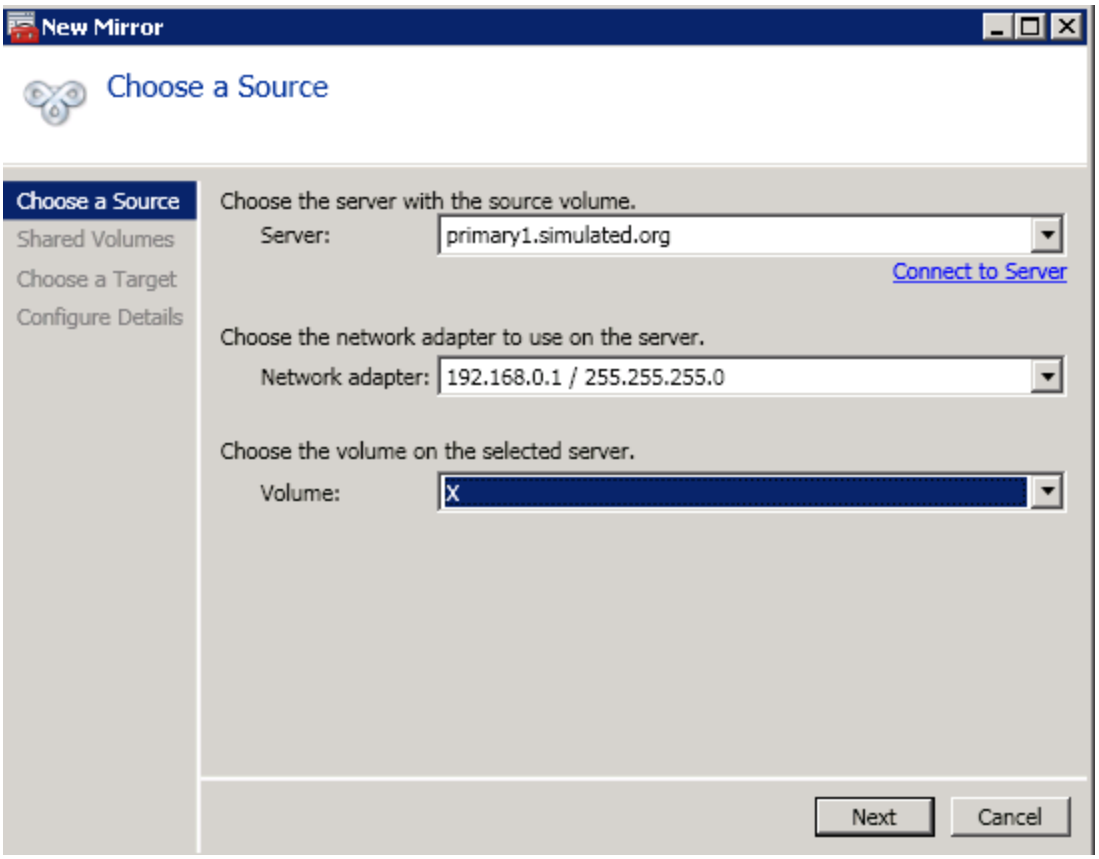
Creating Mirrors With Shared Volumes

In order to properly configure DataKeeper in a shared volume configuration, use the **DataKeeper GUI** to connect to all systems where the shared volumes are configured. When connected, the DataKeeper GUI uses hardware signatures to automatically detect which volumes are shared and which are not.

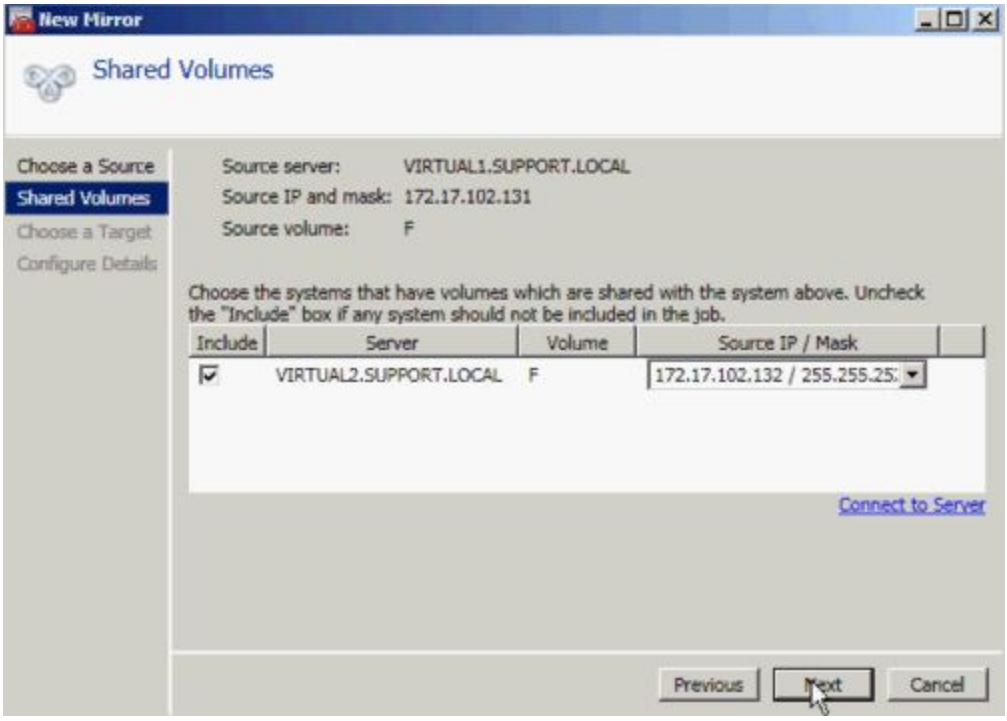
Important: If the GUI is not connected to a system, the GUI cannot detect shared volumes on that system.

Note: DataKeeper allows mirrors to be created on shared volumes where more than one system has access to the same physical storage. To prevent simultaneous access, see Safe Creation of a Shared-Storage Volume Resource prior to performing the following steps.

1. Connect to all systems via the **DataKeeper GUI**.
2. Choose Create Job.
3. Define a job name and job description and select **Create Job**. The **Choose a Source** dialog box appears.



4. Choose a **Source System**, **IP Address** and **Volume**.
5. Select **Next**. The **Shared Volumes** dialog box appears.



6. Choose the systems that have volumes which are shared with the source system.

   Note: If using Windows Server 2012, please see the Known Issue regarding this dialog.

   Note: All systems connected to the shared volumes must be configured with IP addresses on the same subnet. The **Next** button will not be enabled until all included systems have a valid IP address.

   While it is possible to uncheck the **Include** box for a given system, the user should be very careful to make sure that the volume listed really is not a shared volume. It is possible (although unlikely) that the hardware signatures of two volumes will match even if they are not shared. In this case, it is valid for the user to uncheck the **Include** box.

7. Select **Next**. The **Choose a Target** dialog box appears.
8. Choose a **Target System**, **IP Address** and **Volume**.
9. Select **Next**.

   Note: If there are volumes on other systems that are shared with this target volume, the **Shared Volumes** dialog will appear next. Configure these shared target volumes as you would for shared source volumes, described above.

10. Select **Next** to continue. The **Configure Details** dialog box appears.
11. Use the slide bar to set the **data compression level** for data sent from the source to the target system.

    Note: Compression is only recommended to be used when replicating across WAN connections.

12. Select how (Asynchronously or Synchronously) the source volume data should be sent to the target volume.
13. If you wish to limit the amount of bandwidth used by replication, enter the maximum bandwidth for transmission; otherwise leave the default setting.
14. Select **Done**. The job with the new mirror will appear in the left tree pane and the main window displays.

Safe Creation of a Shared-Storage Volume Resource

DataKeeper allows mirrors to be created on shared volumes where more than one system has access to the same physical storage. The shared volume can be on the source side of the mirror or on the target side.

In order to safely create a shared-storage volume resource, the user must ensure that only one system has write access to the volume at any time. This includes the time prior to the creation of the DataKeeper mirror. Since DataKeeper doesn't know that the volume is shared before a mirror is created, manual steps must be taken to ensure that the volume is never writable on two or more systems at the same time.

To protect the volume from simultaneous write access, use the following procedure. In this example, two systems - *SYSA* and *SYSB* - are connected to shared storage, then replicated to a third system, *SYSC*, the target system. This storage is configured with two volumes which should be assigned drive letters *E:* and *F:* on all three systems.

1. Power on *SYSA*, while leaving *SYSB* powered off.
2. Install DataKeeper if it has not been installed.
3. Assign drive letters *E:* and *F:* to the volumes; format with NTFS if not formatted yet.
4. Power off *SYSA*.
5. Power on *SYSB*.
6. Install DataKeeper if it has not been installed and reboot the system after the installation.
7. Assign drive letters *E:* and *F:* to the shared volumes.
8. In a command prompt, run the following commands to set the "shared" config flag:

```
"%ExtMirrBase%\emcmd" . setconfiguration E 256

"%ExtMirrBase%\emcmd" . setconfiguration F 256
```

9. Reboot *SYSB*. It will come up with the *E:* and *F:* drives locked.
10. Power on *SYSA*. It will come up with the *E:* and *F:* drives writable.
11. Use the DataKeeper GUI to create a job and mirror from *SYSA E*: (source) to *SYSC E*: (target) and from *SYSA F*: (source) to *SYSC F*: (target). DataKeeper will detect that *SYSB* is a shared source system.

An alternative to powering the systems off is to use **Disk Management** to take the shared physical disk offline.

This procedure can also be used to safely create a mirror on a shared target volume. In the example above, the mirror could have been created from *SYSC* to *SYSA* - in that case, the volume *SYSB* would be a shared target.

If you have more than two shared systems at a site, this same procedure can be used to lock the volume on all systems that will not be part of the initial mirror.

Creating Mirrors With Multiple Targets

SteelEye DataKeeper provides the ability to replicate data from a single source volume to one or more target volumes. In addition, DataKeeper also allows you to switch over control and make any of the target volumes become the source. Assuming you have already created a job with a mirror using the Create a Mirror procedure, use the following procedure to create a second mirror from the same source volume to a different target volume:

1. Right-click on an existing job.
2. Choose the **Create a Mirror** action.
3. Choose the **source** of the existing mirror (as this will also be the source of the new mirror).
4. Choose the **target** for the new mirror.
5. Select **Done**.

   The next dialog displayed prompts you for additional information that DataKeeper requires to be able to properly switch over the source volume to one of the target volumes. You already specified the network endpoints between the source system and the first target system when you created the first mirror. You also specified the network endpoints between the source system and the second target system when you created the second mirror.

   The final piece of information DataKeeper requires is the network endpoints of a (potential) mirror between the first target system and the second target system so that no matter which system becomes the source of the mirrors, mirrors can be properly established between all three systems.
6. On the **Additional Information Needed** dialog, choose the **network endpoints** that will be used to create a mirror between the first target system and the second target system.

   Note: If using Windows Server 2012, please see the Known Issue regarding this dialog.

   This mirror will not be created now. DataKeeper is simply storing these mirror endpoints for future use.
7. Select **OK**.

   Note: If you are replicating a single source volume to more than two target volumes, you will have to provide network endpoints for mirrors between all of the systems involved.

   Examples:

3 Nodes (A,B,C) - Define Endpoints for Mirrors	
Created Mirrors	**Additional Mirror Relationships**
A → B	B → C
A → C	

4 Nodes (A,B,C,D) - Define Endpoints for Mirrors	
Created Mirrors	**Additional Mirror Relationships**
A → B	B → C
A → C	B → D
A → D	C → D

Switchover and Failover with Multiple Targets

In a multiple target configuration, it is important to understand how DataKeeper mirrors will work in the following scenarios:

- Manual switchover to a target server
- Source server failure followed by a manual switchover to a target server

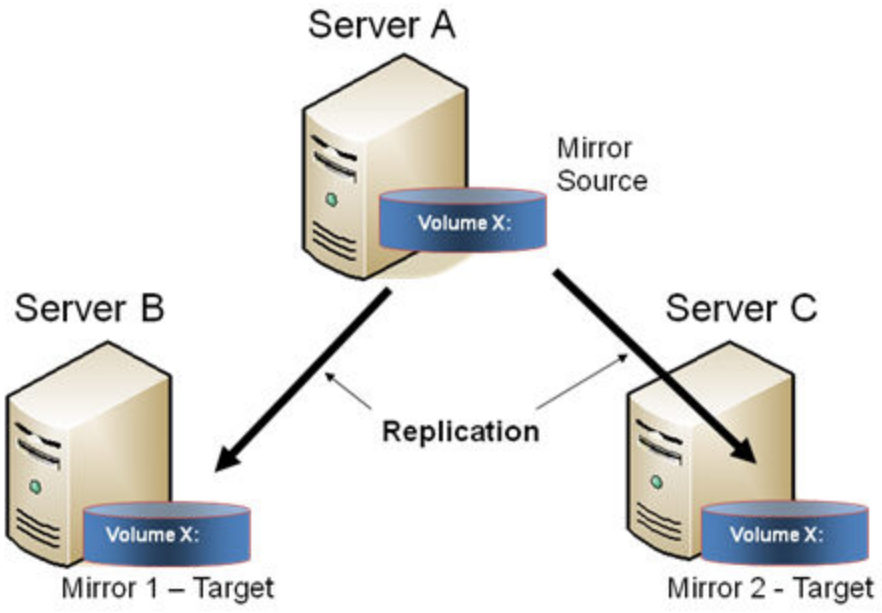
Example:

In the following scenario, there are three servers:

- Server A (source)
- Server B (target 1)
- Server C (target 2)

Note that there are two separate mirrors and Server A is replicating to two different target volumes.

- Mirror 1: Server A → B
- Mirror 2: Server A → C



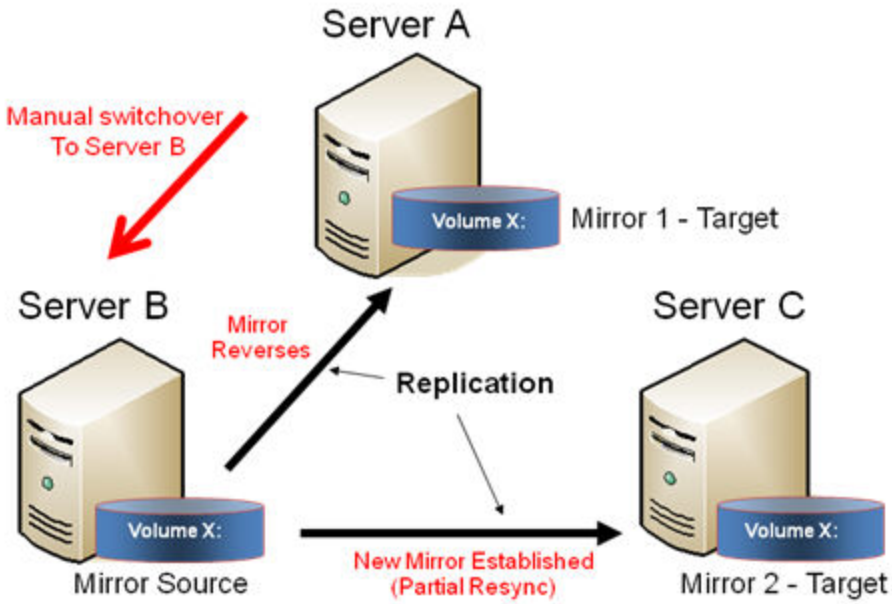
Manual Switchover to a Target Server

In the event the administrator wants to make Server B become the active (source) server, the following actions will occur:

1. Administrator initiates a switchover to Server B via the **Switchover Mirror** option in the DataKeeper UI.
2. Server A flushes its data to the source volume.
3. Mirror 1 is automatically deleted and recreated from Server B to Server A.
4. The mirror between Server A and Server C is also automatically deleted. (**Note**: There will be a few seconds delay noticed in the DataKeeper GUI; this delay can take some time based on network bandwidth and server performance.)
5. A new mirror is established between Server B and Server C. The intent log from Server A is copied to Server B. Only a partial resync of the data between Server B and Server C is required to bring them in sync. (A partial resync is the resynchronization of only the necessary data to establish the new end points and is usually much quicker than a full resync.)

   RESULT

- Mirror 1: Server B → A (partial resync)
- Mirror 2: Server B → C (copy intent log from Server A, partial resync)



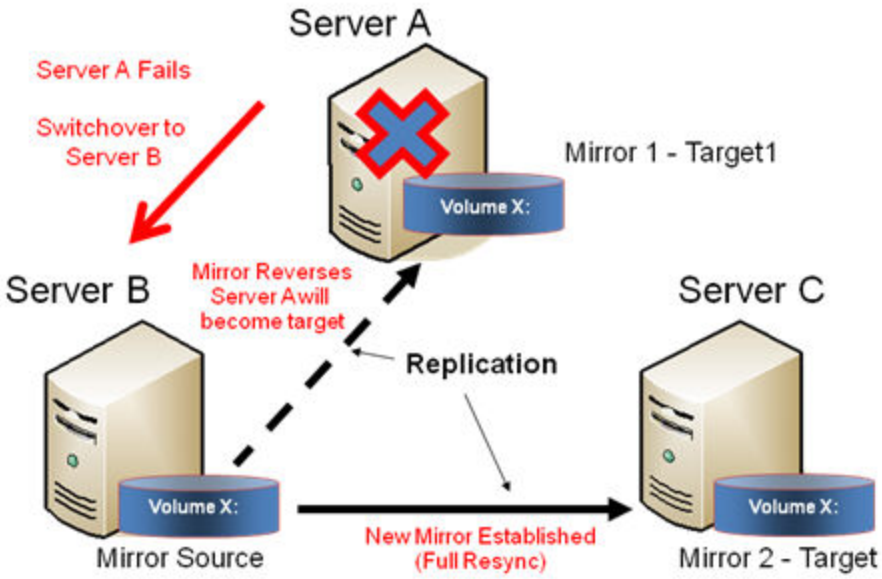
Source Server Failure - Manual Switchover to a Target Server

In the event the active (source) server fails, DataKeeper allows you to make Server B become the active (source) server. The following actions will occur:

1. Server A fails.
2. Administrator initiates a switchover to Server B via the "Switchover Mirror" option in the DataKeeper UI.
3. Server B deletes the local side of the mirror and creates a new mirror from Server B to Server A.
4. The mirror between Server A and Server C is deleted.
5. A new mirror is established between Server B and Server C. Because the intent log on the original source server (Server A) is not available, a full resync of the volume between Server B and Server C is required. The duration of a full resync depends on the size of the drive, server performance and network bandwidth and will usually take a longer time for large amounts of data.
6. When Server A comes back up, Server A detects that Server B became the source of the mirror while Server A was down and Server A automatically becomes the target of the mirror.

   RESULT

- Mirror 1: Server B → A (partial resync when Server A comes back up)
- Mirror 2: Server B → C (full resync)



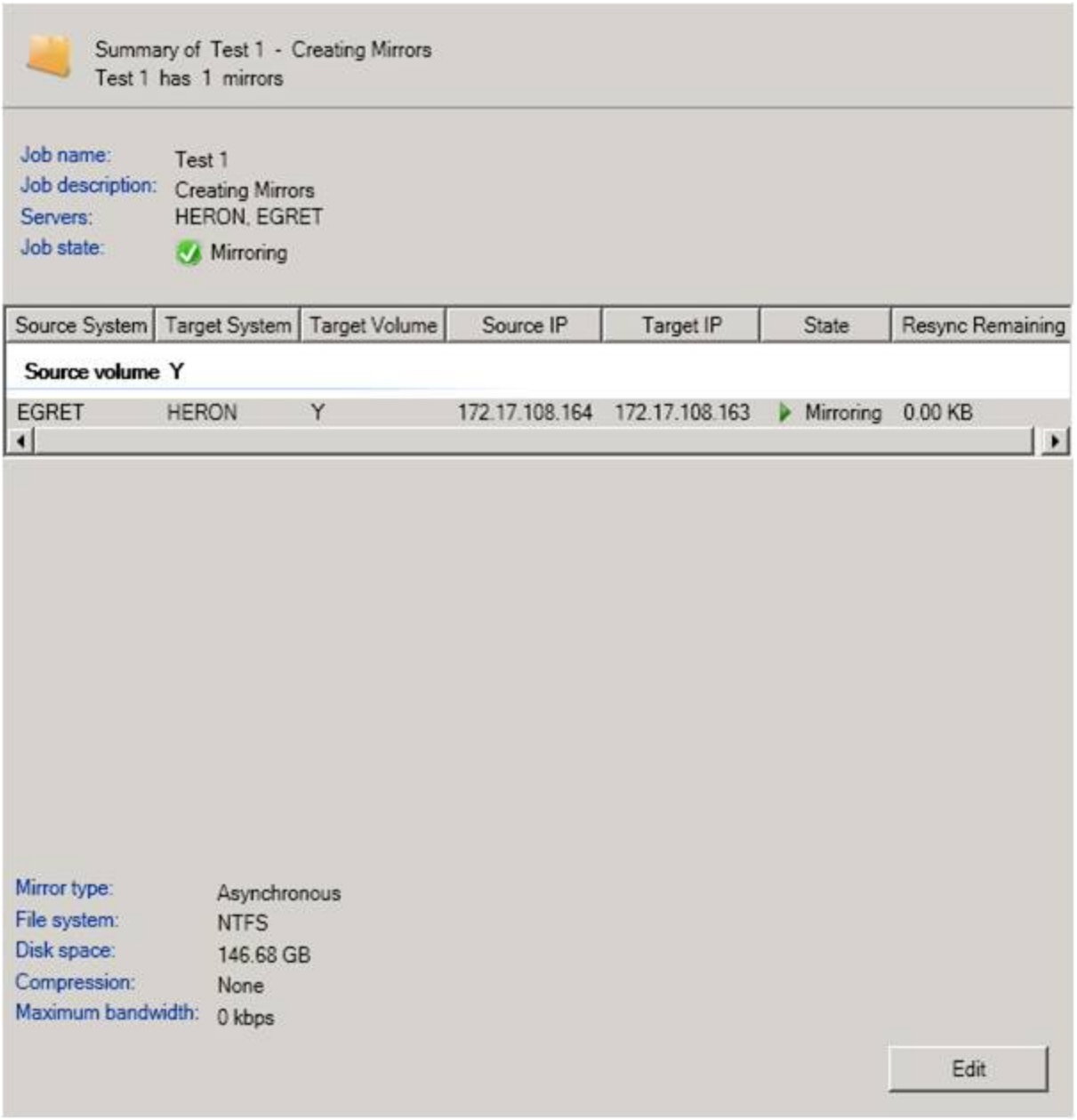
Working With Jobs

Jobs

For ease of use and configuration, SteelEye DataKeeper does much of its management of mirrors through an entity called a job. A job is a logical grouping of related mirrors and servers. This feature allows you to create a job for complex repetitive tasks and run them quickly from the SteelEye DataKeeper user interface.

Mirrors that are related should be placed in a single job. For instance, multiple mirrors protecting an application like SQL Server should be placed in the same job. Mirrors that are unrelated should be placed in separate jobs.

Note: Mirrors created in previous versions of SteelEye Data Replication will be imported as individual jobs. The administrator must take care to edit these jobs to ensure that mirrors are logically grouped together.



Renaming a Job

1. Select the job in the left **Console Tree** pane of the main DataKeeper window.
2. You can select **Rename Job** from the **Actions** pane or right-click on the selected job and choose **Rename Job** from the menu that displays.
3. Enter the new **Job Name** and new **Job Description**.

Deleting a Job

1. Select the job in the left **Console Tree** pane of the main DataKeeper window.
2. You can select **Delete Job** from the **Actions** pane or right-click on the selected job and choose **Delete Job** from the menu that displays.
3. Select **Yes** to delete the selected job and associated mirror(s).

Reassigning a Job

Use the **Reassign Job** function to move an existing mirror from one job to another without deleting the mirror.

1. Select the job from the middle **Summary** panel.
2. Right-click and select **Reassign Job** or select **Reassign Job** from the **Actions** panel.
3. Select an existing job from the **Existing Jobs** dropdown list and press the **Assign Job** button. The new job assignment will display in the middle **Summary** panel.

Note: You can also choose to **Create a New Job** from this dialog if you do not want to use an existing job.

Switching Over a Mirror

The Switchover Mirror function enables you to switch over all the mirrors in a job or just one of the mirrors in a job. A "mirror" includes all variants for mirrors such as standard single-target replication and complex geometries such as multi-target replication and shared node sources and targets. These complex mirror configurations and geometries actually implement a related collection of individual mirrors working as a single unit.

Note: Before switching over a mirror to the current target system, the mirror must be in the **Mirroring** state. Please see the **Requirements for Switchover** table below to understand switchover requirements in multiple target and shared source/target configurations. Please use the DataKeeper GUI to view the state of the mirror; the WSFC GUI will not provide that level of detail and will state that the resources are on-line (Green) even when the mirrors are not in the mirroring state.

1. Select the job in the left column tree pane.
2. Right-click on the selection and select **Switchover Mirrors**.
3. A dialog displays allowing you to designate which node/host(s) in the selected job or mirror should become the new mirror source.

   In the case of complex mirrors, it is valid to choose either a shared peer of the current mirror source or any one of the active targets that are currently in the mirroring state. Choosing a shared peer of an active target or one that is not currently mirroring will result in an error and leave the current mirror status and configuration unchanged.
4. An hour glass will appear over the mirror icon in the left tree panel.
5. You can confirm the switchover is complete by checking the mirror status in the **Summary** panel.

Note: If the **Switchover** option is grayed out (not available), this could mean the volume is under clustering protection (Microsoft clustering or SteelEye Protection Suite clustering).

Requirements for Switchover

Configuration Type	Example Configuration	Switchover Action	Requirements for Switchover
Single Target	A → B	Switchover to B	Allowed if mirror is in MIRRORING STATE
Multiple Target	A → B (mirroring) A → C (paused)	Switchover to B	Allowed because A→B mirror is in MIRRORING state
		Switchover to C	Not allowed
Shared Source/Target	*S1,S2,S3 → *T1,T2 (S1 is current source) (T1 is current target)	Switchover to shared source (S2 or S3)	Always allowed
		Switchover to current target (T1)	Only allowed if mirror in MIRRORING state
		Switchover to shared target (T2)	Not allowed -- Switchover will fail

Working With Mirrors

Managing Mirrors

From the **Actions** pane, you can select a job and manage all the mirrors in a job, or you can perform an action on a single mirror in a job.

After selecting a job, you can:

- Pause and Unlock All Mirrors
- Continue and Lock All Mirrors
- Break All Mirrors
- Resync All Mirrors
- Switchover All Mirrors

The target-level actions (at the bottom of the **Actions** pane) are for individual mirrors. For example, if you have a job with two mirrors and you select one of the mirrors then choose the target **Pause and Unlock Mirror** action, only the selected mirror would be paused.

Pause and Unlock

This command pauses the mirror and unlocks the volume on the target system. You may wish to unlock the target volume in order to make a backup of the volume.

Warning: Any writes to the target volume while it is unlocked will be lost when the mirror is continued. Also, if a reboot or shutdown is performed on the target system while a volume is unlocked, a full resync will occur when the mirror is continued. To prevent the full resync in this case, be sure to perform a "**Continue and Lock**" prior to rebooting or shutting down the target system.

Note: If *replacing* the target volume, either break the mirror or delete the mirror in order to ensure a full resync of the data from the source volume to the new target volume when the new target volume is in place. See Replacing a Target for further information.

The Continue and Lock command will relock the target volume, perform a partial resync **(or full resync if the target has been rebooted or shut down while unlocked)** and resume the mirroring process.

1. Select the job that contains the mirror you want to unlock.
2. Right-click on the job selection and choose **Pause and Unlock All Mirrors** or select **Pause and Unlock All Mirrors** from the **Actions** task pane.
3. Select **Yes** to pause and unlock all mirrors in the selected job.

Continue and Lock

This action locks the volume on the target system and then resumes the mirroring process.

While the mirror is paused, writes on the source system are recorded in the SteelEye DataKeeper Intent Log. When the **Continue and Lock** operation occurs, these changed blocks - along with any blocks that also changed on the target volume - are sent from the source to the target, and the mirror is resynchronized in what is called a Partial Resync.

Warning: Any writes to the target volume while unlocked are lost when the mirror is continued. Also, if a reboot or shutdown is performed on the target system during the Pause and Unlock, a full resync will occur when the Continue and Lock is performed.

Note: If *replacing* the target volume, either Break the mirror or Delete the Mirror, which requires either a **Resync** or **Recreate** instead of Continue and Lock. See Replacing a Target for further information.

1. Select the job that contains the mirror you want to continue.
2. Right-click on the job selection and choose **Continue and Lock All Mirrors** or select **Continue and Lock All Mirrors** from the **Actions** task pane.
3. Select **Yes** to continue and lock all mirrors in the selected job.
4. The mirror state will change to **Mirroring** in the **Mirror Summary** window.

Partial Resync

A partial resync is the resynchronization of only the necessary data to establish the new end points and is

usually much quicker than a full resync.

Break

Breaking a mirror is similar to the **Pause and Unlock** function. It suspends mirror operation and unlocks the target volume for read/write access. The difference is that the **Break** operation marks all bits in the DataKeeper Intent Log as dirty, which forces a full resync to occur when the mirror is resync'ed to resume mirroring.

Warning: Do not write to the target volume while the mirror is broken. Any writes to the target while the mirror is broken will be lost when the mirror is resynchronized.

1. Select the job that contains the mirror you want to break.
2. Right-click on the job selection and choose **Break All Mirrors** or select **Break All Mirrors** from the **Actions** task pane.
3. Select **Yes** to break all mirrors in the selected job.
4. The mirror state will change to **Broken** in the **Mirror Summary** window.

Note: The **Resync** command will relock the Target volume, perform a **full resync** and resume the mirroring process.

Resync

Use this command to re-establish a broken mirror. A full resync will be performed.

1. Select the job that contains the mirror you want to resync.
2. Right-click on the job selection and choose **Resync All Mirrors** or select **Resync All Mirrors** from the **Actions** task pane.
3. Select **Yes** to resync all mirrors in the selected job.
4. The mirror state will change to **Mirroring** in the **Mirror Summary** window.

Deleting a Mirror

This action discontinues replication and removes the mirror from the associated job. The target volume is unlocked and made fully accessible.

1. Select the job that contains the mirror you want to delete.
2. Right-click on the mirror and choose **Delete Mirror** or select **Delete Mirror** from the **Actions** task pane.
3. Select **Yes** to delete the mirror.
4. The mirror will be deleted and removed from the associated job.

Note: If the **Delete Mirror** option is grayed out (not available), this could mean the volume is under clustering protection (Microsoft clustering or SteelEye Protection Suite clustering).

Replacing a Target

When replacing the target volume, you must either break the mirror or delete the mirror in order to ensure a full resync of the data from the source volume to the target volume when the target volume is back in place. Though similar to the Pause and Unlock, breaking the mirror marks all bits in the DataKeeper Intent Log as dirty which forces a full resync to occur. Deleting the mirror discontinues replication altogether removing the mirror from the job so that when your mirror is recreated with the new target, a full resync will be performed.

Using the BREAK Command

1. Select the mirror that contains the target you want to replace.
2. Right-click on the mirror and choose **Break Mirror** or select **Break Mirror** from the **Actions** task pane.
3. Select **Yes** to break the mirror.
4. Once new target is in place, right-click on the mirror and choose **Resync Mirror** or select **Resync Mirror** from the **Actions** task pane.
5. The target volume will be locked, a full resync will be performed and the mirroring process is resumed.

Using the DELETE Command

1. Select the mirror that contains the target you want to replace.
2. Right-click on the mirror and choose **Delete Mirror** or select **Delete Mirror** from the **Actions** task pane.
3. Select **Yes** to delete the mirror.
4. Once new target is in place, recreate the mirror.

DataKeeper Volume Resize

DataKeeper allows users to extend and shrink their DataKeeper volumes dynamically while retaining mirror settings. Once the resize is complete, a partial resync will be performed.

Note: This resize procedure should be performed on only one volume at a time.

WARNING: Do NOT attempt to perform the resize in releases prior to DataKeeper for Windows v7.4.

Non-Shared Volume Procedure

Example configurations for using this procedure include the following:

- Disk-to-Disk
- One-to-One

- One-to-Many (Multiple Targets)
- Many-to-One

To resize your DataKeeper volume in a non-shared volume configuration, perform the following steps.

1. Pause all mirrors and unlock all target volumes via the Pause and Unlock mirror option in the DataKeeper UI.
2. Using the **Windows Disk Management** utility, increase (or decrease if allowed by the Operating System) the volume size on the source system by selecting "**Extend Volume**" or "**Shrink Volume**" in the ***Resizing Wizard***. Once that resize is complete and verified, resize the target system(s). Make sure that the raw volume size of each target is greater than or equal to the size of the source volume.

   Note: The Windows Disk Management utility will take longer to start on the target node based on the number of drives. Because the Windows operating system has error condition retries built in when a volume is locked, the speed with which it starts on the "locked" target node is affected.
3. Continue and Lock the mirrors after volumes have been resized. The mirroring process should resume and a partial resync should occur.

Shared Volume Procedure - Basic Disk

This resizing procedure will work on shared volumes if the shared volume is configured on a **Basic Disk**. Example configurations for using this procedure include the following:

- N-Shared-Disk Replicated to One
- N-Shared-Disk Replicated to N-Shared-Disk
- N-Shared-Disk Replicated to Multiple N-Shared-Disk Targets

If there is free space on the disk, the volume can be extended to use the additional space.

1. Pause all mirrors and unlock all target volumes via the Pause and Unlock mirror option in the DataKeeper UI.
2. Shut down (power off) all shared source and/or shared target systems. (**Note**: Current source and current target systems should not be shut down.)
3. Change the volume sizes as noted above in the Non-Shared Volume procedure.
4. Continue and Lock the mirrors after resizing has completed.
5. Power on all shared systems. The new volume configuration will automatically be recognized.

Error Handling:

1. After performing the **Continue and Lock**, if the GUI abnormally maintains the "**Paused**" mirror state, check the system logs on both source and target nodes.
2. DataKeeper will prevent a mirror resync from starting if the target volume is smaller than the source volume. If the system logs show such an error, the target volume must be unlocked manually via the UNLOCKVOLUME command, and the volume must again be resized making sure that the volume

size of the target is greater than or equal to the size of the source volume. Then proceed with the Continue and Lock step above.

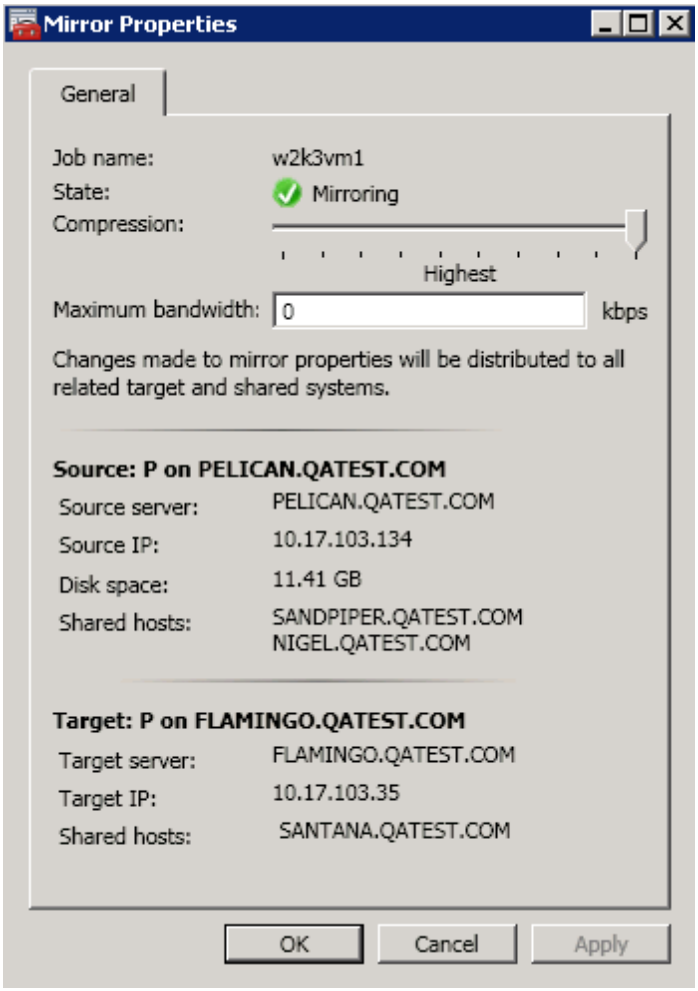
3. DataKeeper, upon continuing the mirror, will reallocate the bitmap file and in-memory bitmap buffer using the new volume size. In the event DataKeeper is unsuccessful in reallocating the bitmap buffer - due to insufficient memory resources on the source or target - the mirror will be placed into a '**Broken**' state which will require a FULL resync.
4. Once resizing a volume has begun, there is no way to back out of the resizing feature and the associated error handling as DataKeeper will have to reallocate the bitmap file and in-memory bitmap buffer. Any failure of this reallocation procedure will break the mirror and force a FULL resync.

Restrictions

- DataKeeper does not support changing the disk type of the physical disk where a mirrored volume is located (for example, **Basic Disk** to **Dynamic Disk --** mirror must be deleted prior to creating your dynamic disk).
- DataKeeper does not support third-party partition resizing products.
- DataKeeper does not support volume resizing on shared volumes configured on **Dynamic Disks**. Windows 2008 R2 and Windows 2003 R2 cannot reliably use a shared Dynamic Disk.

Mirror Properties

Select a job in the **Job Summary** pane and right-click to choose **Mirror Properties**.



This dialog displays the following information about the job, source and target systems:

- **Job Name**
- **State** (current state of the job; for example, Active)
- **Source System**
  - Server - name of source server
  - Source IP - IP address of source server
  - Disk Space - capacity of the source volume
  - Shared Hosts - other systems that have access to this volume via shared storage
- **Target System**
  - Server - name of target server
  - Target IP - IP address of target server

You can modify the following settings through the **Mirror Properties** dialog:

- Compression Level - specifies the compression level for the given mirror. The value can be set from lowest to highest. We recommend a level of "**Medium low**", but users should test several different settings to see what level works best in their specific environment. Compression is typically not required for LAN connections > 100 Mbps.

  Note: Any changes made to the compression level setting are automatically propagated to all the systems listed in the **Mirror Properties** display.

- Maximum Bandwidth - Specifies the maximum amount of network bandwidth (in kilobits per second) that a particular mirror is allowed to use. A value of 0 means unlimited.

Note: In a multi-target configuration where A is mirroring to B and C, the properties of the mirror between B and C cannot be set until B or C becomes the source.

Changing the Compression Level of an Existing Mirror

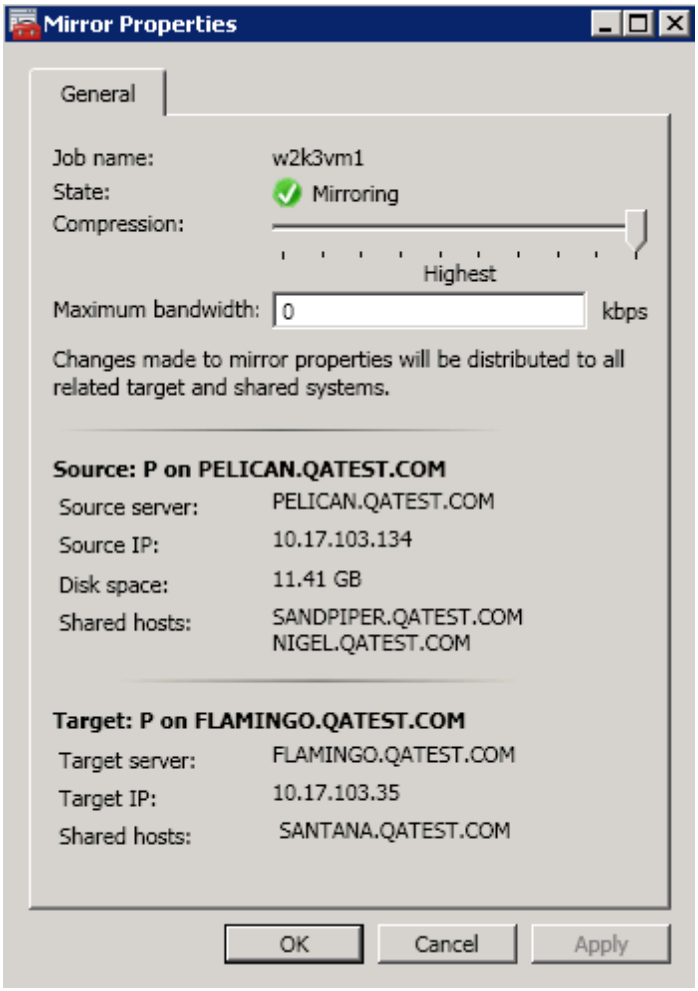
The compression level of the mirror is set during mirror creation and applies to that specific mirror only.

To change the compression level of an existing the mirror, edit the properties of the mirror from within the DataKeeper GUI.

1. Select the mirror and click on **Edit**.
2. Change the compression level by dragging on the slider button.

The values change from lowest to the highest. We recommend a level of “Medium low”, but users should test several different settings to see what level works best in their specific environment.

Also note that by changing the parameter as the comment suggests in the dialog, the compression properties will be propagated to all the systems listed in the Mirror Properties display.



Working With Shared Volumes

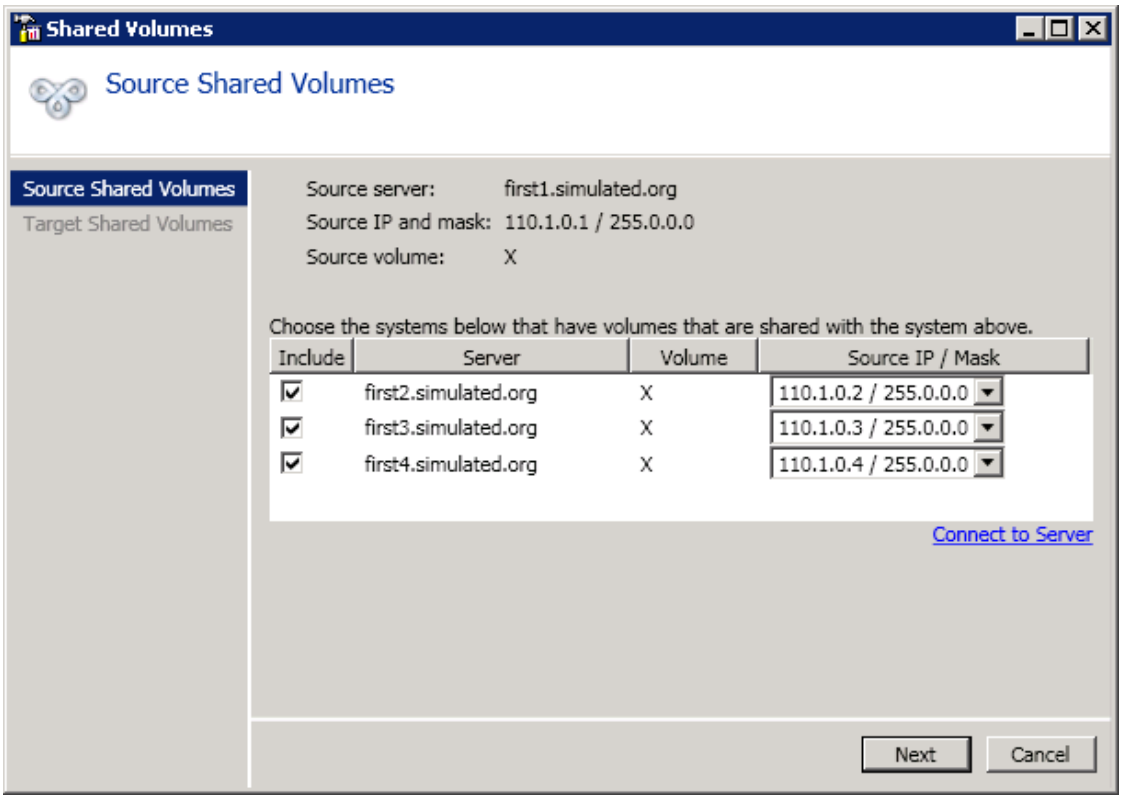
Managing Shared Volumes

Once your mirrors have been created, DataKeeper allows you to manage your shared volumes. By choosing **Manage Shared Volumes** from the DataKeeper GUI, you can add another system, which is sharing a mirrored volume, to a job. It also allows you to remove a shared system from a job. These systems can exist on either the source side or the target side of the mirror.

To add or remove a system that is sharing a mirrored volume on either the source or target end of a mirror, select the job that you want to manage and highlight the mirror that contains the volume that is to be edited.

If a volume is mirrored to more than one target and you want to add or remove a shared system on the source side of the mirror, you can choose any of the mirrors, since they all refer to the same source volume. Choose the **Manage Shared Volumes** action for that mirror, and the **Shared Volumes** dialog will appear.

If you want to add or remove a shared system on the target side of the mirror, you must select that specific mirror.



Adding a Shared System

To add a shared system to either the source or target side of a mirror, you must be connected to that system. You can connect to the system prior to starting the **Manage Shared Volumes** dialog, or you can click **Connect to Server** from within the dialog. In either case, if there are shared volumes that exist on that system that match either the source or target volume, the system and its matching IP address will be displayed in the correct page of the dialog. Leave the **Include** box checked to include the system in the job configuration and choose the correct IP address to be used for that system.

If a shared system does not have an IP address whose subnet matches the existing mirrored systems, the IP Address field will be blank and the **Include** box will be unchecked. You must reconfigure the system so that it has an IP address on that subnet. Then try adding the shared volume again.

When you click **Done** after adding a new shared system, it will be added to the job. If there are multiple mirrors in place, you will be asked to provide the network addresses to be used between the newly-added system and all other targets.

Removing a Shared System

To remove a shared system from either side of the mirror, bring up the **Manage Shared Volumes** dialog and

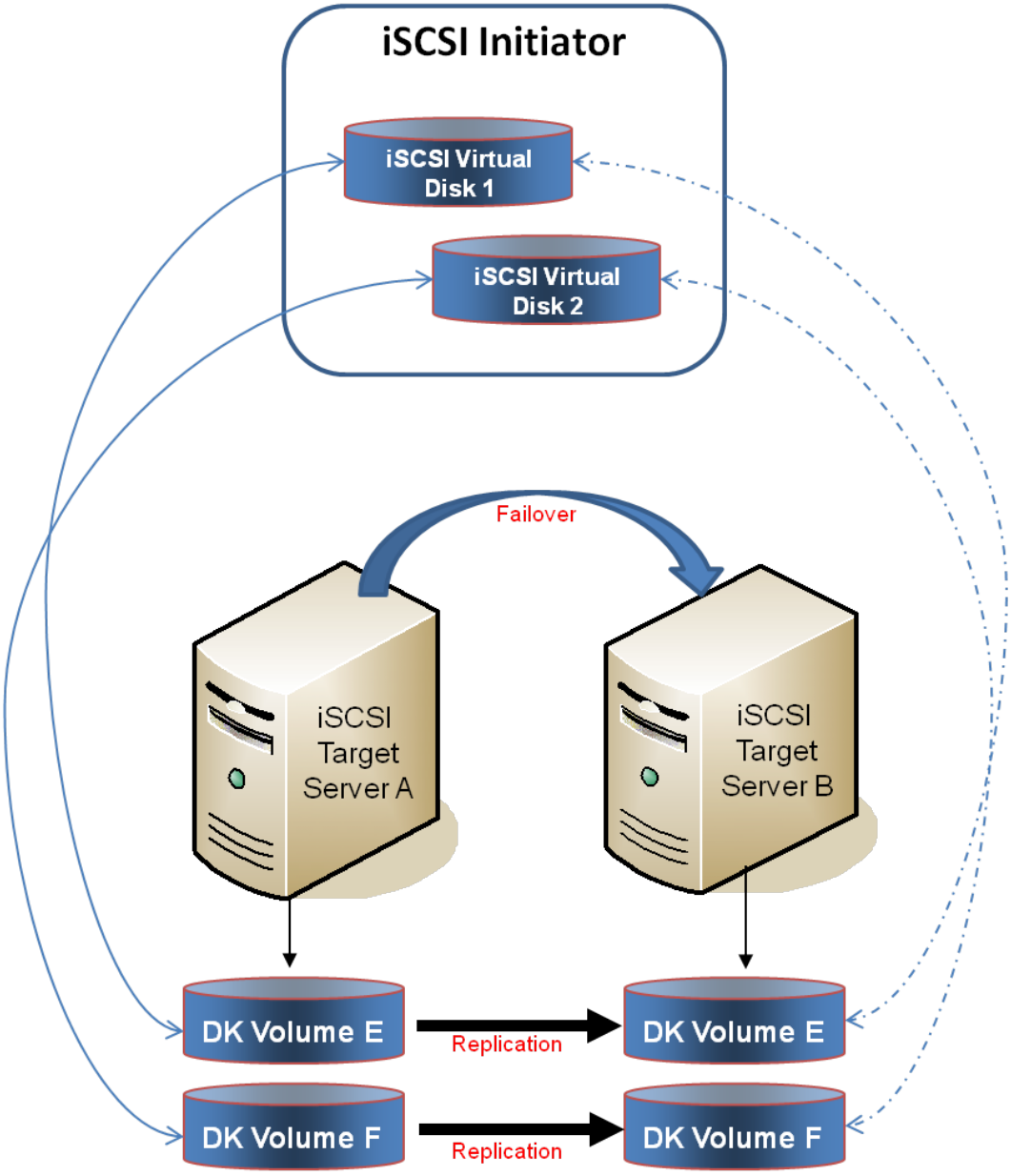
uncheck the **Include** box for the system to be removed. When you click **Done**, the job will be updated so that the system is not part of the job.

Warning: If a shared system is removed from the source side of the mirror, the source volume is now accessible on multiple systems and simultaneous access of the source volume could result in data corruption.

Using Microsoft iSCSI Target With DataKeeper on Windows 2012

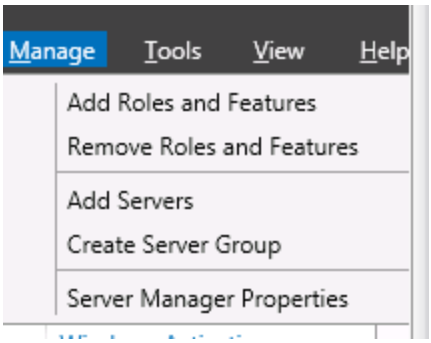
The following topics will guide you in setting up Microsoft iSCSI Target with DataKeeper via the user interface.

	**NOTE**: This configuration is not supported in a VMware ESX environment.

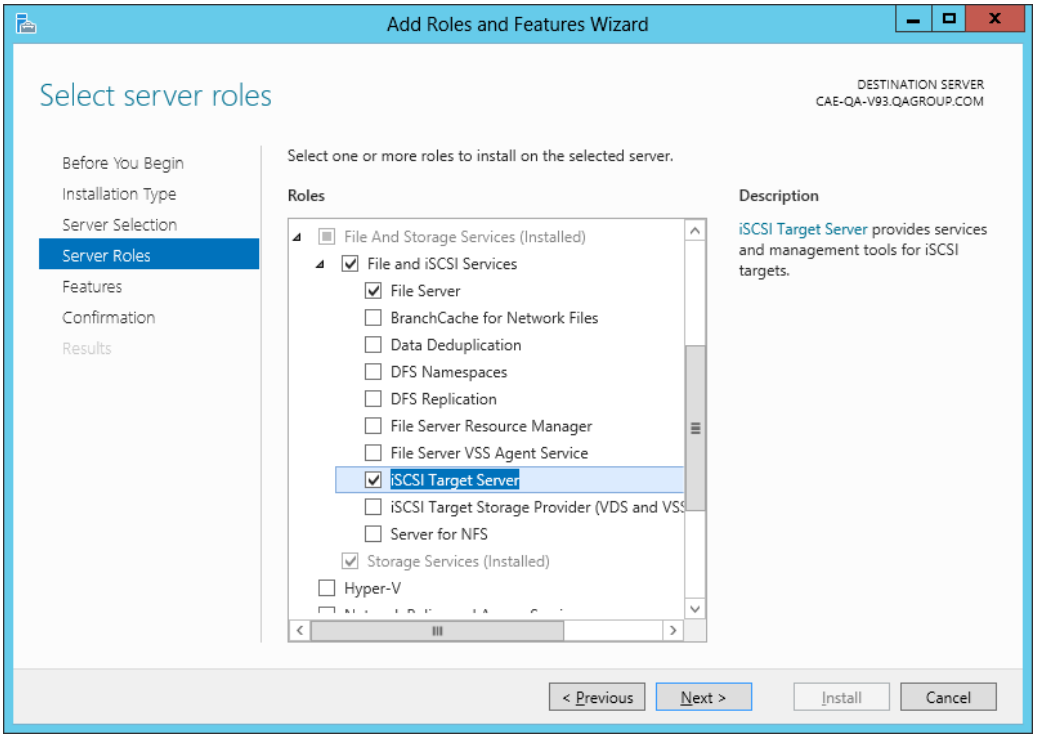


Installation of the iSCSI Target

1. From the **Server Manager** menu, select “**Add Roles and Features**” from the "**Manage**" drop-down.



2. Select the "**Role-based or feature-based installation**" option.
3. From the list of servers presented, select the appropriate server.
4. On the "**Select Server Roles**" screen under "**Server Roles**", navigate to and select "**File and iSCSI Services**" / "**iSCSI Target Server**". **Note**: "**File and iSCSI Services**" is in the tree hierarchy under "**File and Storage Services**" which is typically shaded and difficult to find.



6. Click "**Next**" twice to get to the "**Install**" button to be able to install the role.
7. The feature will install and the progress will be shown.

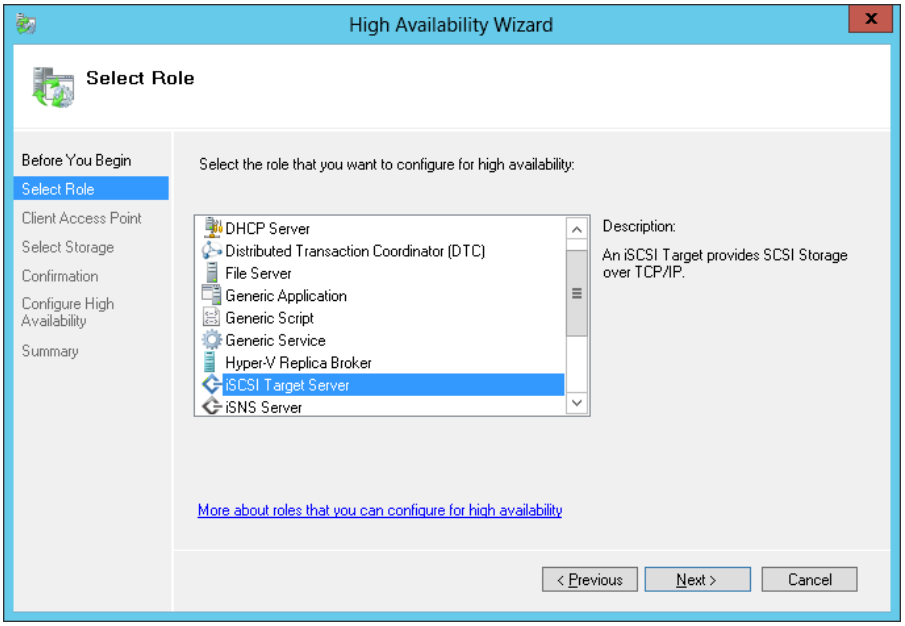
8. Upon completion, the message "**Installation succeeded**" will be displayed.
9. Repeat these steps for all servers in the cluster.

Creation of Mirror and Configuration of Cluster

1. Create your **DataKeeper volumes** and your **cluster**. See Creating a DataKeeper Volume Resource in WSFC for reference.

⚠	**IMPORTANT**: The iSCSI Target Role only supports DataKeeper Volumes that are mirrors of **Simple Volumes** placed on **Basic Disks**. If any of your mirrors are using volumes such as Striped or Spanned volumes on a Dynamic Disk on either the source or target system, then you cannot create an iSCSI Target role that uses those DataKeeper Volume resources for storage.

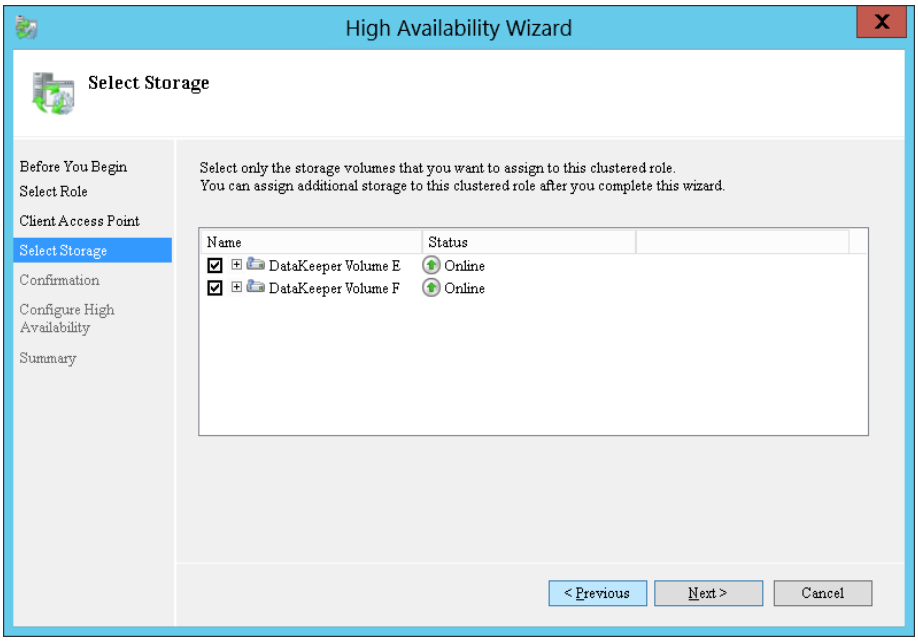
2. From the **Windows Failover Cluster Manager UI** (`cluadmin.msc`), select **Configure Role** and navigate to the screen to select the **iSCSI target role**.



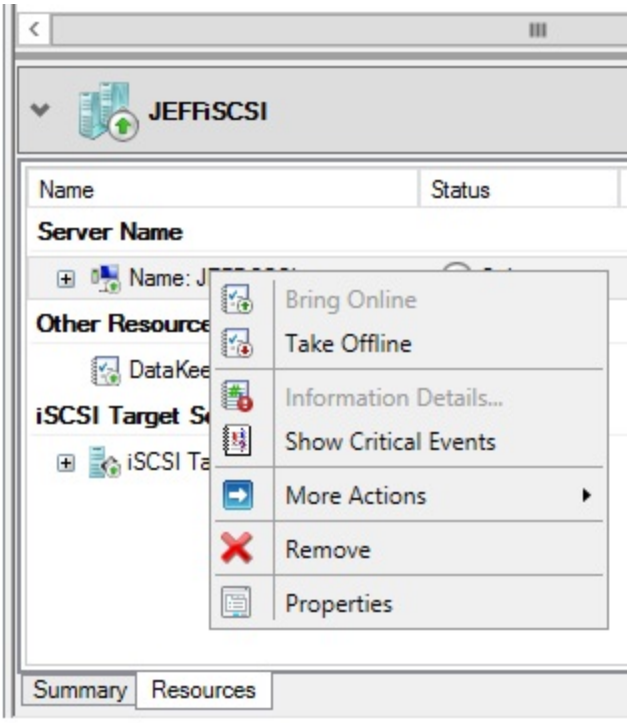
3. Select **iSCSI Target Server** role and select **Next**.
4. The **Client Access Point** page appears. Type the **Client Access Point name** and **IP address** for the iSCSI Target Server instance.

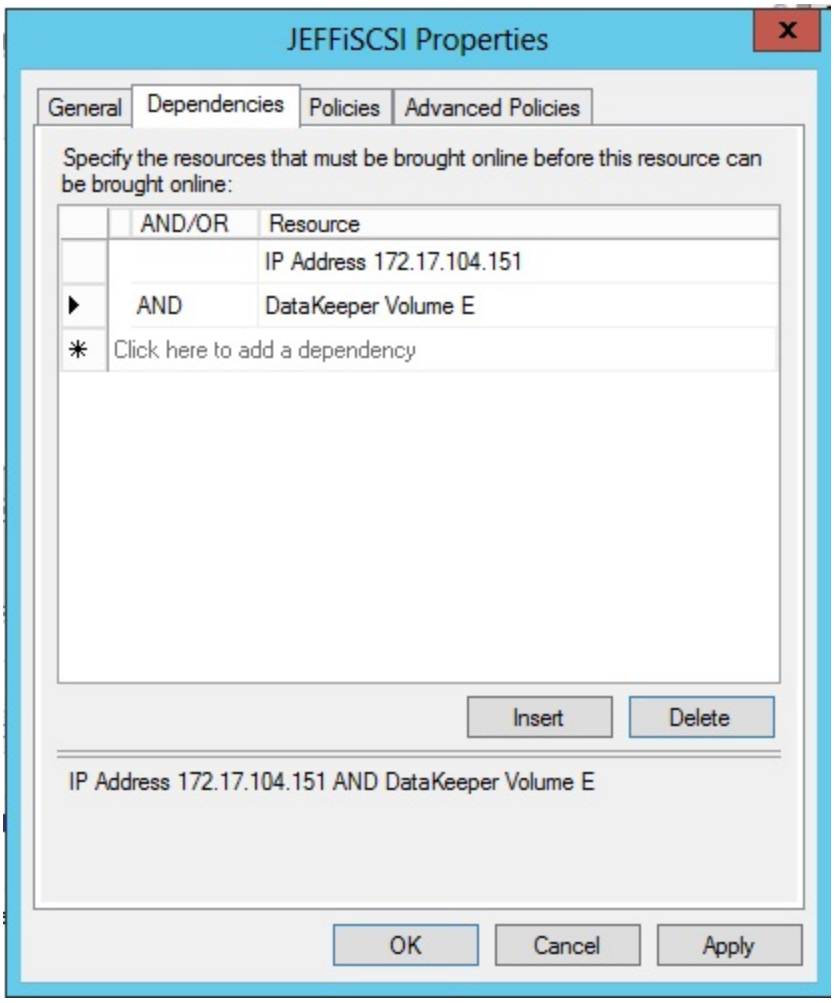
⚠	**IMPORTANT**: This name and IP address will be used later by clients to access the server address, so it should be recorded in DNS. This is very important for the servers to be able to resolve these names.

5. On the **Select Storage** dialog, select your **DataKeeper volume(s)**.



6. With the next set of screens, you should be able to complete the configuration.
7. Following setup, from the **Failover Cluster UI**, add dependencies for the DataKeeper volume(s).
    a. Click on **Roles** in the left pane, then click on the **iSCSI Target Server** resource in the top center pane.
    b. In the lower center pane, select the **Resources** tab, then right-click on the **Name: <client access point name>** under the **Server Name** heading and select **Properties**.
    c. Select the **Dependencies** tab and add the appropriate DataKeeper volume(s) as dependencies.


JEFFiSCSI
Name
Status
Server Name
Name: J
Other Resource
DataKee
iSCSI Target S
iSCSI Ta
Bring Online
Take Offline
Information Details...
Show Critical Events
More Actions
Remove
Properties
Summary
Resources

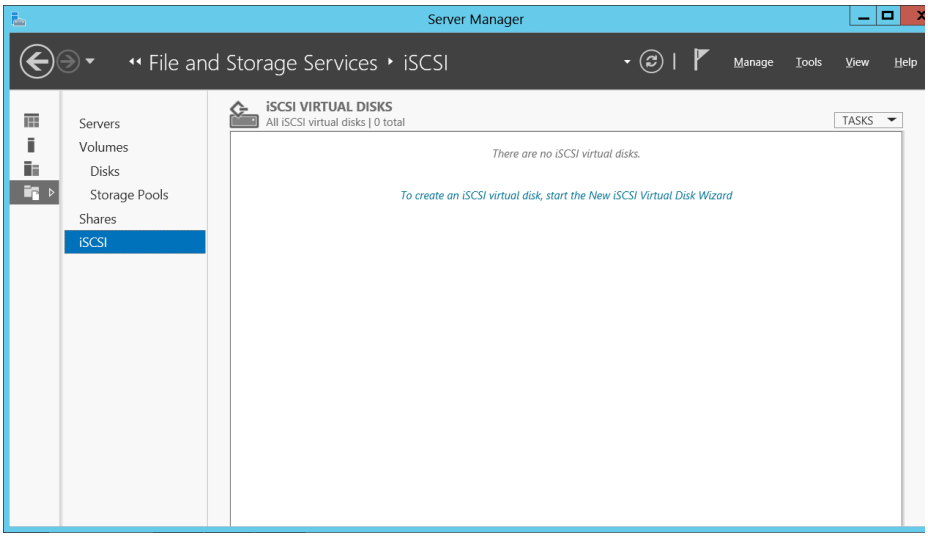


8. Setup is complete. Proceed to the **iSCSI Virtual Disks** configuration.

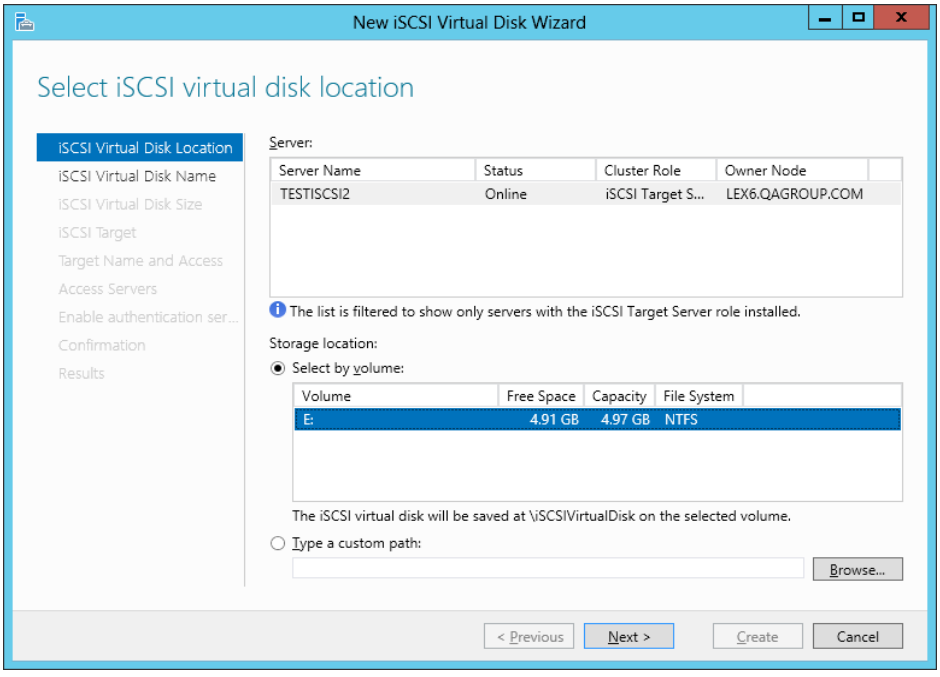
Creation of iSCSI Virtual Disks

Perform the following on the **primary server, wherever the iSCSI Target server is online at the moment**.

1. From **Server Manager**, navigate to **File and Storage Services** and select **iSCSI**. Click on the link "**To create an iSCSI virtual disk, start the *New iSCSI Virtual Disk Wizard***". (Alternatively, select **New iSCSI Virtual Disk** from the **TASKS** drop-down menu on the upper right of the screen.) **Note**: Windows Server 2012 **Server Manager** inherently takes some time to display or update the information presented to the user.



2. On the ***New iSCSI Virtual Disk Wizard***, you will see the server and the volume. Select the **DataKeeper volume** and click **Next**. (**Note**: The server name is the name created in the previous step and the volume is the DataKeeper volume exposed.)



3. Follow the next panel to configure the **iSCSI Virtual Disk**.

    a. Specify **iSCSI Virtual Disk Name**.

b. Specify **iSCSI Virtual Disk Size**. (**Note**: Multiple files can be created. If file size spans the entire disk, the OS may warn that disk is low since the VHD file(s) created can consume the entire disk.)

c. Designate whether the iSCSI Virtual Disk will be assigned to an **Existing iSCSI Target** or a **New iSCSI Target** on the **Assign iSCSI Target** screen. (See below for an explanation on when to select **Existing iSCSI Target**.)

d. Specify **iSCSI Target Name**.

e. On the **Access Servers** screen, select **Add**. Add the **iSCSI Initiators** that will be accessing this **iSCSI Virtual Disk**. **Note**: The iSCSI Initiators should be added one at a time.

4. Once all the answers have been provided, the iSCSI virtual disk/target creation is complete. Proceed to configuration of the iSCSI Initiator.

Setting Up Multiple Virtual Disks Within the Same Target Name

It is also possible to set up multiple iSCSI virtual disks within the same iSCSI target name. Whenever an iSCSI initiator connects to such a target, it will connect to all of the virtual disks that have been assigned to that name.

You need to have a plan ahead of time that describes which files you want to create and whether those files should all be accessed simultaneously or if the disks need to be accessed separate from one another.

Example Use Case

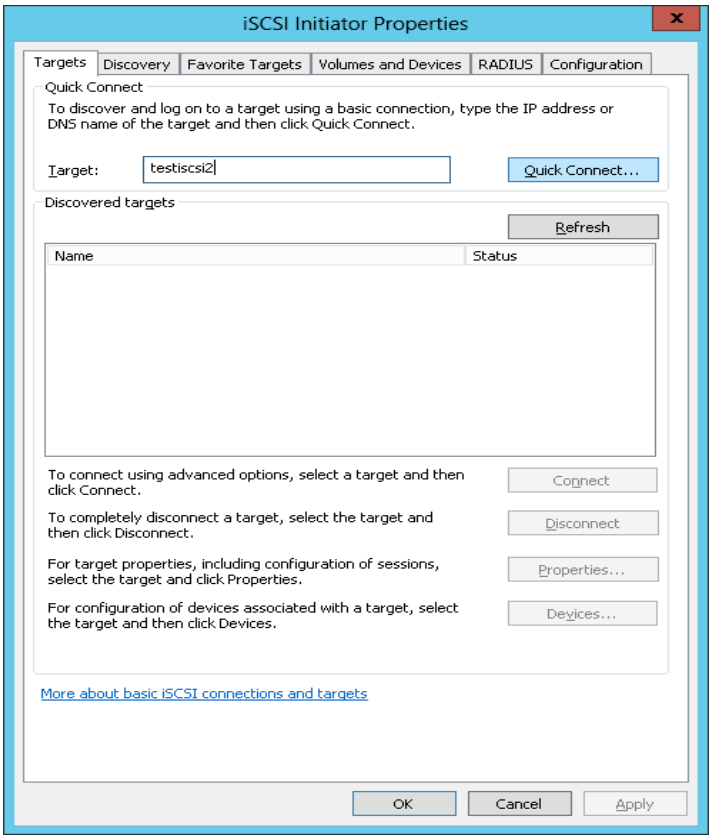
Set up two virtual disks that will be the System and Data disks of the same VM. Name the target "*server-1-disks*". When an iSCSI initiator connects to the "*server-1-disks*" target name, both disks get connected to that initiator system. Also, if setting up an iSCSI Target that has virtual disks that will be part of a cluster, and possibly CSVs, then all of these disks can be in the same target.

To set up multiple virtual disks within the same target name, on Step 3c, instead of selecting **New iSCSI Target** on the **Assign iSCSI Target** screen, select **Existing iSCSI Target** and specify the iSCSI target name that was created previously. This target name will appear in the list of "targets" when an iSCSI Initiator connects to the iSCSI Target Server. If a target has more than one virtual disk associated with it, then the initiator will get a connection to each of those disks (they will appear as a new Disk in Disk Management).

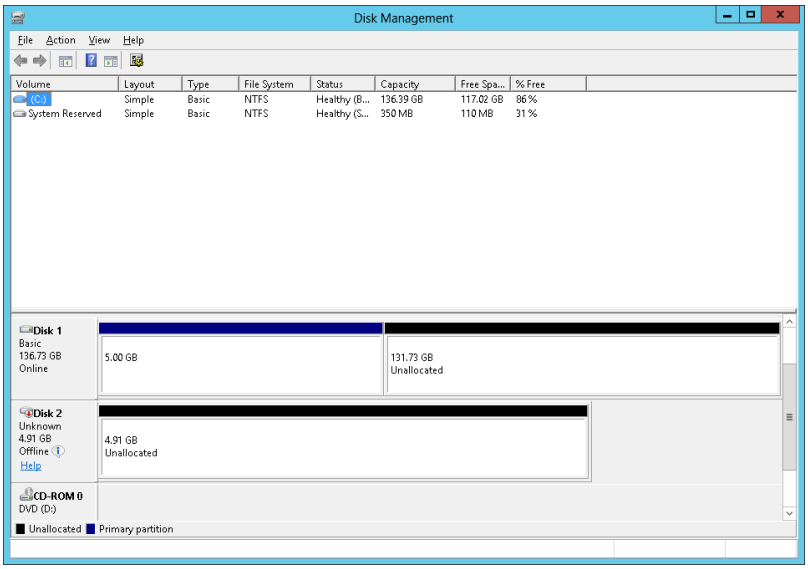
Setup of iSCSI Initiator on Windows 2012

Once the virtual disks/targets are created, each of the cluster servers must initiate a connection to them via Microsoft's iSCSI Initiator.

1. From "**Administrator Tools**" in "**Server Manager**", start "**iSCSI Initiator**".
2. Select the "**Targets**" tab and enter the **Network Name** or **IP address** of the **clustered iSCSI Target** created in the previous step. Select "**Quick Connect**".



3. New panel should indicate that "**Login has succeeded**". Click **OK** to hide the panel.
4. Start "**Disk Manager**". The new iSCSI virtual disk will be displayed and can be initialized.



5. Right-click on the disk(s) to bring it online.
6. Initialize the disk(s).
7. Create the new volume and assign the drive letter.
8. Configuration is now complete.

DataKeeper Target Snapshot

Overview

DataKeeper's target snapshot feature, integrated with both DataKeeper and DataKeeper Cluster Edition, is the process of creating point in time copies of replicated volumes allowing access to data on a standby cluster node without impacting operations such as switchovers and failovers. Data protection is not lost for any period of time. Enabling target snapshot allows data to be used on an otherwise idle target node without negatively impacting the performance of the source.

Without target snapshot, DataKeeper and DataKeeper Cluster Edition are able to maintain a real-time replica of their source system's data on the target system. However, this replica cannot be accessed without pausing the mirror and unlocking the target system. Mirror failover and switchover cannot occur while in this paused and unlocked state, making the protected application less highly available. Application-consistent target snapshot allows access to data on the target system while maintaining high availability of the running application on the source system. The mirror remains in the **mirroring** state and continues to update the target volume with all writes from the source. Target snapshot integrates with Volume Shadow Copy Service (VSS) to ensure that the data which is exposed on the target system is in an application-consistent state.

How Target Snapshot Works

DataKeeper target snapshot uses a copy-on-write strategy to maintain and expose a view of the volume at a particular point in time. A snapshot file is used to store the volume information. Configuring the location of this snapshot file is the first step toward enabling target snapshot.

When the EMCMD command, TAKESNAPSHOT, is run, DataKeeper will create and mount a snapshot file in the configured snapshot folder. A request is then sent to the source system telling it to use VSS to quiesce any VSS writers on the given volume and notify the target when all write operations to the disk are stopped and the volumes are in a well-defined state.

Quiescing the Database/Application

The application-consistent capabilities of this feature integrate with Volume Shadow Copy Service (VSS) to ensure that the data that is exposed on the target system is in an application-consistent state. Once a snapshot is requested, the VSS service pauses the systems and ensures that all applications modifying data on disk bring all their files into a consistent state prior to the creation of the snapshot. This is called quiescing the database/application. Rather than shutting down the database and reopening it in restricted mode, quiescing temporarily freezes application write I/O requests (read I/O requests are still possible) for the short time required to create the snapshot. Once in the quiesced state, the snapshot on each volume is initiated by adding the snapshot message to the driver mirror write queue(s). VSS will then unfreeze the applications and the volume is unlocked, thus minimizing the amount of time the apps are quiesced. The user can now perform actions on the target system while the mirror remains in the **Mirroring** state and the application on the source system remains highly available.

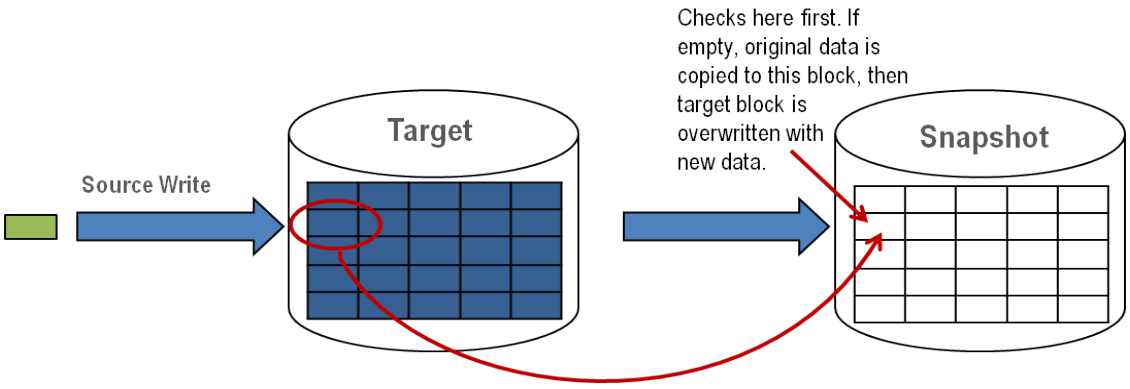
Read and Write I/O Requests

The snapshot exists in parallel with the live copy of the volume to be backed up, so except for the brief period of the snapshot's preparation and creation, an application can continue its work. Writes to the target, however, will now be processed differently while the target is in this state.

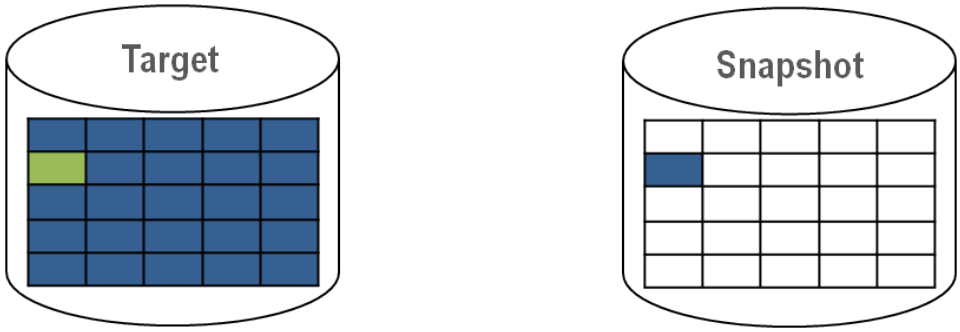
Data mirroring from the source system will continue uninterrupted, but any new data from the source that is received after the snapshot is taken will not be visible on the target system until the snapshot is dropped. This allows an application on the target system to run, using (and updating) data that represents the source system's data at the point in time that the snapshot was taken.

Source Write

In order to accomplish source writes, when new data comes from the source, DataKeeper first determines if that particular block of data has already been written to the snapshot file.



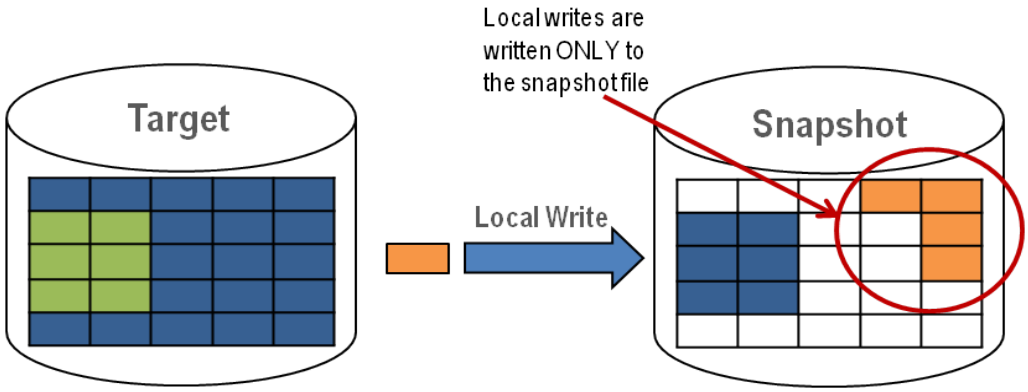
If the block has not been written to, as shown above, that **original** block is written to the snapshot file in order to preserve the snapshot data, then the new data is written to the target. The result is shown below.



If DataKeeper determines that this block has already been written to the snapshot file, then this step is skipped and the block is just written to the target. For blocks on the source volume that are overwritten frequently, the snapshot file only has to be updated once, the first time that block is written after the snapshot is taken.

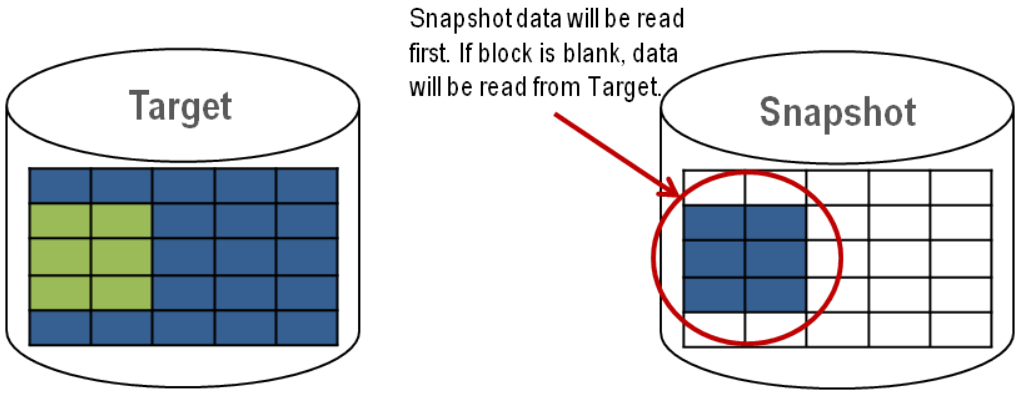
Local Write

If local writes are performed on the target (from applications on the target system), these writes are stored in the snapshot file and do not overwrite any blocks on the replicated volume itself. (**Note**: Any local writes stored in the snapshot file will be lost when snapshot is dropped.)



Target Read Request

Read requests on the target volume will return snapshot data. This is accomplished by first reading data from the blocks written in the snapshot file. Any blocks that have not been saved to the snapshot file will be read from the target volume.



Using Target Snapshot

There are three tasks that must be performed when using target snapshot. The snapshot location must be configured, the snapshot must be initiated, then once target reporting actions are complete, the snapshot must be dropped.

Configuring the Snapshot Location

When target snapshot is initiated, DataKeeper creates and mounts a file in the snapshot location to hold the snapshot data. This location must be configured prior to initiating a snapshot. See Files / Disk Devices / Registry Entries below for more information about the mounted snapshot disk(s).

	**IMPORTANT**: The maximum size of a volume to be snapshotted is 2 TB.

When configuring the snapshot location, make sure it meets the following criteria:

- Is only used when a snapshot is requested.
- Cannot be stored on a DataKeeper mirrored volume.
- Can store multiple snapshot files for different volumes.
- Must have enough free space to create and accommodate a file that will grow depending on the source mirrored volume size and writes during snapshot use.

  Note: Do not change the snapshot location during a snapshot.

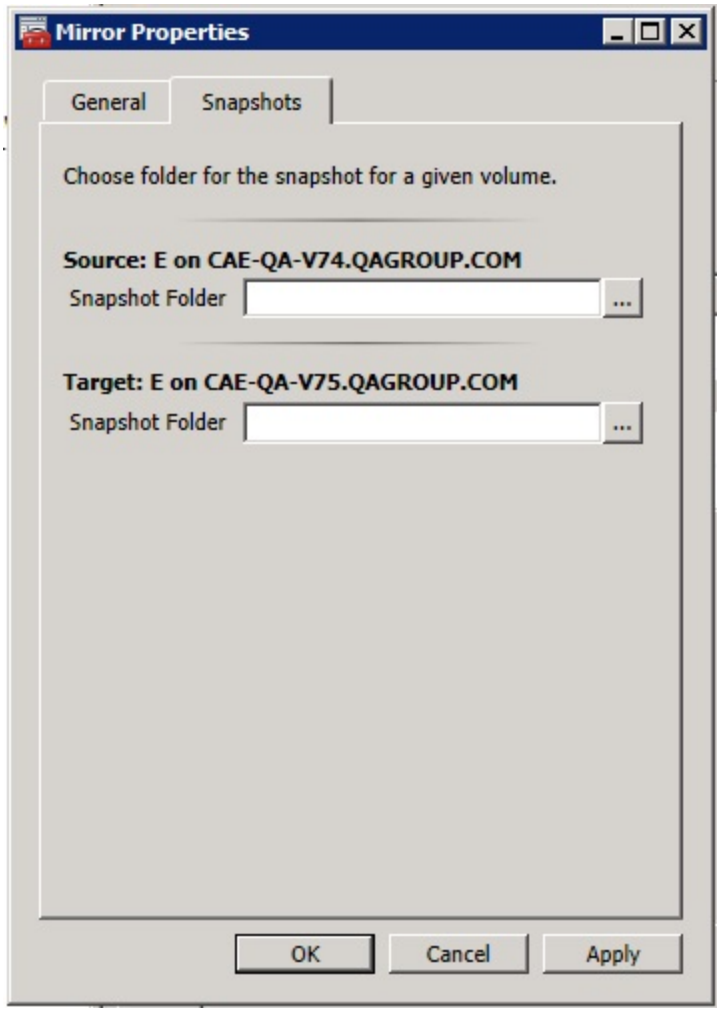
Snapshot Location Size

The size of the snapshot location should be determined on an individual basis based on several criteria. In practice, the size required for the snapshot file will be far less than the size of the volume being snapshotted. The storage required needs to be big enough to contain any data that changes on the source system while the snapshot is being used. All snapshot files will be zeroed out each time a snapshot is initiated and will incrementally grow in size during use. The files will be deleted when the snapshot is dropped. Given that the copy on write process only writes "changed" blocks to the snapshot file, consideration should be given to the duration of the snapshot as well as the rate of change in the volume being mirrored. Once an historical view can be established of snapshots from past activity, the size can be re-evaluated.

	**BEST PRACTICE**: Be conservative in your estimate, assuring that there is excess space available. If enough space is not allocated and the limit is reached, your snapshot will be dropped.

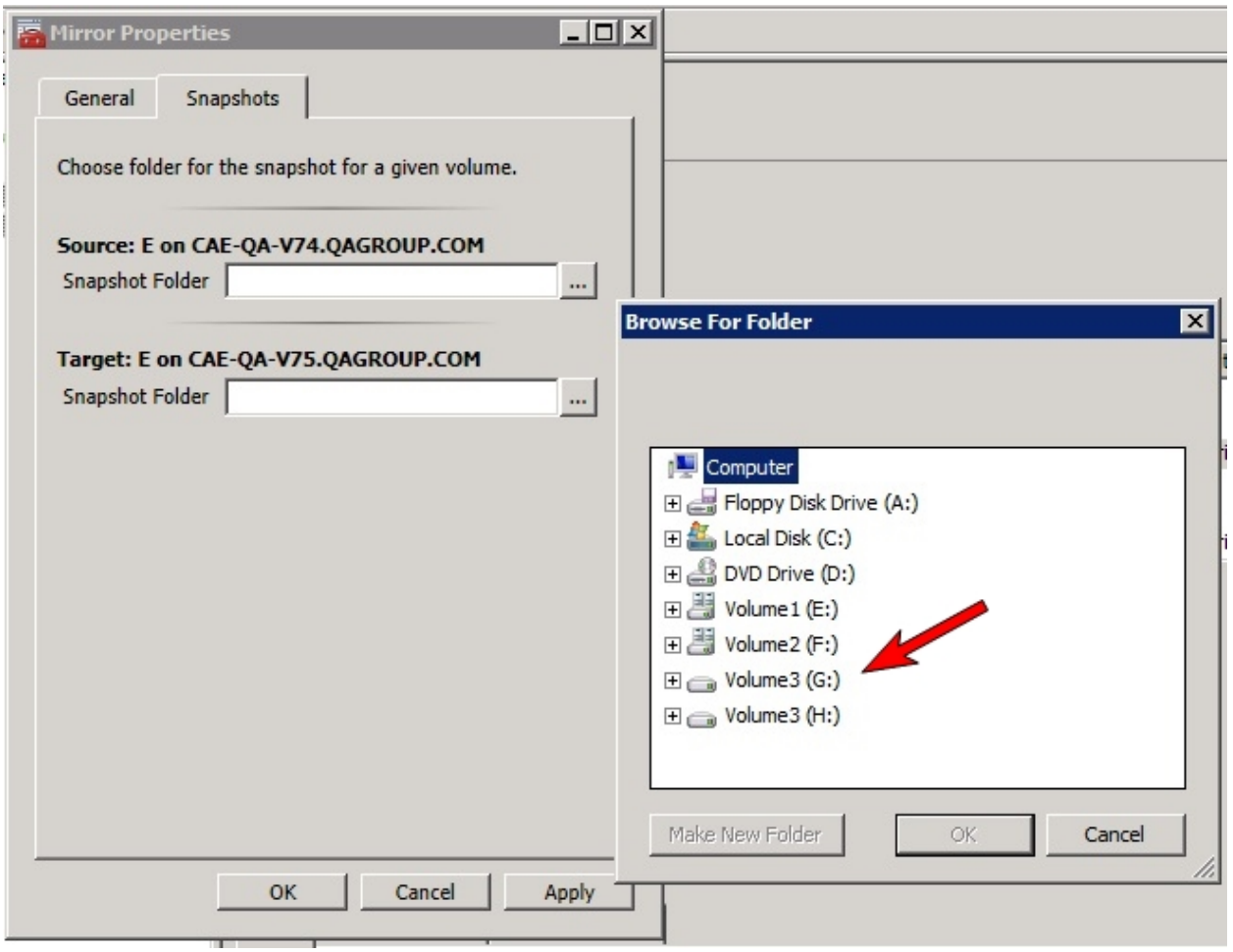
Snapshot Location Selection

1. Right-click on the appropriate mirror and select **Mirror Properties**.
2. From the **Mirror Properties** dialog, select the **Snapshots** tab.



	NOTE: DataKeeper will use the snapshot location configured on the target node; however, since either node in the mirror can become target, the snapshot location may be configured on both the source and the target.

3. Use the **browse** ... button to choose the location for the snapshot or type the **path** into the text box.



When clicking the **browse** button that corresponds to the system where the GUI is running, the **Browse for Folder** dialog will appear. When clicking the **browse** button that corresponds to a system that is not the system where the GUI is running, the **Browse for Folder On Remote** dialog will appear.

4. Select your **snapshot location** for the source and the target. Make sure this volume has sufficient free space in order for the operation to complete successfully. Refer back to Snapshot Location Size for further details when estimating the volume size for your snapshot. Click **Apply**.

   Note: Each volume on a given system can either use the same location or a different location can be selected.

	**In order to Bypass the GUI**, the location of the snapshot file can be set via command line using the SETSNAPSHOTLOCATION command. In order to view the current snapshot location of a given volume, use the GETSNAPSHOTLOCATION command.

Taking a Snapshot

Once a **snapshot location** has been configured on the target system, a snapshot can be taken. From the target node, run the EMCMD command TAKESNAPSHOT.

Dropping a Snapshot

When the snapshot is no longer needed, volume snapshots must be dropped in order to return to normal processing. Run the EMCMD command DROPSNAPSHOT which will lock the volume and clean up the snapshot files that were created. The volume will then return to a normal target where writes from the source will go directly to the volume with no copy-on-write storage.

Disabling Target Snapshot for a Given Volume

To disable target snapshot for a given volume, the snapshot location must be cleared. This can be accomplished via the GUI.

1. Right-click on the appropriate mirror and select **Mirror Properties**.
2. From the **Mirror Properties** dialog, select the **Snapshots** tab.
3. Remove the snapshot folder of the volume you would like target snapshot disabled on.
4. Click **Apply**.

	The snapshot file location can also be cleared via command line by executing the CLEARSNAPSHOTLOCATION command.

Once successfully executed, a snapshot location will have to be reconfigured in order to initiate another snapshot of that volume.

Target Snapshot Notes

Supported Configurations

DataKeeper target snapshot is currently supported in non-shared (1x1 and 1x1x1) environments using Windows 2008 R2 and Windows Server 2012.

Source Out of Service

DataKeeper target snapshot cannot be initiated when the source is out of service. However, if the source is taken out of service after snapshot is initiated, the snapshot will continue to work as expected. You can continue to use the snapshot, and drop it when you are done, while the source is out of service.

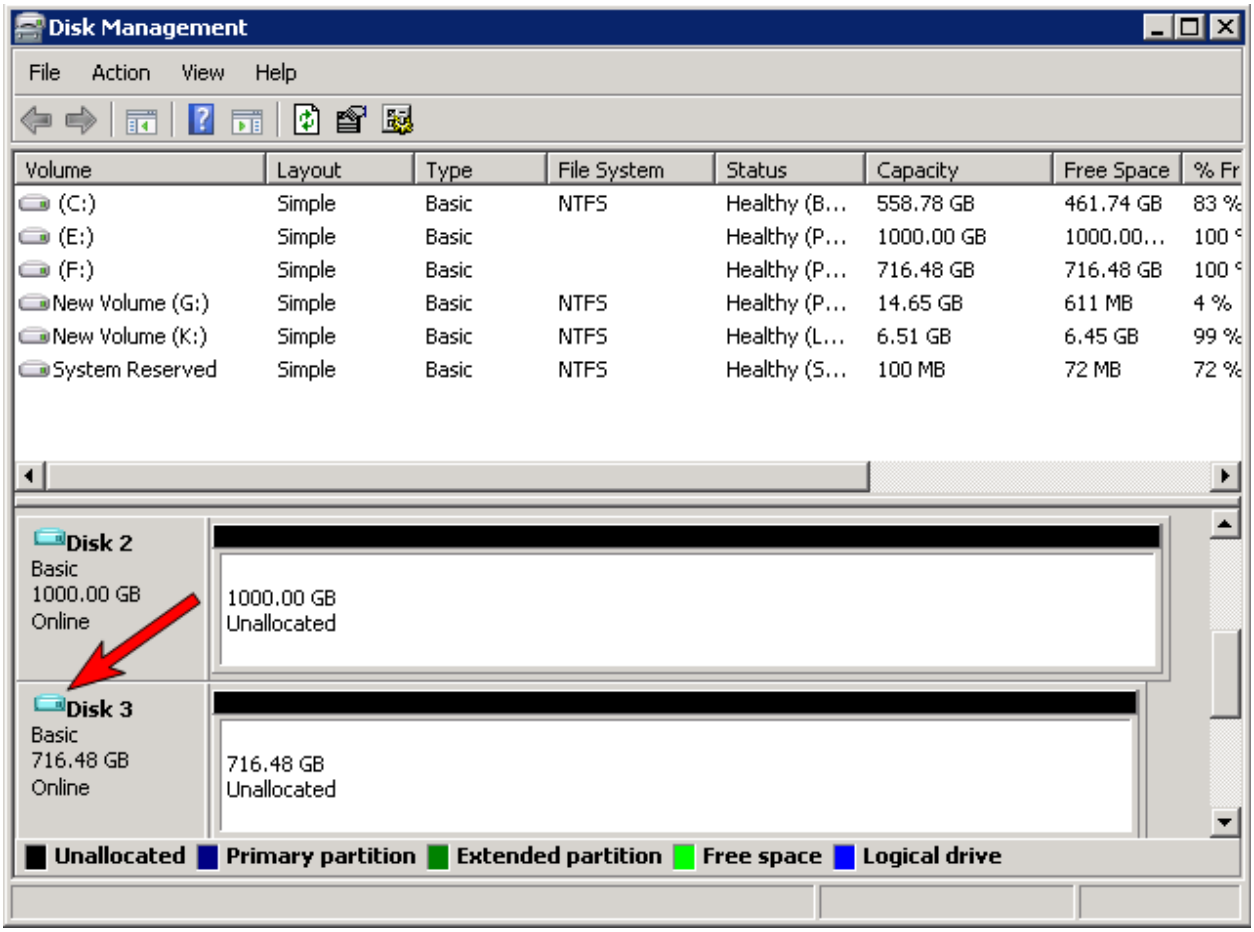
Switchovers and Failovers

If a snapshot is in progress, the volume being snapshotted cannot become the mirror source until the snapshot has been dropped. You must perform a DROPSNAPSHOT in order to allow a switchover or failover of the volume to the local node. Any processes that access data on the snapshotted volume will have their handles invalidated when the snapshot is dropped. However, if the volume is subsequently unlocked, you must make sure that those processes do not re-open their handles. At this point the data will be "**live**" application data and not the snapshotted data.

Files / Disk Devices / Registry Entries

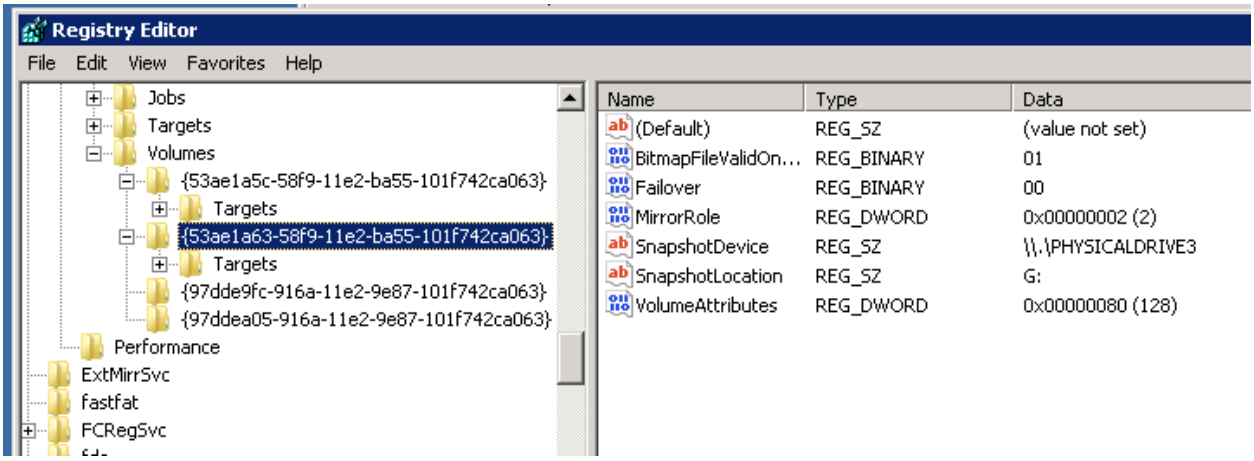
When a snapshot is taken, a snapshot file is created for each snapshotted volume in that volume's snapshot location. The name of the file that is created is `datakeeper_snapshot_vol<X>.vhd`, where `<X>` is the drive letter. This VHD file gets attached as a virtual disk device which can be seen in Windows Disk Management.

NOTE: The colored icon next to the disk number represents this disk as a VHD.



⚠	**CAUTION**: The virtual disk devices that are created will appear as unpartitioned Basic disks. They should be used for **snapshot data only** and should not be detached or partitioned while snapshots are in progress. Doing so may result in corruption of the snapshot data. **Make sure that they are not mistaken for available disks to be partitioned and formatted.**

Once these virtual disk devices are attached, a registry entry named `SnapshotDevice` is created in the volume's key. The value is set to `\\.\PHYSICALDRIVE<x>` where `<x>` is the disk number, as shown below:



`TargetSnapshotBlocksize` Registry Value

DataKeeper target snapshot uses a default block size of 64KB for all entries that are written to the snapshot file. This block size can be modified by creating a `REG_DWORD` value named TargetSnapshotBlocksize in the Volume registry key.

The value should always be set to a multiple of the disk sector size, which is usually 512 bytes. Certain workloads and write patterns can benefit from changing the block size. For example, a volume that is written in a sequential stream of data (*e.g. SQL Server log files*) can benefit from a larger block size. A large block size results in fewer reads from the target volume when consecutive blocks are written. But a volume that is written in a random pattern may benefit from a smaller value or the default 64KB. A smaller block size will result in less snapshot file usage for random write requests.

SQL Server Notes

If you are using DataKeeper target snapshot with SQL Server in a SteelEye Protection Suite environment, it is recommended that you use a separate SQL Server instance to attach database(s) to the snapshot.

For a clustered SQL Server environment, you must use a separate SQL Server instance to attach database(s) to the snapshot.

Known Issues

Microsoft .NET Framework 3.5 SP1 Requirement

The target snapshot feature requires Microsoft .NET Framework 3.5 SP1 to be installed - download from: http://www.microsoft.com/net.

NTFS File System Message

If an internal snapshot error occurs after target snapshot is initiated (such as the snapshot file running out of space or being detached by the user), snapshot will be disabled, the volume will be locked and snapshot files

for any failed volumes will be deleted. While the snapshot error is being handled, you may receive NTFS file system errors. These messages are normal and can be ignored.

Application Data Using Snapshot

When using target snapshot data with your application, if the target snapshot is refreshed, you may need to close and reopen your application(s) to refresh the data.

Volume Shadow Copy Service (VSS) Free Disk Space Requirements

If your target snapshot volume has insufficient disk space, VSS operations involving that volume may fail with an "unexpected error". To avoid this, your snapshot volume should follow the guidelines from the Microsoft article Troubleshoot VSS issues that occur with Windows Server Backup (WBADMIN) in Windows Server 2008 and Windows Server 2008 R2

This article provides the following requirements for free disk space:

For volumes less than 500 megabytes, the minimum is 50 megabytes of free space. For volumes more than 500 megabytes, the minimum is 320 megabytes of free space. If the volume size is more than 1 gigabyte, a minimum of at least 1 gigabyte of free disk space on each volume is recommended.

Using SteelEye DataKeeper Standard Edition To Provide Disaster Recovery For Hyper-V Virtual Machines

Considerations

When preparing a Hyper-V environment that spans subnets, subnetting may need to be taken into consideration for any applications that are running inside the virtual machine. Some applications "hard code" IP addresses into their configurations. When these types of applications are loaded in a virtual machine that is replicated (via a DataKeeper replicated volume) to a target server on a different subnet, they may not operate as expected due to the difference in the network settings.

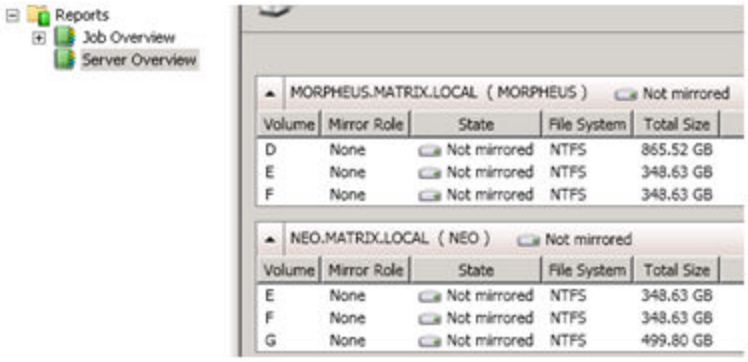
Preparing the Environment

1. Install Windows on two servers with at least two partitions, one for the OS and one for the Hyper-V virtual machine (VM) files. The partition for the files on the target server must be of equal or greater size to the source server's "data" partition. Install and configure the Hyper-V role on each server as described in Microsoft's *Hyper-V Planning and Deployment Guide* and the *Hyper-V Getting Started Guide*, but wait to create your virtual machine until the DataKeeper replicated volume has been created.
2. Complete the installation requirements for the SteelEye DataKeeper software.
3. Connect to the Servers.

   Once you connect, new options will appear in the center pane.

   You can also optionally review the**Server Overview** report to see the status of your volumes.

When you connect to multiple servers that have DataKeeper installed and licensed, you will see multiple servers and volumes listed here.

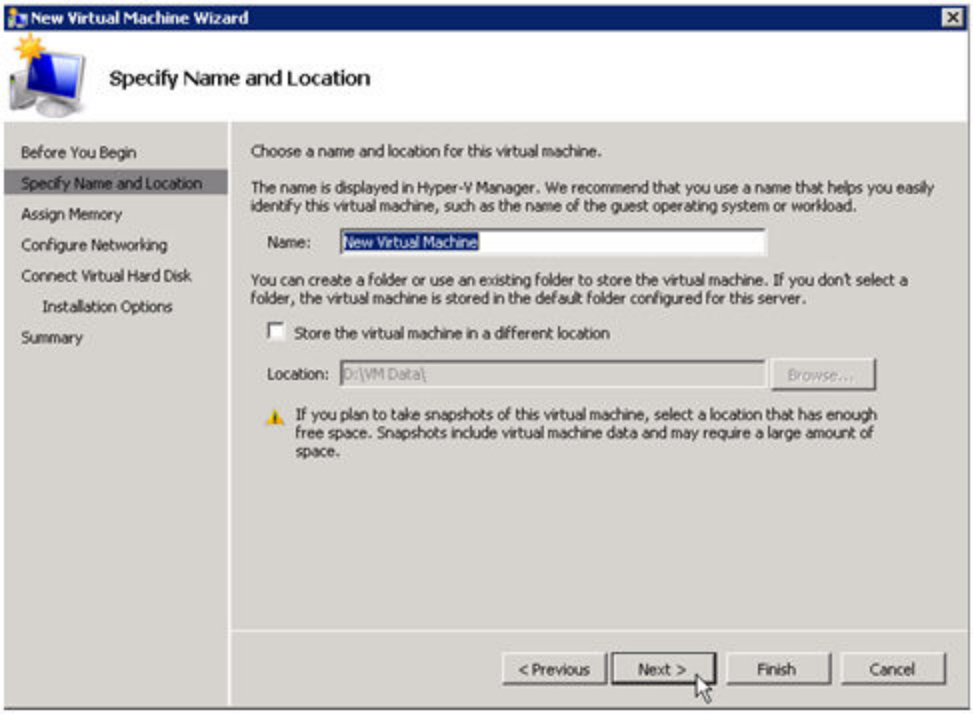


4. Create a Job / Mirrored Volume.

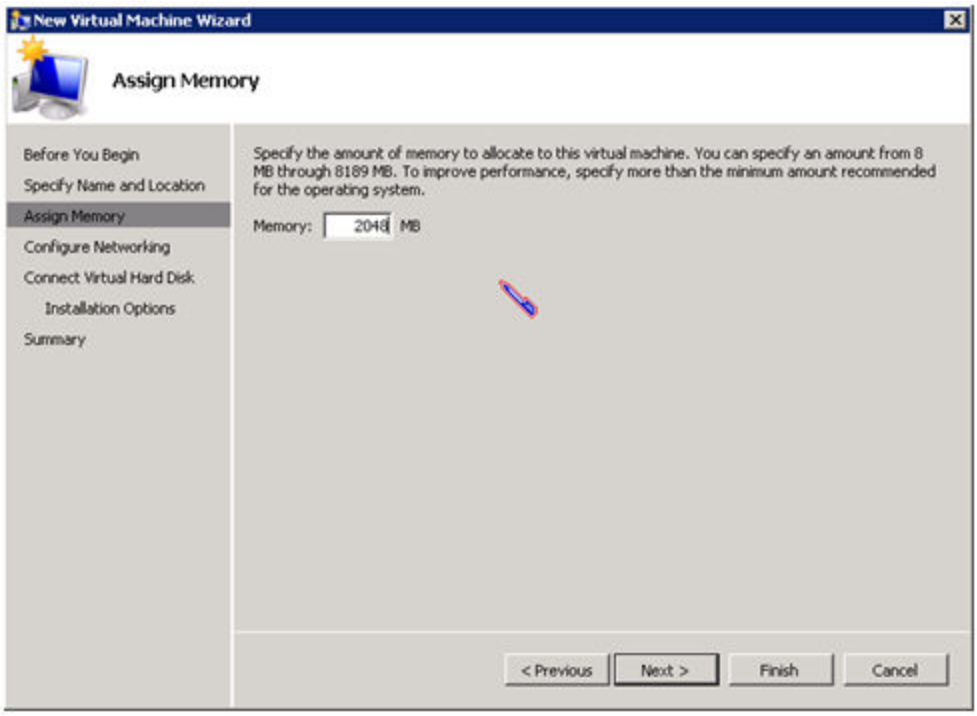
   Note: When you select your source server, ensure you select the server whose volume you want to replicate from. Reversing the source and target in these steps will completely overwrite your source volume with whatever is on the target server's volume, even if it is empty, causing you to lose any and all data stored on the source volume.

Create and Configure a Hyper-V Virtual Machine

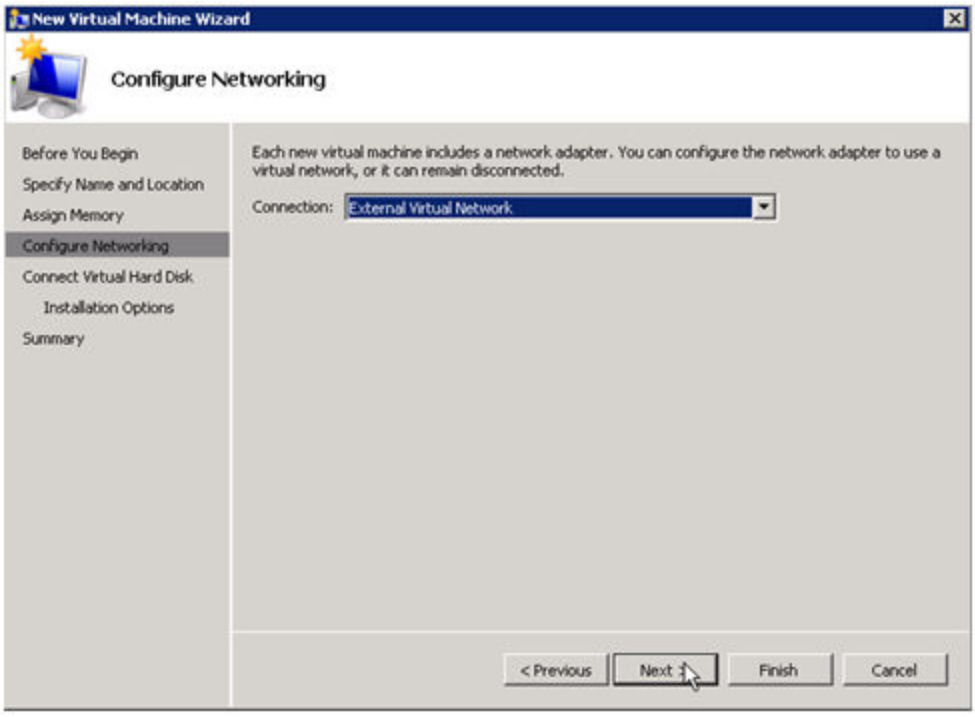
1. Launch the **Hyper-V Console** from **Start** - **Administrative Tools** - **Hyper-V Manager**.
2. Start the **New Virtual Machine Wizard**.



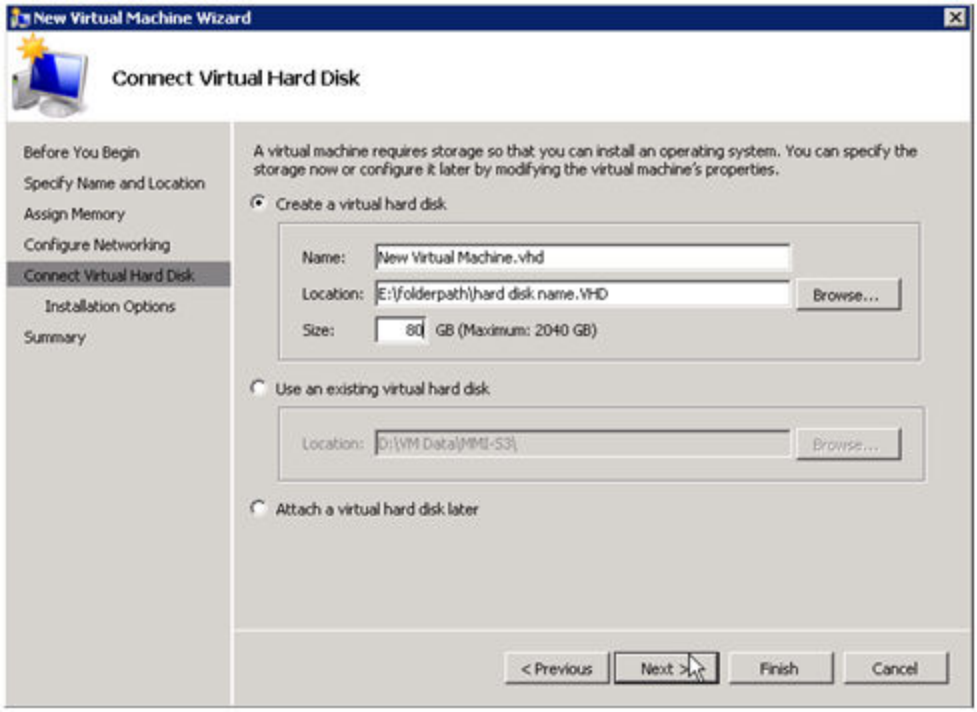
3. Specify the amount of **RAM** to use.



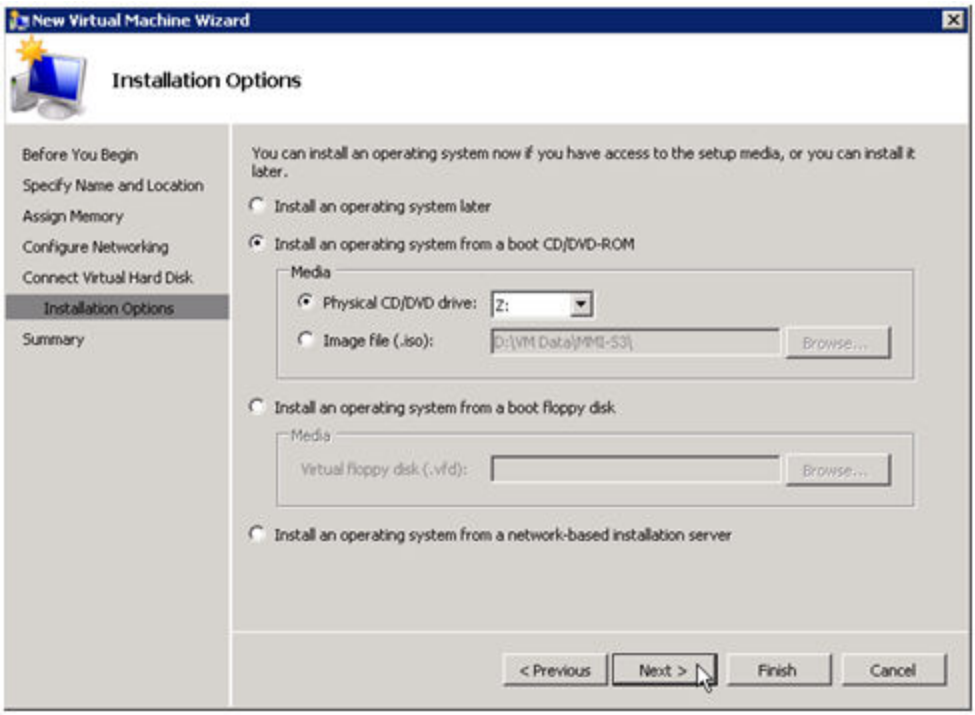
4. Select a **network adapter** to use.



5. Create a new **Virtual Hard Disk** on the replicated volume (or copy an existing VHD onto the replicated source volume and point the creation wizard at it to use as the virtual disk).



6. Specify the **operating system installation options**.



7. **Finish** the wizard and start the **virtual machine**.

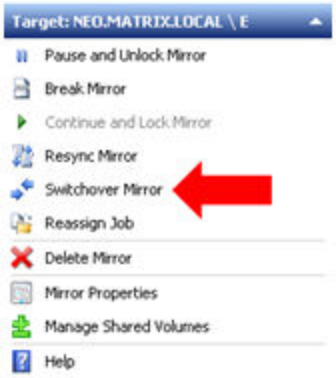
Install an Operating System and Any Required Applications in the Virtual Machine

1. Load the operating system into the virtual machine as dictated by industry or vendor specified best practices.
2. Configure the networking within the virtual machine to use DHCP addresses. Use DHCP reservations and name resolution (DNS or WINS) records as well if necessary for address consistency for client connections.
3. Install any necessary applications in the virtual machine.

Configure the Target Server to Run the Virtual Machine

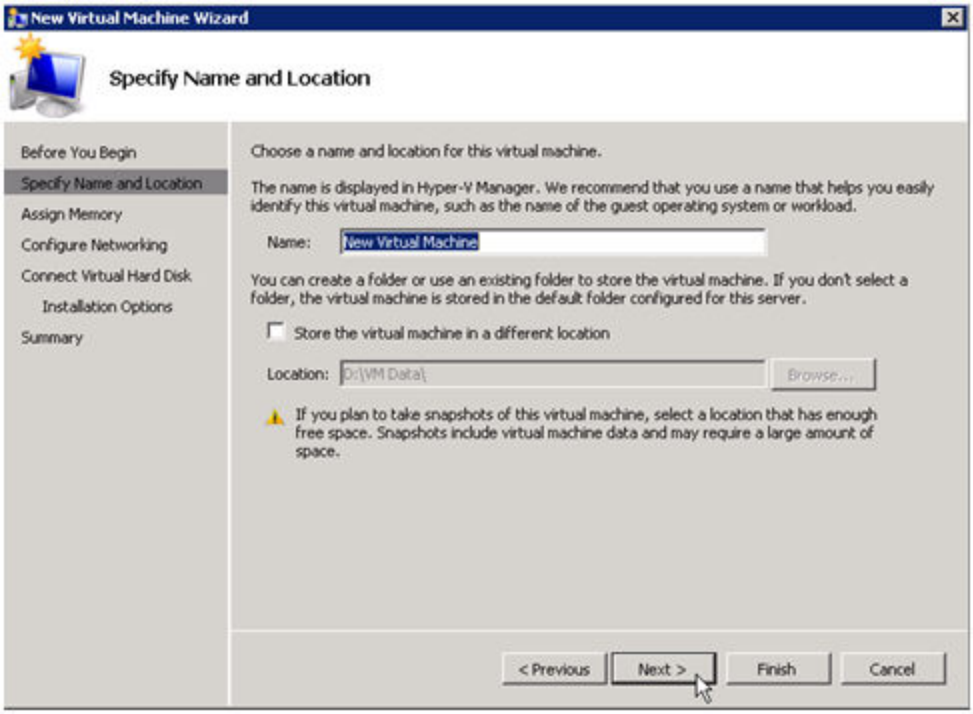
1. On the source Hyper-V host server, open **Hyper-V Manager**, connect to the virtual machine and do a full shutdown of the virtual machine. These actions will quiesce the data on the disk and will maintain data integrity on the target server.
2. Start the **DataKeeper console** as described previously.
3. Ensure the volume has been fully mirrored by checking the mirror status. The status must indicate **Mirroring** with the **zero KB Resync Remaining**.

4. Select the mirror and click **Switchover** in the **Actions** pane.

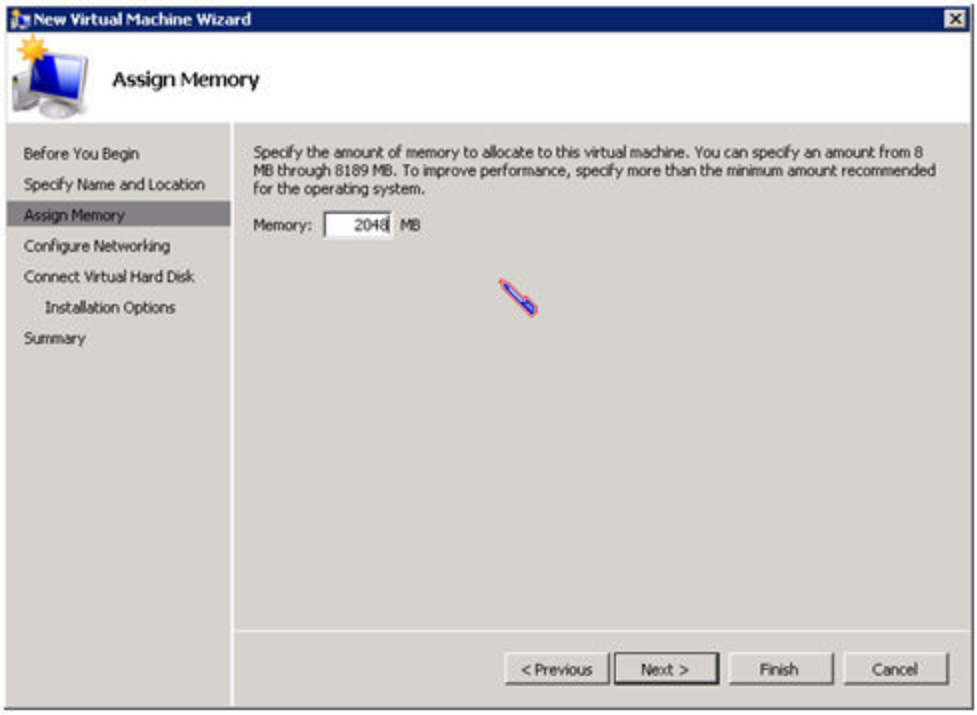


This will reverse the source and target and allow you to provision the virtual machine on the target server.

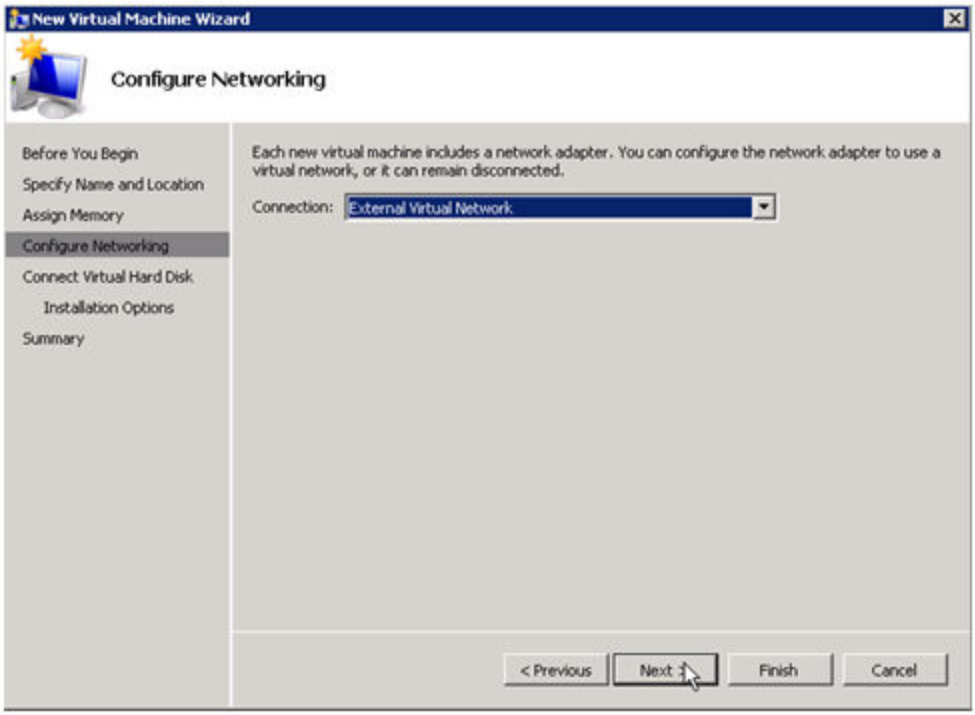
5. On the target server, start the **Hyper-V Manager**.
6. Start the **New Virtual Machine Wizard**.



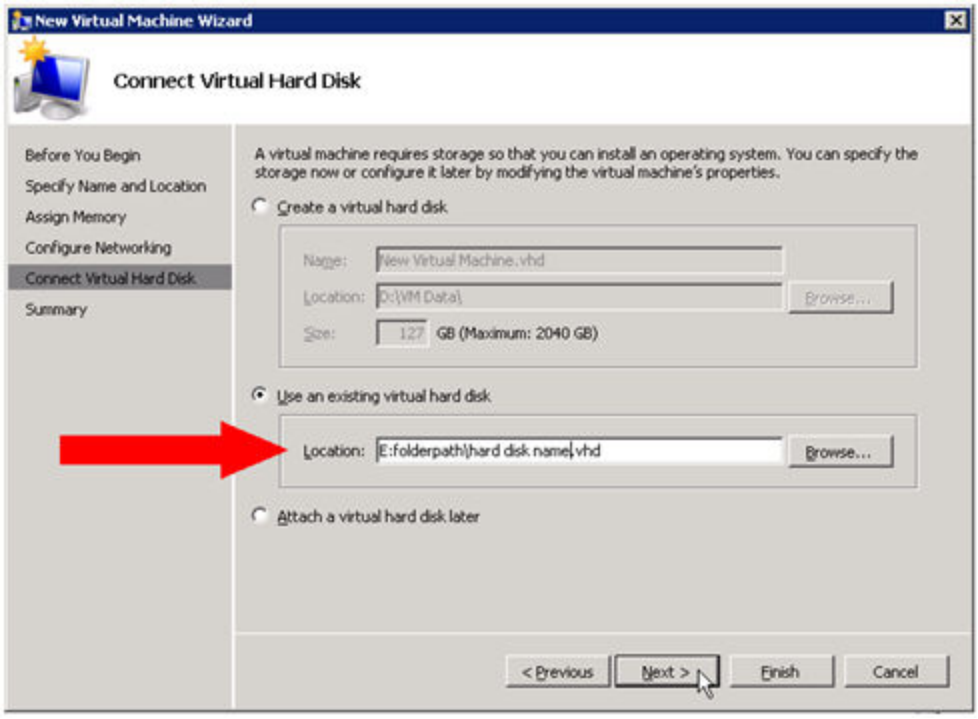
7. Specify the amount of **RAM** to use.



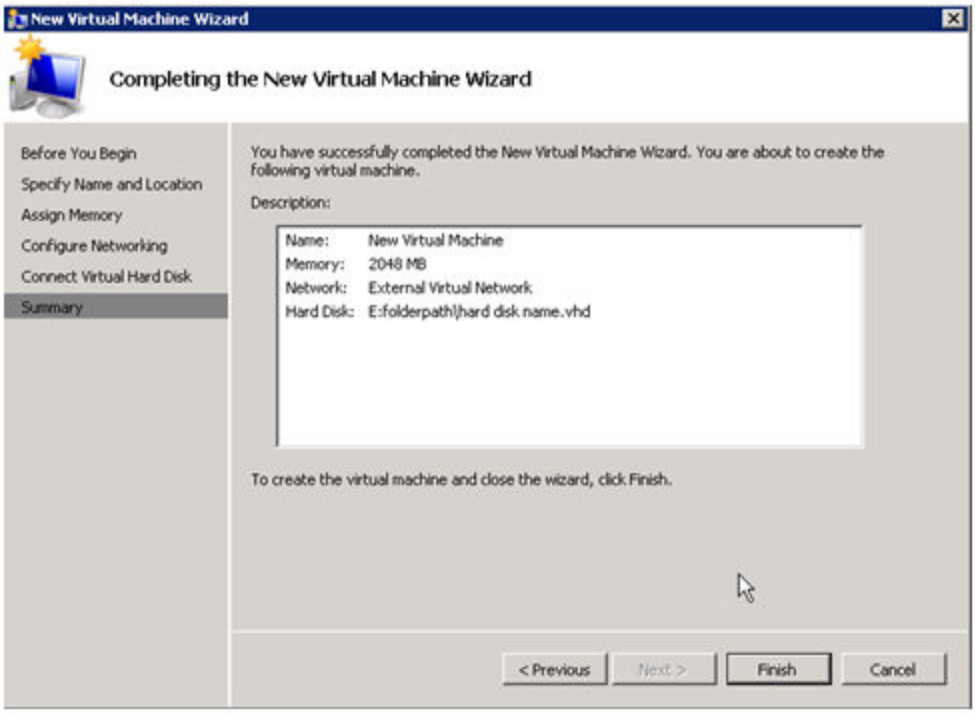
8. Select a **network adapter** to use.



IMPORTANT: Use the existing virtual hard disk on the replicated volume.



9. Click **Finish** to finalize the virtual machine creation process.



Start your virtual machine and test it to make sure it operates as expected.

Planned/Unplanned Switchover

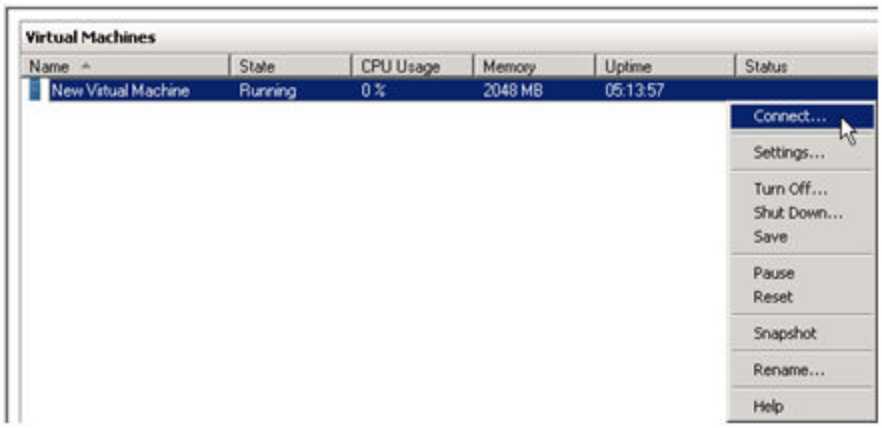
Initiate a **Planned Switchover** to migrate the virtual machine back to your source server.

Initiating a switchover for testing or in the event of an actual outage on the primary server can be completed simply by doing a **Planned Switchover**. There are two types of switchovers, **planned** and **unplanned**.

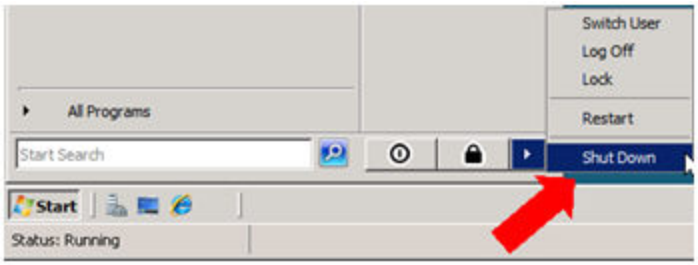
Planned Switchover

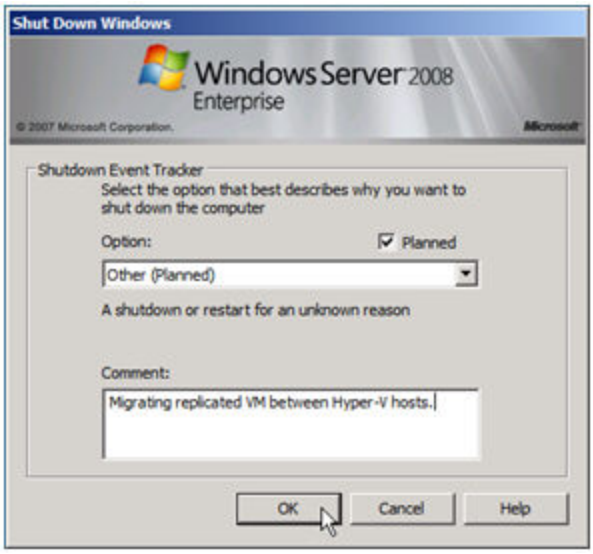
A planned switchover is typically done in a maintenance window when the user community can be advised of planned downtime.

1. On the server on which the virtual machine is running, start **Hyper-V Manager**, as previously described, and connect to the **virtual machine**.



2. From inside the virtual machine, **Shut Down** the virtual machine.

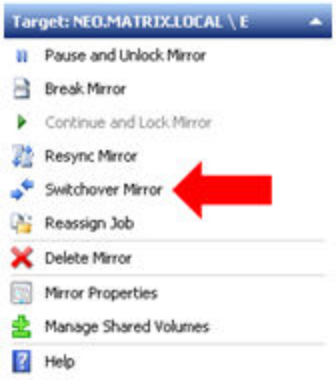




3. On the same server, start the **DataKeeper console** as described previously.

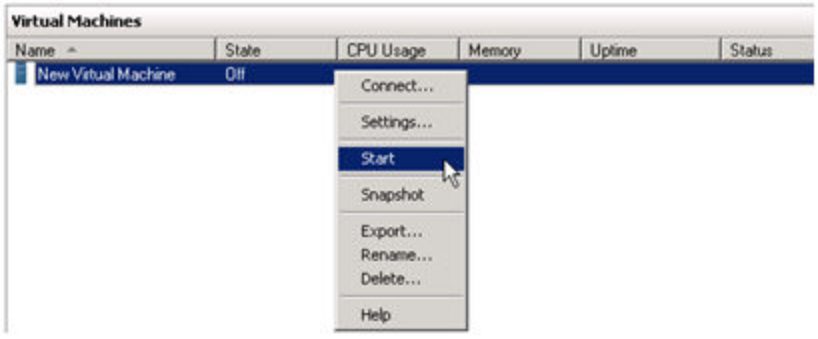
   Ensure the volume is in **mirroring** state by checking the **mirror status**. The status must indicate **Mirroring** with the **zero KB Resync Remaining** before switchover occurs.

4. Select the mirror and click **Switchover** in the **Actions** pane.



   Wait until the mirror has completely switched over and the DataKeeper user interface (UI) indicates the roles have been reversed properly.

5. Log into the **Hyper-V host server** that just became the source server in the DataKeeper interface.
6. Start **Hyper-V Manager** as described previously.
7. Start the virtual machine.

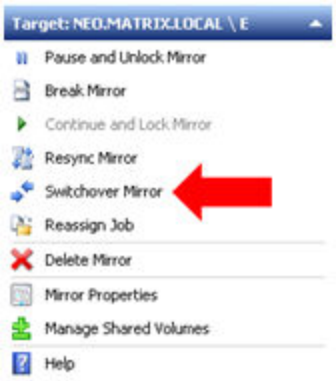


Unplanned Switchover

An unplanned switchover is necessary when a failure of some sort occurs and either the source system is unavailable or the connection between the systems is broken and requires that the virtual machine be brought online on the target server.

Since, in this scenario, the source server is unavailable for some reason, quiescing the data on the source server is not possible and as such, only the following steps are necessary on the target server to bring the virtual machine online.

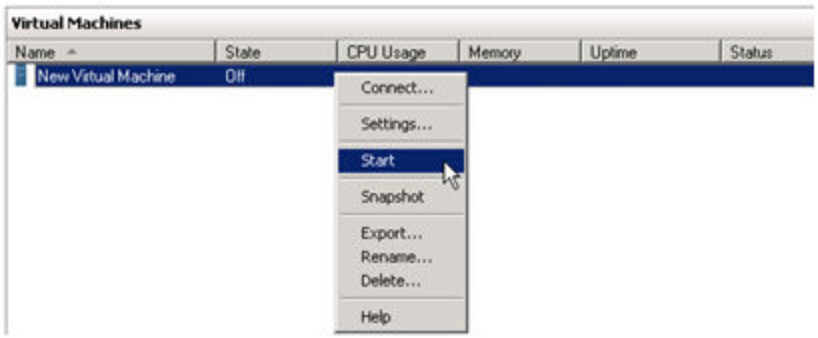
1. On the target server, start the **DataKeeper console** as described previously.
2. Select the mirror and click **Switchover** in the **Actions** pane.



   Wait until the mirror has completely come into service on the server and the DataKeeper user interface (UI) indicates the functional server is the source server.

3. On the same server, start **Hyper-V Manager** as described previously.

Start the virtual machine.



Switchback

Switchback is a planned event which transfers the virtual machine from the target server back to the source server and, in process, is exactly the same as the planned switchover process. Please refer to the steps previously listed in the **Planned Switchover** section to affect a switchback.

Chapter 6: Frequently Asked Questions

Refer to this section for answers to the most frequently asked questions about SteelEye DataKeeper.

Awareness of Windows Filenames and Directory Names

Question

Is SteelEye DataKeeper aware of Windows filenames and directory names?

Answer

SteelEye DataKeeper is implemented with a Windows kernel mode filter driver that sits above the physical disk driver but below the file system. As a result, the SteelEye DataKeeper driver knows nothing about individual files or the file system itself. It is only aware of raw writes to the disk.

Change Mirror Endpoints

Question

Can I change the mirror endpoints (IP address) of a system currently associated with an existing mirror?

Answer

Yes. The EMCMD called CHANGEMIRRORENDPOINTS allows you to change the endpoints of a mirrored volume that is configured on 3 nodes or fewer. (If your configuration consists of more than three nodes, the mirrors must be deleted and recreated.)

Change Mirror Type

Question

Can you change the mirror type of an existing mirror from Synchronous to Asynchronous or vice-versa?

Answer

No, you cannot change the mirror type of an existing mirror. You must delete and recreate the mirror and specify the new mirror type.

Create a Mirror and Rename Job and Delete Job Actions Grayed Out

Question

Why are the Create a Mirror, Rename Job and Delete Job actions grayed out?

Answer

If a node that is part of the job is down, these actions will not be enabled.

Data Transfer Network Protocols

Question

What are the network protocols used for SteelEye DataKeeper Data Transfer?

Answer

SteelEye DataKeeper uses named pipe communication and TCP Sockets.

Delete and Switchover Actions Grayed Out

Question

Why are the Delete and Switchover actions grayed out on the DataKeeper User Interface?

Answer

If the volume is under clustering protection (Microsoft clustering or SteelEye LifeKeeper clustering), these actions are disabled.

Deleting a Mirror FAQ

Question

What actually happens when you delete a mirror?

Answer

The data remains on both sides, but the target and source data are no longer synchronized. The target volume is unlocked and made fully accessible.

Error Messages Log

Question

Where does DataKeeper log error messages?

Answer

DataKeeper events are logged in the **Windows Application Event Log** and the **Windows System Event Log**. Here is a breakdown of the messages you can look for.

Application Event Log:

- Source = ExtMirrSvc – events related to the DataKeeper service.
- Source = SteelEye.SDRSnapIn – events related to the DataKeeper GUI connecting to the DataKeeper systems.

System Event Log:

- Source = ExtMirr – events directly related to mirror creation, mirror manipulation and replication.

Note: The **System Event Log** should always be set to "**Overwrite events as needed**." If the System Event Log fills up or becomes corrupted, it will prevent DataKeeper from properly recognizing mirror state changes.

Inability to Create a Mirror

Question

Why can't I create a mirror?

Answer

- The common cause of this problem is that the volume on either source or target is in use by another process. Stop the process that is accessing the volume and try again. The SteelEye DataKeeper software requires exclusive access to the target volume during the creation of the mirror.
- The target volume must be as large, or larger, than the source volume. It is recommended that the user compare the target volume size with the source volume size using the Disk Management utility. If the sizes are not the same, recreate the target partition a little larger. See Volume Considerations for more information.
- An error experienced during Create Mirror could indicate that the target volume is corrupt. If this occurs, format the target volume and attempt to create the mirror again.

Network Disconnect

Scenario #1

In a 2-Node, non-clustering configuration (1x1) replicating a 100TB volume between Source server and Target server over a WAN connection, the network goes down for twenty minutes.

Question

In this scenario, what would happen to the **Mirror State** with DataKeeper Standard Edition?

Answer

After a couple of minutes, the Source server will detect that the network is down and the mirror will go from the **MIRRORING** state to the **PAUSED** state.

Question

Does DataKeeper continue to track changes on the Source server?

Answer

Yes. The Bitmap (# of Dirty Sectors) will continue to be updated on the Source server while the mirror is in the **PAUSED** state.

Question

Once the network is resumed, will a partial resync to the Target server occur?

Answer

Yes. The mirror will go to the **RESYNC** state and remain there until all dirty sectors are written to the Target server. It will be a partial resync.

Scenario #2

In a 2-Node, non-clustering configuration (1x1) replicating a 100TB volume between Source and Target over a WAN connection, the network goes down for twelve hours. The Source server is rebooted while the network is down.

Question

In this scenario, what would happen to the status of the Source server in DataKeeper Standard Edition?

Answer

The Bitmap on the Source server is persistent (on disk), so it will not be affected by a Source reboot. Only a partial resync is needed if the Source server is rebooted. The Target server will report that it is in the **MIRRORING** state until it is reconnected to the Source server. Then it will go to the **RESYNC** state while the resync is proceeding.

Reclaim Full Capacity of Target Drive

Question

How do I reclaim the full capacity of my target drive when I no longer need it for mirroring?

Answer

The file system on the target drive is overlaid by SteelEye DataKeeper, thereby making it smaller than the actual partition size. Although Disk Management indicates the full partition size, SteelEye DataKeeper and Windows Explorer indicate the smaller mirror size. To reclaim full capacity of the drive, reformat the partition or use a partition resizing utility such as GParted (http://gparted.sourceforge.net/).

Resize or Grow Mirrored Volumes

Question

Can you resize or grow mirrored volumes?

Answer

Yes, beginning with Version 7.4, users can extend and shrink their DataKeeper volumes dynamically while retaining mirror settings. See DataKeeper Volume Resize for more information.

Split-Brain FAQs

Scenario

I am using DataKeeper in a non-cluster environment. I am mirroring from Server1 at one site to Server2 at a second site. Communication is broken due to site-to-site VPN, and I need to fail over from Server1 to Server2. I cannot access Server1 from anywhere. Server1 is actually still on but not reachable internally or externally, and there may be some processes still running in the backend.

Question

How can I fail over from Server1 to Server2?

Answer

Using the SWITCHOVERVOLUME command or the **Switchover Mirror** option in the DataKeeper UI, switch the source of the mirror to Server2. There will be a delay while the Target tries to connect to the Source, but that should complete in 30-40 seconds or so.

Question

During the switchover period, both Server1 and Server2 are writing new data to the disk (Volume F on both Server1 and Server2). When the connection comes back online, will Server1 automatically become the Target?

Answer

No. This scenario will cause a split-brain condition. Perform one of the following to resolve this issue:

- Using the DataKeeper User Interface, perform the Split-Brain Recovery Procedure.

  or

- Run the EMCMD PREPARETOBECOMETARGET command on the system that is going to become the Target, and then run the CONTINUEMIRROR command on the system that is going to become the Source.

Question

Which of the two methods above do you recommend for resolving the split-brain issue?

Answer

Whichever you prefer - they both perform the same functions.

Question

Can the command for the Target server be run from the Source server?

Answer

Yes, the command for the Target server can be run from the Source server.

Question

How does DataKeeper sync the changed and unchanged blocks?

Answer

When resolving a split-brain condition, any changes on the system that is becoming the Target will be overwritten and lost. If there are changes on that system that you want to retain, manually copy those changes to the system that is going to become the Source.

Question

When running the PREPARETOBECOMETARGET command to resolve a split-brain condition, will a full resync or partial resync occur from the Source?

Answer

The **PREPARETOBECOMETARGET** command will delete the mirror(s) on that system but will leave the volume locked. The bitmap will remain intact so that a partial resync can be performed in the next step (CONTINUEMIRROR).

Question

How can I simulate a split-brain scenario?

Answer

To simulate a split-brain scenario, unplug the network between two systems so they cannot communicate. Run the SWITCHOVERVOLUME command (or select the **Switchover Mirror** option in the DataKeeper UI) on the Target so they both become Source, then reconnect the network. You are in a split-brain condition at that point.

Question

Should I wait for the **PREPARETOBECOMETARGET** command to complete before running **CONTINUEMIRROR** on the Source?

Answer

The **PREPARETOBECOMETARGET** command completes immediately.

Stop Replication Between Source and Target

Question

How do I stop the replication between the Source and Target volumes?

Answer

Replication occurs at the driver level and can only be stopped or interrupted by sending a command from the DataKeeper GUI or the DataKeeper command line (EMCMD) to the DataKeeper driver to do one of the following:

- PAUSE the mirror – Mirror endpoints still exist, but all replication is suspended. Writes are tracked on the source system so only a partial resync of the data is necessary to bring the target volume back into sync when the mirror is CONTINUED.
- BREAK the mirror – Mirror endpoints still exist, but all replication is suspended. Writes to the source system are not tracked. RESYNCING the mirror will initiate a full resync of the data which is required to bring the target volume back into sync with the source.
- DELETE the mirror – Mirror endpoints are deleted and replication stops.

Note: Stopping the DataKeeper service does not stop replication.

Using Volume Shadow Copy

Question

Can Volume Shadow Copy (VSS) be Used with DataKeeper Volumes?

Answer

On Windows 2003 and 2003 R2, VSS Shadow Copy cannot be enabled on DataKeeper volumes. Configuring a snapshot of a replicated volume, even if the snapshot is stored on a different volume, will prevent DataKeeper from being able to lock the volume, making it impossible to protect the data on the volume.

On Windows 2008 and 2008 R2, VSS Shadow Copy can be enabled for DataKeeper volumes. However, the following guidelines apply:

- VSS snapshot images must not be stored on a DataKeeper volume. Storing VSS snapshots on a DataKeeper volume will prevent DataKeeper from being able to lock the volume and switch it over to another node.
- When a DataKeeper volume is switched or failed over, any previous snapshots that were taken of the DataKeeper volume are discarded and cannot be reused.
- VSS snapshot scheduling is not copied between the DataKeeper servers. If snapshots are scheduled to be taken twice a day on the primary server and a switchover occurs, this schedule will not be present on the backup server and will need to be redefined on the backup server.
- When switching back to a server where snapshots were previously enabled, VSS snapshots are automatically re-enabled; HOWEVER, any previous snapshots that were taken of the DataKeeper volume are discarded and cannot be reused.

Volumes Unavailable for Mirroring

Question

Why are some of my volumes not available for mirroring?

Answer

The SteelEye DataKeeper service filters out the following types of disk partitions:

- Windows system volume
- Volume(s) that contain the Windows pagefile
- Non-NTFS formatted volumes (e.g. FAT, Raw FS)
- Non-fixed drive types (e.g. CD-ROMs, diskettes)
- Target volumes that are smaller than the source volume

Chapter 7: Troubleshooting

The topics in this section contain important information about known issues and restrictions offering possible workarounds and/or solutions.

Known Issues and Workarounds

Included below are known issues open against DataKeeper as well as possible workarounds and/or solutions.

Access to Designated Volume Denied

If access to the designated volume is denied, then check whether you are attempting to create the mirror while other applications are accessing the volume. During Mirror Creation, the volumes must be locked on the target system for exclusive access by the SteelEye DataKeeper software.

In particular, the Distributed Tracking Client service, which is set to run by default in Windows, keeps two file handles open for each volume. If the volume houses a SteelEye DataKeeper target, the SteelEye DataKeeper driver cannot lock the volume. You must therefore stop the Distributed Tracking Client service and set its startup policy to Manual.

Compatibility Issue with Symantec Endpoint Protection Version 12

If using DataKeeper 7.4.2 or earlier with Symantec Endpoint Protection Version 12 (previous to 12.1.2), DataKeeper switchovers will not perform correctly due to Symantec's drivers. DataKeeper cannot lock the target volume with SEP Version 12 (previous to 12.1.2) installed on the server. The following error will be received:

```
IOCTL_DISK_IS_WRITABLE - STATUS_MEDIA_WRITE_PROTECTED

EmVolumeGetExclusiveLock - Error locking volume 0xc0000022
```

Solution: Currently, there are four workarounds for this issue.

1. Upgrade to DataKeeper for Windows 7.4.3.
2. Upgrade to Symantec Endpoint Protection Version 12.1.2 or roll back to Symantec Endpoint Protection Version 11.x .
3. Uninstall Symantec Endpoint Protection Version 12.
4. Disable the drivers as described below.
    a. Disable two LifeKeeper drivers.

`BHDrvx86/BHDrvx64` (depending on 32-bit or 64-bit OS) and `SymEFA`

b. Disable Tamper Protection

Tamper Protection must be disabled in order to successfully change the start types for the registry keys below.

1. Open the Client UI
2. Select **Change Settings**
3. Select **Configure Settings for Client Management**
4. Select the **Tamper Protection** tab and uncheck **Protect Symantec security software from being tampered with or shut down**.

c. Set the following values below:

`HKLM\System\CurrentControlSet\Services\BHDrvx86\`

Set "**`Start`**" to "`4`"

`HKLM\System\CurrentControlSet\Services\BHDrvx64\`

Set "**`Start`**" to "`4`"

`HKLM\System\CurrentControlSet\Services\SymEFA`

Set "**`Start`**" to "`4`"

d. Reboot the system. Once disabled, a reboot is required.

Note: There is some SEP client functionality loss with these drivers disabled.

Antivirus and Antispyware – Enabled (Reputation lookups disabled)

Advanced Download Protection (Browser Protection) - Disabled

Sonar Protection (Suspicious Behavior Detection) - Disabled

Network Threat Protection (Firewall) - Enabled

Intrusion Prevention - Enabled

Application and Device Control - Enabled

Symprotect (Tamper Protection) - Disabled

Note: The following warnings will be displayed on the UI:

"Proactive Threat Protection is disabled."

"Download Insight is not functioning correctly."

"Network Intrusion Prevention is not functioning correctly."

Counter Logs Do Not Work on Windows 2003

Performance Monitor - Counter Logs Do Not Work on Windows 2003

Error/Message

Performance Monitor counter logs do not work on Windows 2003. This problem will be resolved in a future release of the SteelEye DataKeeper for Windows product.

Suggested Action

Open the Services MMC snap-in and change the log on account for the "Performance Logs and Alerts" (SysmonLog) service to "Local System Account".

To reset the "Performance Logs and Alerts" (SysmonLog) service account to the original account, change the log on account back to "NT Authority/Network Service" specifying no password.

Failed to Create Mirror

User Interface - Failed to Create Mirror - Application Event Log

Error/Message

Logged in the **Application Event Log**:

File: .\GuiThread.cpp Line: 3099 Attempt to connect to remote system REMOTESERVER failed with error 5. Please ensure that the local security policy for "**Network Access: Let Everyone permissions apply to anonymous users**" is enabled on all the servers running DataKeeper.

Check: Local security policy setting on the specified system.

Description

Failed to create the mirror. Mirror is created but not stored in the job.

Suggested Action

Make local security policy change, open command prompt and run "`%EXTMIRRBASE %\emcmd. deletemirror <volume>`", then perform the mirror creation action again.

MaxResyncPasses Value

If, during a volume resynchronization, the number of passes made through the intent log exceeds the **MaxResyncPasses** registry value (200 by default), SteelEye DataKeeper logs a message to the **Event Log** indicating that the resync process is taking too many passes and requests that the administrator stop whatever process is writing to the drive being resynchronized. The mirror then goes to the **Paused** state. You can increase the **MaxResyncPasses** value from the registry to give the resync process more time.

Mirroring with Dynamic Disks

When changing from a **Basic Disk** to a **Dynamic Disk**, the underlying volume GUID may be changed by the OS upon reboot. This will cause a DataKeeper mirror to break.

Suggested Action

When mirroring with dynamic disks, your **dynamic** volumes should be created and a reboot should be performed PRIOR to creating your mirror. If the mirror has already been created, it must be deleted prior to creating your dynamic volumes.

Server Login Accounts and Passwords Must Be Same on Each Server in the Cluster

The DataKeeper GUI cannot connect to the target server in a cluster if server **Login Accounts** and **Passwords** are different on each server.

Error Message

An Error Code 1326 will appear in the Application log (**Note**: The Error Code may also be a 2 with Event ID 0):

```
SteelEye.Dialogs.AddServerWindow: Failed to connect to server:
172.17.105.112 System.ApplicationException: Failed to open a
connection to 172.17.105.112 (error_code = 1326) at
SteelEye.DAO.Impl.DataReplication.ClientLibrarySDRService.throwIfNo
nZero(UInt32 errorCode, String message) at
SteelEye.DAO.Impl.DataReplication.ClientLibrarySDRService.getServic
eInfo(String serverName) at
SteelEye.DAO.Impl.DataReplication.CachingSDRService.<>c__
DisplayClass2.<getServiceInfo>b__0() at
SteelEye.DAO.Impl.DataReplication.Cacher`1.fetch(String typekey,
String datakey, Fetcher fetcher) at
SteelEye.DAO.Impl.DataReplication.CachingSDRService.getServiceInfo
(String serverName) at
SteelEye.DataKeeper.SDR.SDRDataKeeperService.ConnectToServer(String
serverName) at SteelEye.Dialogs.AddServerWindow.<>c__
DisplayClass4.<ShowDialog>b__0(Object s, DoWorkEventArgs e) at
System.ComponentModel.BackgroundWorker.WorkerThreadStart(Object
argument)
```

```
net helpmsg 1326 shows:
```

```
Logon failure: unknown user name or bad password
```

Description/Cause

The Service Account User Names and Passwords being used to start DataKeeper are the same on both servers and the firewalls are disabled on the servers; however, the Passwords used to log in to the servers themselves are different.

Suggested Action

The DataKeeper GUI uses the server Login ID and Password; therefore, the User Name and Password used to log in to the servers themselves must be the same on each server and must have administrator privileges.

System Event Log - Create Mirror Failed in the GUI

Error/Message

Create Mirror Failed in the GUI.

Description

This can result if a vmms.exe program is holding on to volume and preventing SteelEye DataKeeper from locking it.

Unable to Determine Previous Install Path

Installation - Fatal Error: Unable to Determine Previous Install Path

Error/Message

Fatal Error: Unable to determine previous install path. DataKeeper cannot be uninstalled or reinstalled.

Description

When performing a "**Repair**" or "**Uninstall**" of DataKeeper, the "**ExtMirrBase**" value is missing in the installation path of DataKeeper in the registry under ***HKLM\System\CurrentControlSet\Control\Session Manager\Environment***.

Suggested Action

Perform one of the following:

- Under the **Environment** key, create "**ExtMirrBase**" as a REG_SZ and set the value to the DataKeeper installation path (i.e. ***C:\Program Files(x86)\SteelEye\DataKeeper*** for x64, or ***C:\ProgramFiles\ Steeleye\DataKeeper*** for x86).
- To force InstallShield to perform a new install of DataKeeper, delete the following registry key:

```
HKLM\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Uni
nstall\
{B00365F8-E4E0-11D5-8323-0050DA240D61}.
```

This should be the installation key created by InstallShield for the DataKeeper v7.1.0 product.

User Interface - Failed to Create Mirror

User Interface - Failed to Create Mirror, Event ID 137

Error/Message

Failed to create the mirror.
Event Id: 137
System Event Log

Unable to initialize mirror on the target machine.

Volume Device:
Source Volume: E
Target Machine: 10.17.103.135
Target Volume: E
Failed operation: Target reports error
Error Code: 0xC0000055

Description

DataKeeper cannot lock the Target volume during mirror creation.

Suggested Action

1. Verify the Distributed Link Tracking Client service is not running on either system.
2. Stop any other processes that may prevent DataKeeper from locking the Target volume (e.g. anti-virus software).
3. Recreate the mirror.

User Interface - Shows Only One Side of the Mirror

If the SteelEye DataKeeper UI shows a volume as a source and its corresponding target as available or a volume as a target with the corresponding source volume as available, you can use the command line utility to force an update to the SteelEye DataKeeper GUI or delete the orphaned side of the mirror. From a command prompt, go to the SteelEye DataKeeper directory on the server which is displaying unexpected mirror status and perform the following steps:

1. Make sure that the mirror is not in a **Paused** or **Broken** state on the source. If so, continue the mirror on the source. This should result in the mirror being re-established to the target.
2. Run EMCMD <system name> UpdateVolumeInfo <volume letter>

   Where

   <system name> is the name of the system;

   <volume letter> is the letter of the volume.
3. If the problem is not resolved in Step 1, then stop and restart the SteelEye DataKeeper service.

Windows Server 2012 Specific Issues

For issues related to **Windows Server 2012**, see the following topics:

Windows Server 2012 MMC Snap-in Crash

Description

When using the DataKeeper user interface (MMC Snap-in) on Windows Server 2012, the `mmc.exe` process may crash unexpectedly due to an internal .Net or Windows Presentation Foundation (WPF) issue. The error may show up on the screen and/or the event viewer.

Suggested Action

This crash does not affect the server(s) to which the snap-in was connected or any DataKeeper mirrors established at the time of the crash. The MMC Snap-in may be safely relaunched. Simply close the UI and restart it.

The following are examples of **Application Event Log messages** that may be logged during this failure.

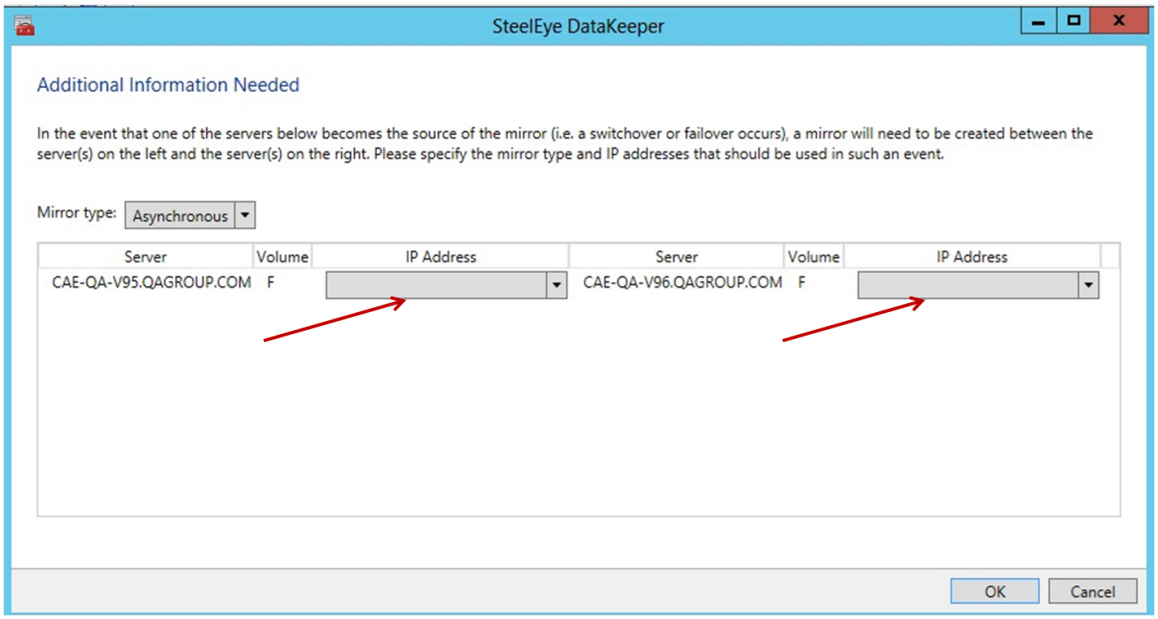
```
-----------------------------------------------------------
Log Name: Application
Source: Desktop Window Manager
Date: 11/28/2012 8:34:00 AM
Event ID: 9009
Task Category: None
Level: Information
Keywords: Classic
User: N/A
Computer: CAE-QA-V96.QAGROUP.COM
Description:
The Desktop Window Manager has exited with code (0xd00002fe)
-----------------------------------------------------------



-----------------------------------------------------------
Log Name: Application
Source: .NET Runtime
Date: 11/28/2012 8:34:00 AM
Event ID: 1026
Task Category: None
Level: Error
Keywords: Classic
User: N/A
Computer: CAE-QA-V96.QAGROUP.COM
Description:
Application: mmc.exe
Framework Version: v4.0.30319
Description: The process was terminated due to an unhandled exception.
-----------------------------------------------------------



-----------------------------------------------------------
Log Name: Application
Source: Application Error
```

```
Date: 11/28/2012 8:34:00 AM
Event ID: 1000
Task Category: (100)
Level: Error
Keywords: Classic
User: N/A
Computer: CAE-QA-V96.QAGROUP.COM
Description:
Faulting application name: mmc.exe, version: 6.2.9200.16384, time stamp:
0x50109efd
Faulting module name: KERNELBASE.dll, version: 6.2.9200.16384, time stamp:
0x5010ab2d
Exception code: 0xe0434352
Fault offset: 0x00000000000189cc
Faulting process id: 0xdc4
Faulting application start time: 0x01cdccd27c68a1c6
Faulting application path: C:\Windows\system32\mmc.exe
Faulting module path: C:\Windows\system32\KERNELBASE.dll
Report Id: 443c3ed3-3960-11e2-9400-0050569b131b
Faulting package full name:
Faulting package-relative application ID:
```

Windows Server 2012 Default Information Missing During Mirror Creation

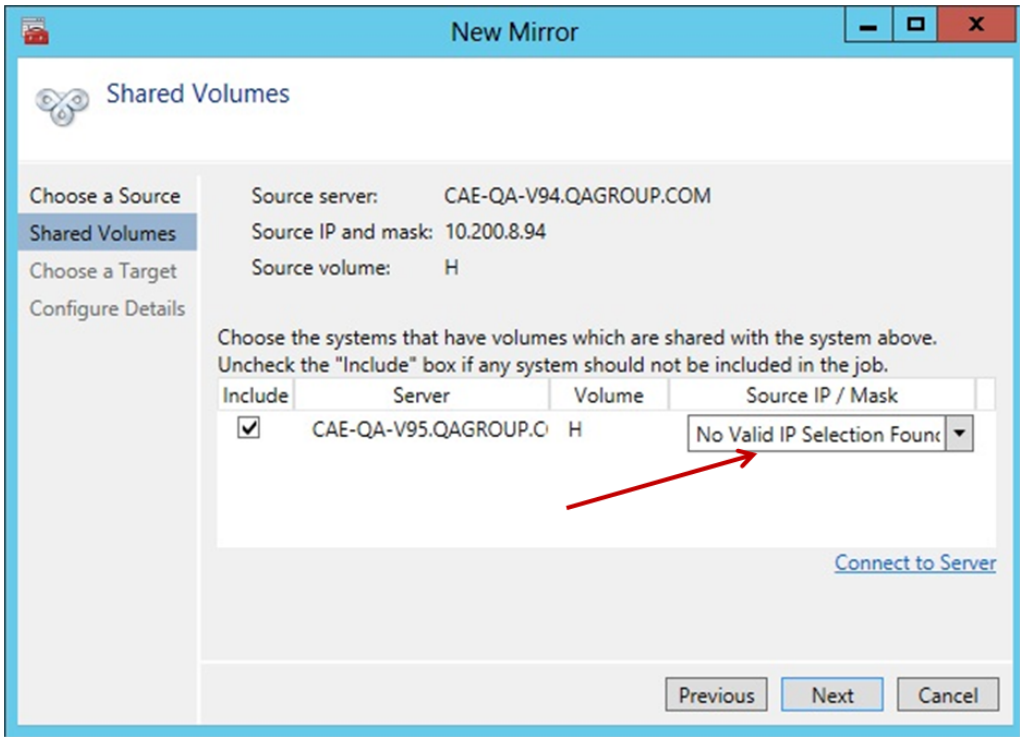
Creating Mirrors with Multiple Targets

The first issue is during mirror creation in a multi-target configuration. In the final step, the user is prompted for secondary relationship information. In previous OS versions, a default Source IP is provided on this **Additional Information Needed** dialog. In Windows Server 2012, however, this default IP is not provided, but the correct IP address must still be selected. If **OK** is clicked without selecting the IP address, the mirror will still create, but key relationship information will be missing.



Creating Mirrors with Shared Volumes

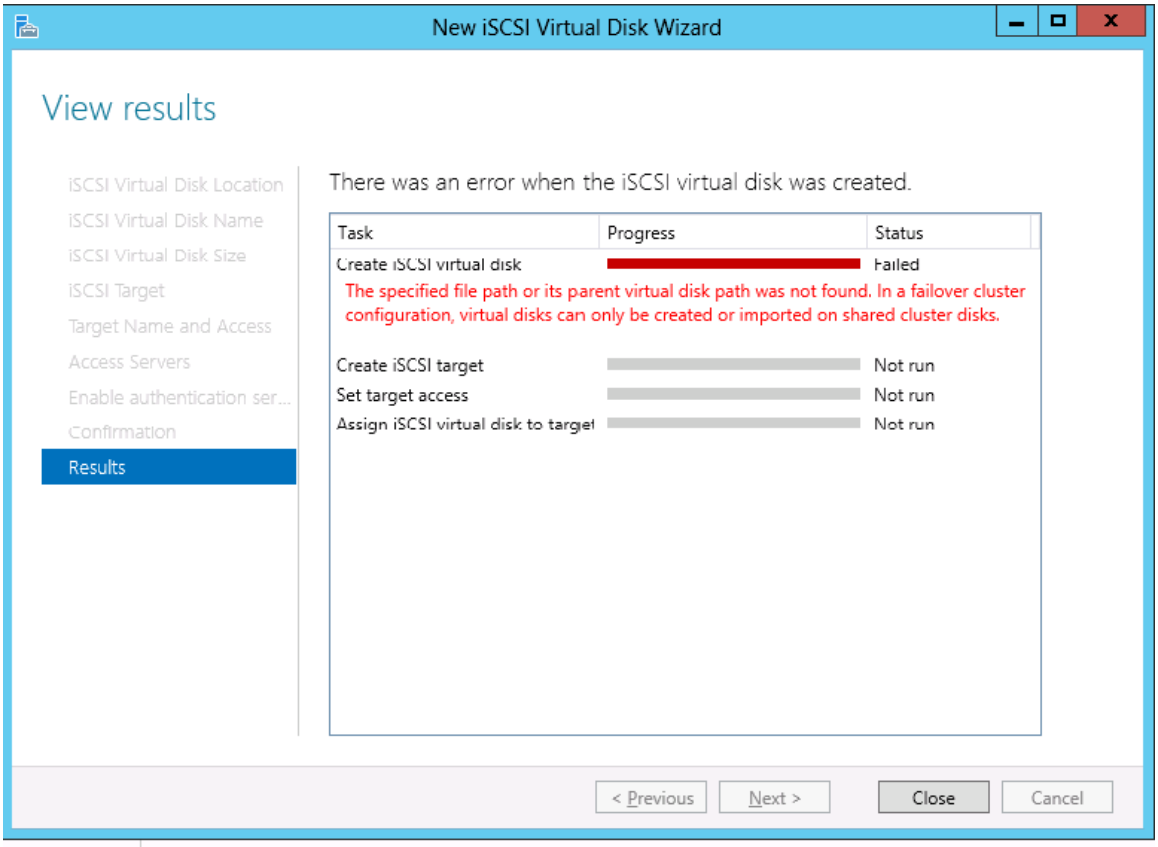
The other issue is with the **Shared Volumes** dialog box when creating mirrors with shared volumes. In previous OS versions, a default Source IP is provided on this screen. In Windows Server 2012, however, this dialog will display "**No Valid IP Selection Found**." The correct Source IP will still need to be selected.



Windows Server 2012 iSCSI Target Role Does Not Support Dynamic Disks

Description

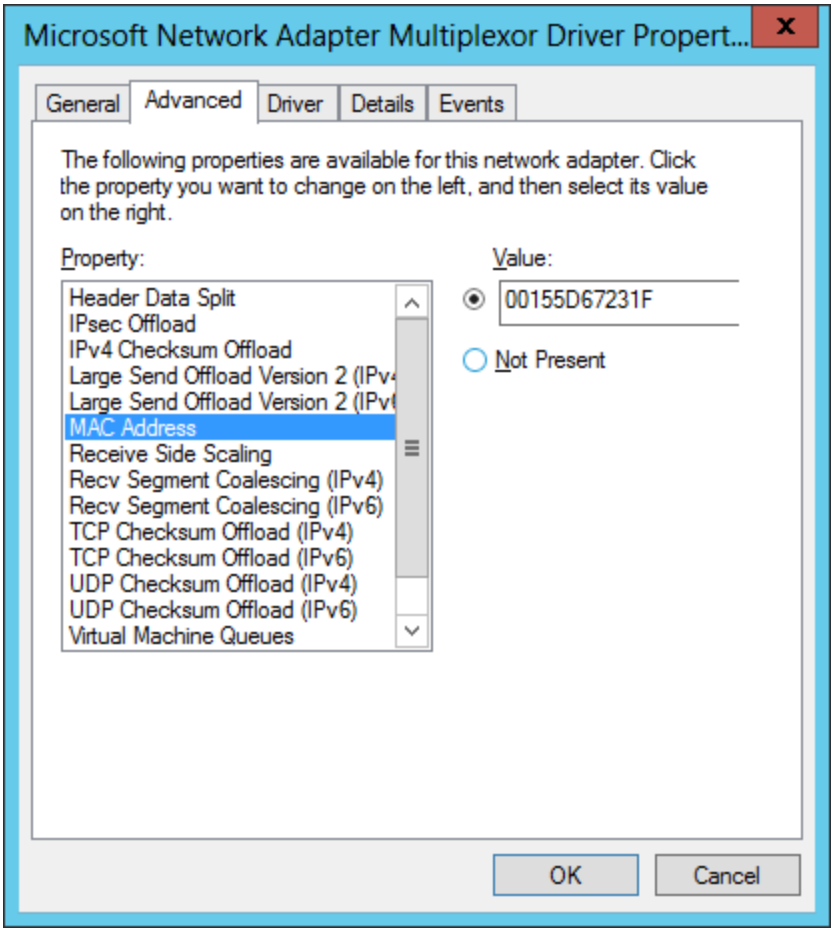
The iSCSI Target Role only supports DataKeeper Volumes that are mirrors of **Simple Volumes** placed on **Basic Disks**. If any of your mirrors are using volumes such as Striped or Spanned volumes on a Dynamic Disk on either the source or target system, then you cannot create an iSCSI Target role that uses those DataKeeper Volume resources for storage.



Windows Server 2012 NIC Teaming Issue

If you use the **NIC Teaming** feature of Windows Server 2012, Windows 2012 will report only one adapter MAC address for the license. If you have many underlying adapters, the MAC address will arbitrarily change and Windows may pick one of the adapters that may no longer be licensed.

To resolve this issue, configure the **MAC address** property of the virtual team adapter. This property can be changed using the **Advanced** tab of the **Adapter Properties** as shown in the diagram:



Restrictions

Included below are restrictions associated with DataKeeper as well as possible workarounds and/or solutions.

Bitlocker Does Not Support DataKeeper

According to Microsoft, Bitlocker is not supported to work with Software RAID configurations. Since DataKeeper is essentially a software RAID 1, Microsoft does not support Bitlocker working with DataKeeper.

The specific article and section can be found here:

> http://technet.microsoft.com/en-us/library/ee449438#BKMK_R2disks

CHANGEMIRRORENDPOINTS

Description:

This command, which is used to move a DataKeeper protected volume to another network location, only supports changing the endpoints of a mirrored volume that is configured on 3 nodes or fewer.

Workaround:

For configurations greater than three nodes, the mirrors must be deleted and recreated with the final endpoint at the local site and use route adds to get the mirrors created and resynced before moving the server to the final location/address/DR site.

CHKDSK

Description

If you must run `CHKDSK` on a volume that is being replicated by SteelEye DataKeeper, it is recommended that you **PAUSE** the mirror before initiating the `CHKDSK`. After running `CHKDSK`, **CONTINUE** the mirror. A partial resync occurs (updating those writes generated by the `CHKDSK`) and replication will continue.

Note: The bitmap file (for non-shared volumes) is located on the C drive which is defined by BitmapBaseDir as the default location. Running CHKDSK on the C drive of the **Source** system will cause an error due to the active bitmap file. Therefore, a switchover must be performed so that this Source becomes Target and the bitmap file becomes inactive. The CHKDSK can then be executed on this system as the new target (original source).

DataKeeper Volume Resize Restriction

The DataKeeper volume resize procedure should be performed on only one volume at a time.

Directory for Bitmap Must Be Created Prior to Relocation

Description

If you choose to relocate the bitmap file from the default location (***%EXTMIRRBASE%\Bitmaps***), you must first create the new directory before changing the location in the registry and rebooting the system.

Intensive I-O with Synchronous Replication

Description

Due to the nature of synchronous replication (blocking volume writes while waiting for a response from the target system), you may experience sluggish behavior with any applications that are writing to the mirrored volume. The frequency of these events could be high depending on the ratio of "Volume I/O traffic" to "system resource". It is recommended that you use asynchronous replication when continuous and intensive I/O traffic is anticipated for the volume or when SteelEye DataKeeper is used on a low bandwidth network.

Path Name Restriction

Description

SteelEye DataKeeper must be installed into a directory with 32 characters or less in the path name. Failure to do this will result in non-functioning performance monitor counters for SteelEye DataKeeper as well as failures when attempting to use the environment variable **%EXTMIRRBASE%**.

Resource Tag Name Restrictions

Tag Name Length

All tags within DataKeeper may not exceed the 256 character limit.

Valid "Special" Characters

-	_	.	/					

However, the first character in a tag should not contain "." or "/".

Invalid Characters

+	;	:	!	@	#	$	*	=	"space"

Index

A

B

C

M

N

P

R

S

T

U

V

W