IBM XIV Storage System Gen3
Version 11.6.2

Product Overview

GC27-3912-10

Note
Before using this document and the product it supports, read the information in "Notices" on page 145.

Edition notice

Publication number: GC27-3912-10. This publication applies to IBM XIV Storage System version 11.6.2 and to all subsequent releases and modifications until otherwise indicated in a newer publication.

Contents

Figures

Tables

About this document

This document provides a technical overview of the IBM XIV Storage System functional features and capabilities.

Intended audience

This document is intended for technology officers, enterprise storage managers, and storage administrators who want to learn about the different functional features and capabilities of IBM FlashSystem® Storage System.

Document conventions

These notices are used in this guide to highlight key information.

Note: These notices provide important tips, guidance, or advice.

Important: These notices provide information or advice that might help you avoid inconvenient or difficult situations.

Attention: These notices indicate possible damage to programs, devices, or data. An attention notice appears before the instruction or situation in which damage can occur.

Related information and publications

You can find additional information and publications related to IBM XIV Storage System on the following information sources.

- IBM XIV Storage System on IBM® Knowledge Center (ibm.com/support/knowledgecenter/STJTAG) – on which you can find the following related publications:
  - IBM XIV Storage System – Release Notes
  - IBM XIV Storage System – Planning Guide
  - IBM XIV Storage System – Command-Line Interface (CLI) Reference Guide
  - IBM XIV Storage System – API Reference Guide
  - IBM Hyper-Scale Manager – Release Notes
  - IBM Hyper-Scale Manager – User Guide
  - IBM Hyper-Scale Manager – Quick-Start Guide
  - IBM Hyper-Scale Manager – Representational State Transfer (REST) API Specifications
  - Management Tools Operations Guide
  - Management Tools XCLI Utility User Guide
- IBM Storage Redbooks® website (redbooks.ibm.com/portals/storage)

IBM Publications Center

The IBM Publications Center is a worldwide central repository for IBM product publications and marketing material.

The IBM Publications Center website (ibm.com/shop/publications/order) offers customized search functions to help you find the publications that you need. You can view or download publications at no charge.

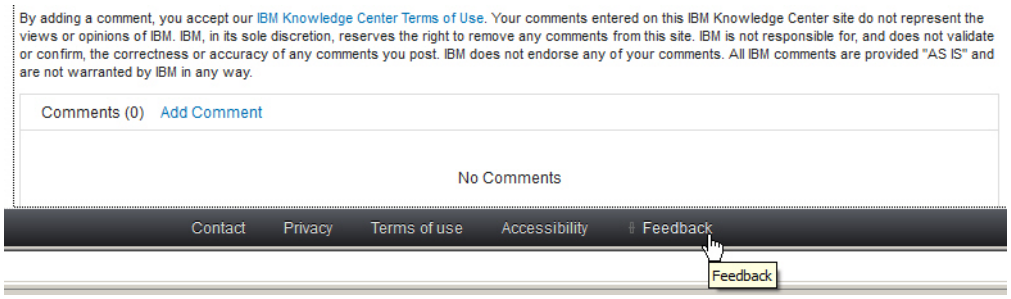
Sending your comments

Your feedback is important in helping to provide the most accurate and highest quality information.

Procedure

To submit any comments about this guide or any other IBM XIV® Storage System documentation:

- Go to http://www-01.ibm.com/support/knowledgecenter/STJTAG/com.ibm.help.xivgen3.doc/xiv_kcwelcomepage.html (http://www-01.ibm.com/support/knowledgecenter/STJTAG/com.ibm.help.xivgen3.doc/xiv_kcwelcomepage.html), drill down to the relevant page, and click the **Feedback** link that is located at the bottom of the page. You can use this form to enter and submit comments privately.



- Post a public comment on the Knowledge Center page that you are viewing by clicking **Add Comment**. For this option, you must first log in to IBM Knowledge Center with your IBM ID.
- Send your comments by email to starpubs@us.ibm.com. Be sure to include the following information:
  - Exact publication title and version
  - Publication form number (for example, GA32-0770-00)
  - Page, table, or illustration numbers that you are commenting on
  - A detailed description of any information that needs to be changed

Getting information, help, and service

If you need help, service, technical assistance, or want more information about IBM products, you can find various sources to assist you. You can view the following websites to get information about IBM products and services and to find the latest technical information and support.

- IBM website (ibm.com®)
- IBM Support Portal website (ibm.com/storage/support)

- IBM Directory of Worldwide Contacts website (ibm.com/planetwide)
- IBM developerWorks Answers website (www.developer.ibm.com/answers)
- IBM service requests and PMRs (ibm.com/support/servicerequest/Home.action)

Use the Directory of Worldwide Contacts to find the appropriate phone number for initiating voice call support. Voice calls arrive to Level 1 or Front Line Support.

Chapter 1. Introduction

IBM XIV is a high-end grid-scale storage system that delivers consistently high performance, high resiliency and management simplicity while offering exceptional data economics, including powerful real-time compression. Industry benchmarks underscore stellar XIV performance and cost benefits.

Figure 1. IBM XIV Storage System unit

As a grid-scale offering, every IBM XIV storage system contains multiple modules that are interconnected by integrated InfiniBand switches, forming a scale-out grid fabric that delivers exceptional IOPS performance. In addition, it includes a maintenance module for remote access to the system and an uninterruptible power supply modules to ensure system operation if an external power source fails.

IBM XIV storage system is ideal for cloud environments, offering predictable service levels for dynamic workloads, simplified scale management including in multi-tenant environments, flexible consumption models — and robust cloud automation and orchestration through OpenStack, RESTful API, and VMware. It offers data-at-rest encryption, advanced mirroring and self-healing, and provides

investment protection with perpetual licensing that is transferable to all Spectrum Accelerate Family offerings (XIV Gen3, FlashSystem A9000/A9000R and Spectrum Accelerate).

Features and functionality

IBM XIV Storage System is characterized by powerful features and functionality.

These features and functionality include:

Performance

- Perfect load balancing
- Cache and disks in every module
- Extremely fast rebuild time in the event of disk failure
- Constant, predictable high performance with zero tuning

Reliability

- Unique data distribution method that eliminates "hot spots"
- Fault tolerance, failure analysis, and self-healing algorithms
- No single-point-of-failure

Scalability

- Support for thin provisioning
- Support for instant space reclamation
- Data migration

Connectivity

- iSCSI and Fibre Channel (FC) interfaces
- Multiple host access

Snapshots

- Innovative snapshot functionality, including support for practically unlimited number of snapshots, snap-of-snap and restore-from-snap

Replication

- Synchronous and asynchronous replication of a volume (as well as a consistency group) to a remote system

Ease of management

- Support for storage pools administrative units
- Remote configuration management
- Non-disruptive maintenance and upgrades
- Management software, including a graphical user interface (GUI) and a command-line interface (CLI)
- Notifications of events through e-mail, SNMP, or SMS messages
- XIV is supported by the following IBM products:
  - IBM Power Virtualization Center (PowerVC). This is an advanced virtualization management offering that simplifies creating and managing virtual machines on IBM Power Systems™ servers using PowerVM® or PowerKVM hypervisors.
  - IBM Spectrum Protect Snapshot. Formerly Tivoli® Storage FlashCopy® Manager, IBM® Spectrum Protect™ Snapshot delivers high levels of protection for key applications and databases using advanced integrated application snapshot backup and restore capabilities.

– IBM Spectrum Control. Formerly IBM Tivoli Storage Productivity Center, IBM Spectrum Control is integrated data and storage management software that provides monitoring, automation and analytics for organizations with multiple storage systems.

Hardware overview

This section provides a general overview of the IBM XIV Storage System hardware.

Hardware components

The IBM XIV Storage System configuration includes data modules, interface modules, Ethernet switches, and uninterruptible power supply units.

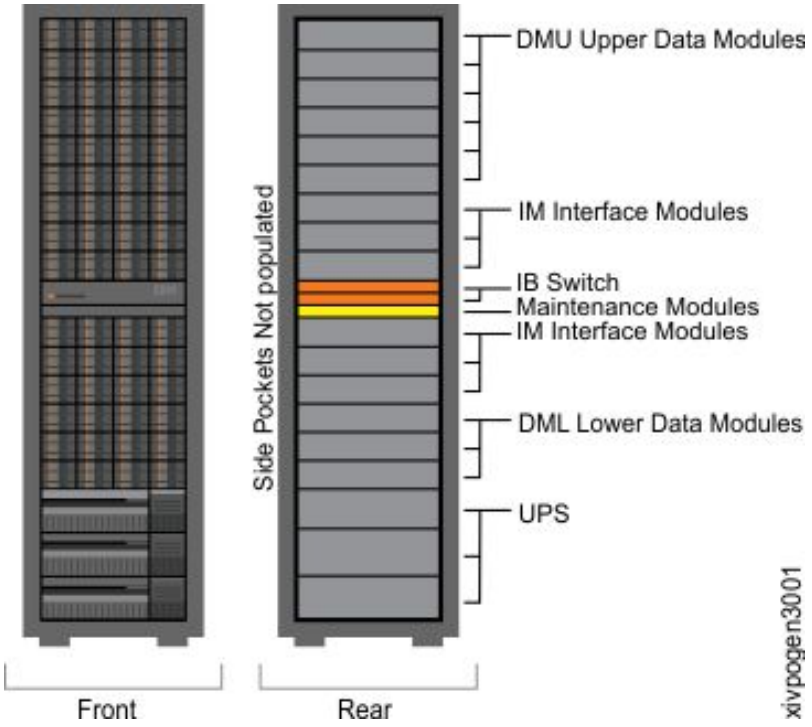


Figure 2. IBM XIV Storage System

Data modules

Each data module contains 12 disks DDR3 and 24GB of cache memory. IBM XIV supports all sorts of Near-line (7200RPM) SAS drives, in particular 2TB, 4TB, or 6TB disks. The disk drives serve as the nonvolatile memory for storing data in the storage grid and the cache memory is used for caching data previously read, prefetching of data from a disk, and for delayed destaging of previously written data.

Note: Data modules are located on both upper and lower sections of the rack.

InfiniBand switch (x2)

- Dual power supply
- Each IB switch is connected to each XIV module
- Both IB switches are inter-connected on their ports 16, 17

- Port 18 is left empty for spare
- Ports 19-36 are inactive

Maintenance module
Allows remote support access using a modem.

Interface modules (IM)
Each contains disk drives and cache memory similar to the data modules. In addition, these modules have Host Interface Adapters with FC and iSCSI ports.

8Gbps FC

- 2 dual-port FC HBAs on each interface module, i.e. 4 ports on each interface module, i.e. 24 ports on a full rack.

Uninterruptible power supply module complex
The uninterruptible power supply module complex consists of three units. It maintains an internal power supply in the event of a temporary failure of the external power supply. In the case of a continuous external power failure, the uninterruptible power supply module complex maintains power long enough for a safe and ordered shutdown of the IBM XIV Storage System. The IBM XIV Storage System can sustain the failure of one uninterruptible power supply unit while protecting against external power failures.

ATS The Automatic Transfer Switch (ATS) switches between line cords in order to allow redundancy of external power.

Modem
Allows the system to receive a connection for remote access by IBM support. The modem connects to the maintenance module.

Data and interface modules are generically referred to as "modules". Modules communicate with each other by means of the PCIe adapter. Each module contains redundant ports for module to module communication. The ports are all linked to the internal network through the switches. In addition, for monitoring purposes, the UPSs are directly connected to the individual modules.

Hardware enhancements

IBM XIV Storage System Gen3 model 314 is a hardware-enhanced XIV Gen3 storage array targeted to customers who want high utilization of IBM Real-time Compression (RtC).

With double the RAM and CPU resources, IBM XIV Storage System model 314 delivers improved IOPS per compressed capacity and 1 to 2 PB of effective capacity without performance degradation. IBM XIV Storage System model 314 hardware enhancements include:

- 2 x 6-core CPUs per module (versus 1 x 6-core CPU per module in Model 214)
- 96 GB RAM per module (versus 48 GB RAM per module in Model 214)

Note: The additional CPU and 48 GB RAM are dedicated to Real-time Compression. For more information on Real-time Compression, see Chapter 7, “IBM Real-time Compression with XIV,” on page 49.

Available configurations

IBM XIV Storage System model 314 is available for ordering in the following configurations:

- 9 to 15 modules in a system
- 4 TB or 6 TB drives
- 800 GB SSD cache (mandatory)

IBM XIV Storage System version 11.6.2 with IBM XIV Storage Gen3 System Model 314 also supports the following:

- Up to 2 PB of available soft capacity
- Reduced minimum compressible volume size from 103 GB (in model 214) to 51 GB
- Support for IBM Spectrum Accelerate software licenses

To learn more about IBM Real-time Compression, go to Chapter 7, "IBM Real-time Compression with XIV," on page 49.

For more information, see the *IBM XIV Storage System Release Notes*, version 11.6.1 documentation.

Management options

The IBM XIV Storage System provides several management options.

GUI and CLI management applications
: These applications must be installed on each workstation that will be used for managing and controlling the system. All configurations and monitoring aspects of the system can be controlled through the GUI or the CLI.

SNMP
: Third-party SNMP-based monitoring tools are supported using the IBM XIV MIB.

E-mail notifications
: The IBM XIV Storage System can notify users, applications or both through e-mail messages regarding failures, configuration changes, and other important information.

SMS notifications
: Users can be notified through SMS of any system event.

Reliability

IBM XIV Storage System reliability features include data mirroring, spare storage capacity, self-healing mechanisms, and data virtualization.

Redundant components

IBM XIV Storage System hardware components are fully redundant, and ensure failover protection for each other to prevent a single point of system failure.

System failover processes are transparent to the user because they are swiftly and seamlessly completed.

Data mirroring

Data arriving from the host for storage is temporarily placed in two separate caches before it is permanently written to two disk drives located in separate modules. This guarantees that the data is always protected against possible failure of individual modules, and this protection is in effect even before data has been written to the nonvolatile disk media.

Self-healing mechanisms

The IBM XIV Storage System includes built-in mechanisms for self-healing to take care of individual component malfunctions and to automatically restore full data redundancy in the system within minutes.

Self-healing mechanisms dramatically increase the level of reliability in the IBM XIV Storage System. Rather than necessitating a technician's on-site intervention in the case of an individual component malfunction to prevent a possible malfunction of a second component, the automatically restored redundancy allows a relaxed maintenance policy based on a pre-established routine schedule.

Self-healing mechanisms are not just started in a reactive fashion following an individual component malfunction, but also proactively - upon detection of conditions indicating potential imminent failure of a component. Often, potential problems are identified well before they might occur with the help of advanced algorithms of preventive self-analysis that are continually running in the background. In all cases, self-healing mechanisms implemented in the IBM XIV Storage System identify all data portions in the system for which a second copy has been corrupted or is in danger of being corrupted. The IBM XIV Storage System creates a secure second copy out of the existing copy, and it stores it in the most appropriate part of the system. Taking advantage of the full data virtualization, and based on the data distribution schemes implemented in the IBM XIV Storage System, such processes are completed with minimal data migration.

As with all other processes in the system, the self-healing mechanisms are completely transparent to the user, and the regular activity of responding to I/O data requests is thoroughly maintained with no degradation to system performance. Performance, load balance, and reliability are never compromised by this activity.

Protected cache

IBM XIV Storage System cache writes are protected. Cache memory on a module is protected with ECC (Error Correction Coding). All write requests are written to two separate cache modules before the host is acknowledged. The data is later destaged to disks.

Redundant power

Redundancy of power is maintained in the IBM XIV Storage System through the following means:

- Three uninterruptible power supply units - the system can run indefinitely on two uninterruptible power supply units. No system component will lose power if a single uninterruptible power supply unit fails.
- Redundant power supplies in each data and interface module. There are two power supplies for each module and each power supply for a module is powered by a different uninterruptible power supply unit.

- Redundant power for Ethernet switches - each Ethernet switch is powered by two uninterruptible power supply units. One is a direct connect; one is through the Ethernet switch redundant power supply.
- Redundant line cords - to protect against the loss of utility power, two line cords are supplied to the ATS. If utility power is lost on one line cord, the ATS automatically switches to the other line cord, without impacting the system.
- In the event of loss of utility power on both line cords, the uninterruptible power supply units will maintain power to the system until an emergency destage of all data in the system can be performed. Once the emergency destage has completed, the system will perform a controlled power down.

SSD cache drives

The IBM XIV Storage System uses Flash-as-Cache (SSD) as a second, read-only, caching layer between the cache node and the disks.

This way, the system reduces disk access with having a cache that is an order-of-magnitude larger than the DRAM cache. Currently, the system features one SSD disk per module. Flash-as-cache is designed in module granularity, so lacking this feature in one module does not affect its functionality on other modules. Flash-as-cache can be enabled and disabled at run time, so a storage system can be equipped with SSDs anytime.

Installation

The SSD is a new hardware component that is identified as `1:SSD:<module:1>`. It is automatically added when plugged in (no equip command needed). The SSD can be phased-out, tested and phased-in like a regular disk drive. Since it is read-cache only, no rebuild is triggered due to SSD component state change. It is implemented and accessed similarly to other disks.

Performance

The IBM XIV Storage System is a ground breaking, high performance storage product designed to help enterprises overcome this challenge through an exceptional mix of game-changing characteristics and capabilities

Breakthrough architecture and design
The revolutionary design of IBM XIV Storage System enables exceptional performance optimization typically unattainable by traditional architectures. This optimization results in superior utilization of system resources and automatic workload distribution across all system hard drives. It also empowers administrators to tap into the system's rich set of built-in, advanced functionality such as thin provisioning, mirroring and snapshots without adversely affecting performance.

Consistent, predictable performance and scalability
The IBM XIV Storage System's ability to optimize load distribution across all disks for all workloads, coupled with a powerful distributed cache implementation, facilitates high performance that scales linearly with added storage enclosures. Because this high performance is consistent—without the need for manual tuning—users can enjoy the same high performance during the typical peaks and troughs associated with volume and snapshot usage patterns, even after a component failure.

Resilience and self-healing
The IBM XIV Storage System maintains resilience during hardware failures,

continuing to function with minimal performance impact. Additionally, the solution's advanced self-healing capabilities allow it to withstand additional hardware failures once it recovers from the initial failure.

Automatic optimization and management
Unlike traditional storage solutions, the IBM XIV Storage System automatically optimizes data distribution through hardware configuration changes such as component additions, replacements or failure. This helps eliminate the need for manual tuning or optimization.

Functionality

IBM XIV Storage System functions include point-in-time copying, automatic notifications, and ease of management through a GUI or CLI.

Snapshot management

The IBM XIV Storage System provides powerful snapshot mechanisms for creating point-in-time copies of volumes.

The snapshot mechanisms include the following features:

- Differential snapshots, where only the data that differs between the source volume and its snapshot consumes storage space
- Instant creation of a snapshot without any interruption of the application, making the snapshot available immediately
- Writable snapshots, which can be used for a testing environment; storage space is only required for actual data changes
- Snapshot of a writable snapshot can be taken
- High performance that is independent of the number of snapshots or volume size
- The ability to restore from snapshot to volume or snapshot

Consistency groups for snapshots

Volumes can be put in a consistency group to facilitate the creation of consistent point-in-time snapshots of all the volumes in a single operation.

This is essential for applications that use several volumes concurrently and need a consistent snapshot of all these volumes at the same point in time.

Storage pools

Storage pools are used to administer the storage resources of volumes and snapshots.

The storage space of the IBM XIV Storage System can be administratively portioned into storage pools to enable the control of storage space consumption for specific applications or departments.

Remote monitoring and diagnostics

IBM XIV Storage System can email important system events to IBM Support.

This allows IBM to immediately detect hardware failures warranting immediate attention and react swiftly (for example, dispatch service personnel). Additionally, IBM support personnel can conduct remote support and generate diagnostics for

both maintenance and support purposes. All remote support is subject to customer permission and remote support sessions are protected with a challenge response security mechanism.

SNMP

Third-party SNMP-based monitoring tools are supported for the IBM XIV Storage System MIB.

Multipathing

The parallel design underlying the activity of the Host Interface modules and the full data virtualization achieved in the system implement thorough multipathing access algorithms.

Thus, as the host connects to the system through several independent ports, each volume can be accessed directly through any of the Host Interface modules, and no interaction has to be established across the various modules of the Host Interface array.

Automatic event notifications

The system can be set to automatically transmit appropriate alarm notification messages through SNMP traps, or email messages.

The user can configure various triggers for sending events and various destinations depending on the type and severity of the event. The system can also be configured to send notifications until a user acknowledges their receipt.

Management through GUI and CLI

The IBM XIV Storage System offers a user-friendly and intuitive GUI application and CLI commands to configure and monitor the system.

These feature comprehensive system management functionality, encompassing hosts, volumes, consistency groups, storage pools, snapshots, mirroring relationships, data migration, events, and more.

For more information, see the *IBM XIV Management Tools Operations Guide* and *IBM XIV Storage System XCLI User Manual*.

External replication mechanisms

External replication and mirroring mechanisms in the IBM XIV Storage System are an extension of the *internal replication mechanisms* and of the overall functionality of the system.

These features provide protection against a site disaster to ensure production continues. The mirroring can be performed over either Fibre Channel or iSCSI, and the host-to-storage protocol is independent of the mirroring protocol.

Upgradability

The IBM XIV Storage System is available in a partial rack system comprised of as few as six (6) modules, or as many as fifteen (15) modules per rack.

Partial rack systems may be upgraded by adding data and interface modules, up to the maximum of fifteen (15) modules per rack.

The system supports a non-disruptive upgrade of the system, as well as hotfix updates.

Chapter 2. Volumes and snapshots

This section gives an overview of volumes and snapshots.

Volumes are the basic storage data units in the IBM XIV Storage System. Snapshots of volumes can be created, where a snapshot of a volume represents the data on that volume at a specific point in time. Volumes can also be grouped into larger sets called consistency groups and storage pools.

The basic hierarchy may be described as follows:

- A volume can have multiple snapshots.
- A volume can be part of one and only one consistency group.
- A volume is always a part of one and only one storage pool.
- All volumes in a consistency group must belong to the same storage pool.

The following subsections deal with volumes and snapshots specifically.

Volume function and lifecycle

The volume is the basic data container that is presented to the hosts as a logical disk.

The term *volume* is sometimes used for an entity that is either a volume or a snapshot. Hosts view volumes and snapshots through the same protocol. Whenever required, the term *master volume* is used for a volume to clearly distinguish volumes from snapshots.

Each volume has two configuration attributes: a name and a size. The volume name is an alphanumeric string that is internal to the IBM XIV Storage System and is used to identify the volume to both the GUI and CLI commands. The volume name is not related to the SCSi protocol. The volume size represents the number of blocks in the volume that the using host detects.

Support for Symantec Storage Foundation Thin Reclamation

The IBM XIV Storage System supports Symantec's Storage Foundation Thin Reclamation API.

The IBM XIV Storage System features instant space reclamation functionality, enhancing the existing IBM XIV Thin Provisioning capability. The new instant space reclamation function allows IBM XIV users to optimize capacity utilization, thus saving costs, by allowing supporting applications, to instantly regain unused file system space in thin-provisioned XIV volumes instantly.

The IBM XIV Storage System is one of the first high-end storage systems to offer instant space reclamation. The new, instant capability enables third party products vendors, such as Symantec Thin Reclamation, to interlock with the The IBM XIV Storage System such that any unused space is detected instantly and automatically, and immediately reassigned to the general storage pool for reuse.

This enables integration with thin-provisioning-aware Veritas File System (VxFS) by Symantec, which ultimately enables to leverage the IBM XIV Storage System thin-provisioning-awareness to attain higher savings in storage utilization.

For example, when data is deleted by the user, the system administrator can initiate a reclamation process in which the IBM XIV Storage System frees the non-utilized blocks and where these blocks are reclaimed by the available pool of storage.

Instant space reclamation does not support space reclamation for the following objects:

- Mirrored volumes
- Volumes that have snapshots
- Snapshots

Snapshot function and lifecycle

The roles of the snapshot determine its life cycle.

The IBM XIV Storage System uses advanced snapshot mechanisms to create a virtually unlimited number of volume copies without impacting performance. Snapshot taking and management are based on a mechanism of internal pointers that allow the master volume and its snapshots to use a single copy of data for all portions that have not been modified.

This approach, also known as Redirect-on-Write (ROW) is an improvement of the more common Copy-on-Write (COW), which translates into a reduction of I/O actions, and therefore storage usage.

With the IBM XIV snapshots, no storage capacity is consumed by the snapshot until the source volume (or the snapshot) is changed.

Figure 3 on page 13 shows the life cycle of a snapshot.

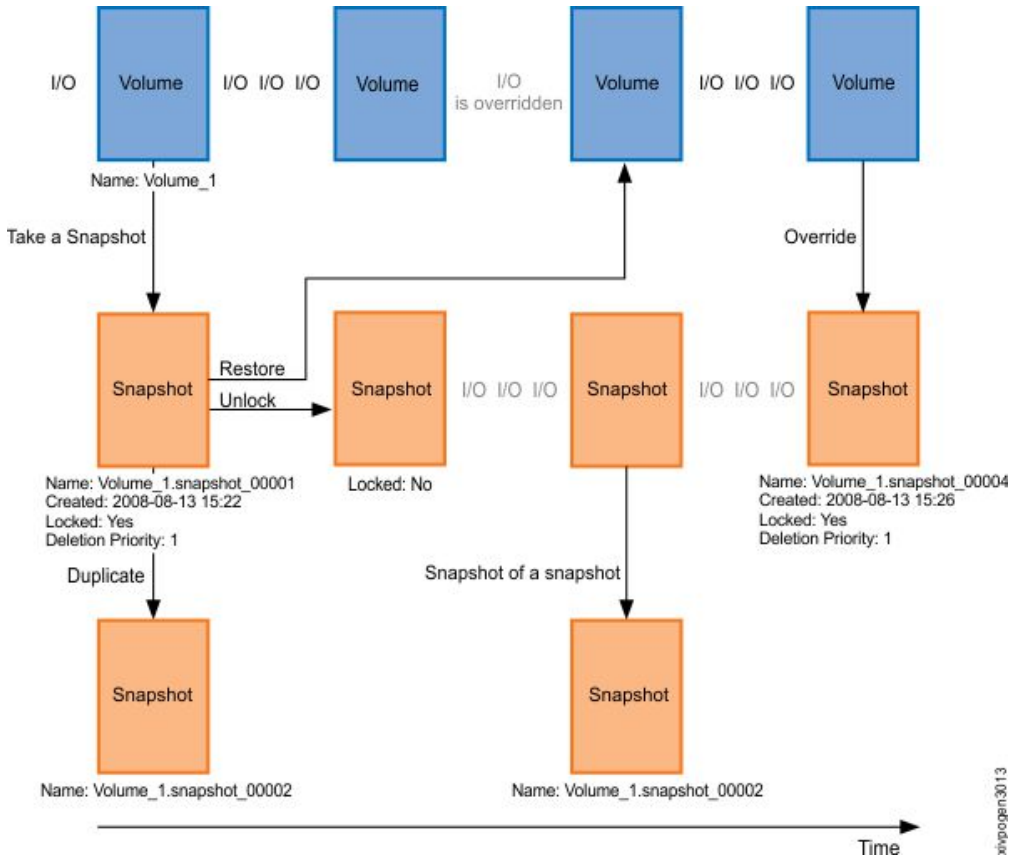


Figure 3. The snapshot life cycle

The following operations are applicable for the snapshot:

Create Creates (takes) the snapshot

Restore
Copies the snapshot back onto the volume. The main snapshot functionality is the capability to restore the volume.

Unlocking
Unlocks the snapshot to make it writable and sets the status to Modified. Re-locking the unlocked snapshot disables further writing, but does not change the status from Modified.

Duplicate
Duplicates the snapshot. Similar to the volume, which can be snapshotted infinitely, the snapshot itself can be duplicated.

A snapshot of a snapshot
Creates a backup of a snapshot that was written into. Taking a snapshot of a writable snapshot is similar to taking a snapshot of a volume.

Overwriting a snapshot
Overwrites a specific snapshot with the content of the volume.

Delete Deletes the snapshot.

Creating a snapshot

First, a snapshot of the volume is taken. The system creates a pointer to the volume, hence the snapshot is considered to have been immediately created. This

is an atomic procedure that is completed in a negligible amount of time. At this point, all data portions that are associated with the volume are also associated with the snapshot.

Later, when a request arrives to read a certain data portion from either the volume or the snapshot, it reads from the same single, physical copy of that data.

Throughout the volume life cycle, the data associated with the volume is continuously modified as part of the ongoing operation of the system. Whenever a request to modify a data portion on the master volume arrives, a copy of the original data is created and associated with the snapshot. Only then the volume is modified. This way, the data originally associated with the volume at the time the snapshot is taken is associated with the snapshot, effectively reflecting the way the data was before the modification.

Locking and unlocking snapshots

Initially, a snapshot is created in a locked state, which prevents it from being changed in any way related to data or size, and only enables the reading of its contents. This is called an *image* or *image snapshot* and represents an exact replica of the master volume when the snapshot was created.

A snapshot can be unlocked after it is created. The first time a snapshot is unlocked, the system initiates an irreversible procedure that puts the snapshot in a state where it acts like a regular volume with respect to all changing operations. Specifically, it allows write requests to the snapshot. This state is immediately set by the system and brands the snapshot with a permanent modified status, even if no modifications were performed. A *modified snapshot* is no longer an image snapshot.

An unlocked snapshot is recognized by the hosts as any other writable volume. It is possible to change the content of unlocked snapshots, however, physical storage space is consumed only for the changes. It is also possible to resize an unlocked snapshot.

Master volumes can also be locked and unlocked. A locked master volume cannot accept write commands from hosts. The size of locked volumes cannot be modified.

Duplicating a snapshot

A user can create a new snapshot by duplicating an existing snapshot. The duplicate is identical to the source snapshot. The new snapshot is associated with the master volume of the existing snapshot, and appears as if it were taken at the exact moment the source snapshot was taken. For image snapshots that have never been unlocked, the duplicate is given the exact same creation date as the original snapshot, rather than the duplication creation date.

With this feature, a user can create two or more identical copies of a snapshot for backup purposes, and perform modification operations on one of them without sacrificing the usage of the snapshot as an untouched backup of the master volume, or the ability to restore from the snapshot.

Creating a snapshot of a snapshot

When duplicating a snapshot that has been changed using the unlock feature, the generated snapshot is actually a snapshot of a snapshot.

The creation time of the newly created snapshot is when the command was issued , and its content reflects the contents of the source snapshot at the moment of creation. After it is created, the new snapshot is viewed as another snapshot of the master volume.

Formatting a snapshot or a snapshot group

This operation deletes the content of a snapshot - or a snapshot group - while maintaining its mapping to the host.

The purpose of the formatting is to allow customers to backup their volumes via snapshots, while maintaining the snapshot ID and the LUN ID. More than a single snapshot can be formatted per volume.

Required reading

Some of the concepts this topic refers to are introduced in this chapter as well as in a later chapter on this document. Consult the following reading list to get a grasp regarding these topics.

Snapshots
"Snapshot function and lifecycle" on page 12

Snapshot groups
"Consistency group snapshot lifecycle" on page 32

Attaching a host
"Host system attachment" on page 40

The format operation results with the following

- The formatted snapshot is read-only
- The format operation has no impact on performance
- The formatted snapshot does not consume space
- Reading from the formatted snapshot always returns zeroes
- It can be overridden
- It can be deleted
- Its deletion priority can be changed

Restrictions

No unlock
The formatted snapshot is read-only and can't be unlocked.

No volume restore
The volume that the formatted snapshot belongs to can't be restored from it.

No restore from another snapshot
The formatted snapshot can't be restored from another snapshot.

No duplicating
The formatted snapshot can't be duplicated.

No re-format
The formatted snapshot can't be formatted again.

No volume copy
The formatted snapshot can't serve as a basis for volume copy.

No resize
The formatted snapshot can't be resized.

Use case

1. Create a snapshot for each LUN you would like to backup to, and mount it to the host.
2. Configure the host to backup this LUN.
3. **Format the snapshot.**
4. Re-snap. The LUN ID, Snapshot ID and mapping are maintained.

Restrictions in relation to other XIV operations

Snapshots of the following types can't be formatted:

Internal snapshot
Formatting an internal snapshot hampers the process it is part of, therefore is forbidden.

Part of a sync job
Formatting a snapshot that is part of a sync job renders the sync job meaningless, therefore is forbidden.

Part of a snapshot group
A snapshot that is part of a snapshot group can't be treated as an individual snapshot.

Snapshot group restrictions
All snapshot format restrictions apply to the snapshot group format operation.

Additional snapshot attributes

Snapshots have the following additional attributes.

Storage utilization

The storage system allocates space for volumes and their snapshots in a way that whenever a snapshot is taken, additional space is actually needed only when the volume is written into.

As long as there is no actual writing into the volume, the snapshot does not need actual space. However, some applications write into the volume whenever a snapshot is taken. This writing into the volume mandates immediate space allocation for this new snapshot. Hence, these applications use space less efficiently than other applications.

Auto-delete priority

Snapshots are associated with an *auto-delete priority* to control the order in which snapshots are automatically deleted.

Taking volume snapshots gradually fills up storage space according to the amount of data that is modified in either the volume or its snapshots. To free up space when the maximum storage capacity is reached, the system can refer to the auto-delete priority to determine the order in which snapshots are deleted. If snapshots have the same priority, the snapshot that was created first is deleted first.

Name and association

A snapshot can either be taken of a source volume, or from a source snapshot.

The name of a snapshot is either automatically assigned by the system at creation time or given as a parameter of the XCLI command that creates it. The snapshot's auto-generated name is derived from its volume's name and a serial number.

The following are examples of snapshot names:

```
MASTERVOL.snapshot_XXXXX
NewDB-server2.snapshot_00597
```

Parameter	Description	Example
MASTERVOL	The name of the volume.	NewDB-server2
XXXXX	A five-digit, zero filled snapshot number.	00597

Redirect-on-Write (ROW)

The IBM XIV Storage System uses the Redirect-on-Write (ROW) mechanism.

The following items are characteristics of using ROW when a write request is directed to the master volume:

1. The data originally associated with the master volume remains in place.
2. The new data is written to a different location on the disk.
3. After the write request is completed and acknowledged, the original data is associated with the snapshot and the newly written data is associated with the master volume.

In contrast with the traditional copy-on-write method, with redirect-on-write the actual data activity involved in taking the snapshot is drastically reduced. Moreover, if the size of the data involved in the write request is equal to the system's slot size, there is no need to copy any data at all. If the write request is smaller than the system's slot size, there is still much less copying than with the standard approach of Copy-on-Write.

In the following example of the Redirect-on-Write process, The volume is displayed with its data and the pointer to this data.

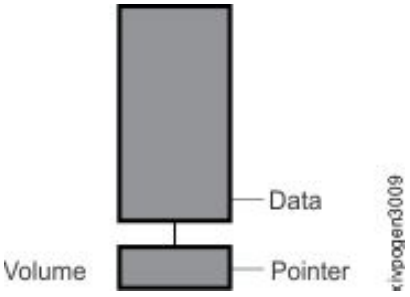


Figure 4. The Redirect-on-Write process: the volume's data and pointer

When a snapshot is taken, a new header is written first.

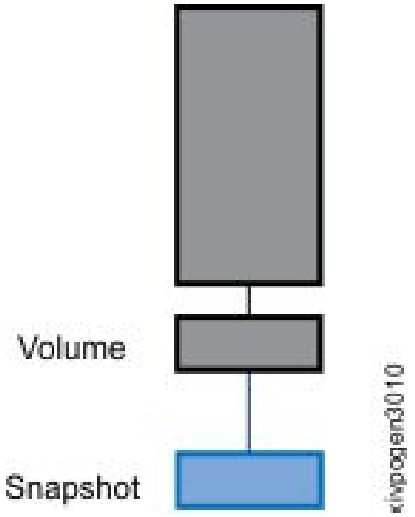


Figure 5. The Redirect-on-Write process: when a snapshot is taken the header is written first

The new data is written anywhere else on the disk, without the need to copy the existing data.

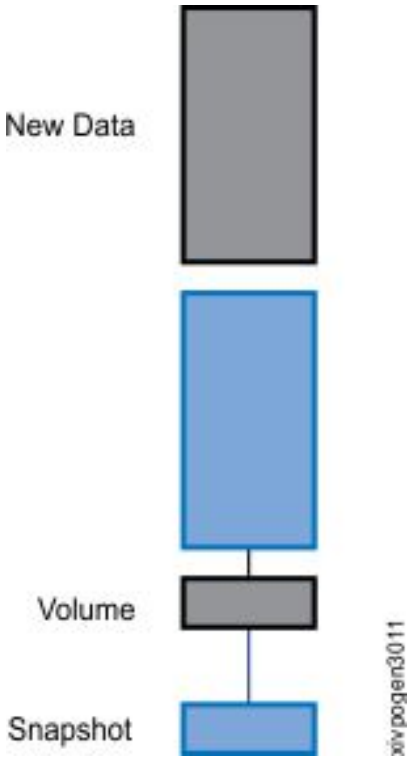


Figure 6. The Redirect-on-Write process: the new data is written

The snapshot points at the old data where the volume points at the new data (the data is regarded as new as it keep updating by I/Os).

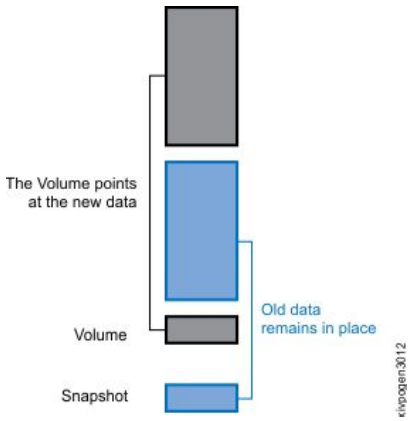


Figure 7. The Redirect-on-Write process: The snapshot points at the old data where the volume points at the new data

The metadata established at the beginning of the snapshot mechanism is independent of the size of the volume to be copied. This approach allows the user to achieve the following important goals:

Continuous backup
As snapshots are taken, backup copies of volumes are produced at frequencies that resemble those of *Continuous Data Protection* (CDP). Instant restoration of volumes to virtually any point in time is easily achieved in case of logical data corruption at both the volume level and the file level.

Productivity
The snapshot mechanism offers an instant and simple method for creating short or long-term copies of a volume for data mining, testing, and external backups.

Full Volume Copy

Full Volume Copy overwrites an existing volume, and at the time of its creation it is logically equivalent to the source volume.

After the copy is made, both volumes are independent of each other. Hosts can write to either one of them without affecting the other. This is somewhat similar to creating a writable (unlocked) snapshot, with the following differences and similarities:

Creation time and availability
Both Full Volume Copy and creating a snapshot happen almost instantly. Both the new snapshot and volume are immediately available to the host. This is because at the time of creation, both the source and the destination of the copy operation contain the exact same data and share the same physical storage.

Singularity of the copy operation
Full Volume Copy is implemented as a single copy operation into an existing volume, overriding its content and potentially its size. The existing target of a volume copy can be mapped to a host. From the host perspective, the content of the volume is changed within a single transaction. In contrast, creating a new writable snapshot creates a new object that has to be mapped to the host.

Space allocation
With Full Volume Copy, all the required space for the target volume is

reserved at the time of the copy. If the storage pool that contains the target volume cannot allocate the required capacity, the operation fails and has no effect. This is unlike writable snapshots, which are different in nature.

Taking snapshots and mirroring the copied volume
The target of the Full Volume Copy is a master volume. This master volume can later be used as a source for taking a snapshot or creating a mirror. However, at the time of the copy, neither snapshots nor remote mirrors of the target volume are allowed.

Redirect-on-write implementation
With both Full Volume Copy and writable snapshots, while one volume is being changed, a redirect-on-write operation will ensure a split so that the other volume maintains the original data.

Performance
Unlike writable snapshots, with Full Volume Copy, the copying process is performed in the background even if no I/O operations are performed. Within a certain amount of time, the two volumes will use different copies of the data, even though they contain the same logical content. This means that the redirect-on-write overhead of writes occur only before the initial copy is complete. After this initial copy, there is no additional overhead.

Availability
Full Volume Copy can be performed with source and target volumes in different storage pools.

Restoring volumes and snapshots

The restoration operation provides the user with the ability to instantly recover the data of a master volume from any of its locked snapshots.

Restoring volumes

A volume can be restored from any of its snapshots, locked and unlocked. Performing the restoration replicates the selected snapshot onto the volume. As a result of this operation, the master volume is an exact replica of the snapshot that restored it. All other snapshots, old and new, are left unchanged and can be used for further restore operations. A volume can even be restored from a snapshot that has been written to. Figure 8 on page 21 shows a volume being restored from three different snapshots.

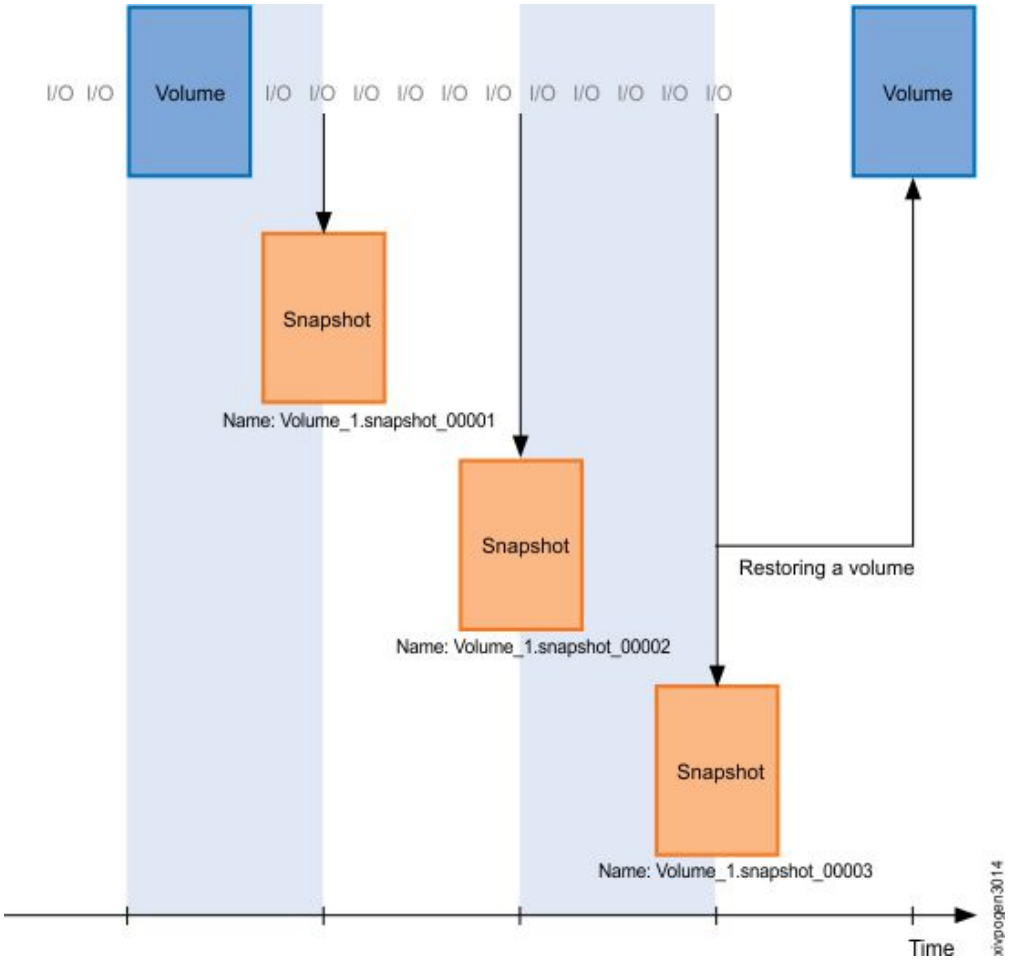


Figure 8. Restoring volumes

Restoring snapshots

The snapshot itself can also be restored from another snapshot. The restored snapshot retains its name and other attributes. From the host perspective, this restored snapshot is considered an instant replacement of all the snapshot content with other content. Figure 9 on page 22 shows a snapshot being restored from two different snapshots.

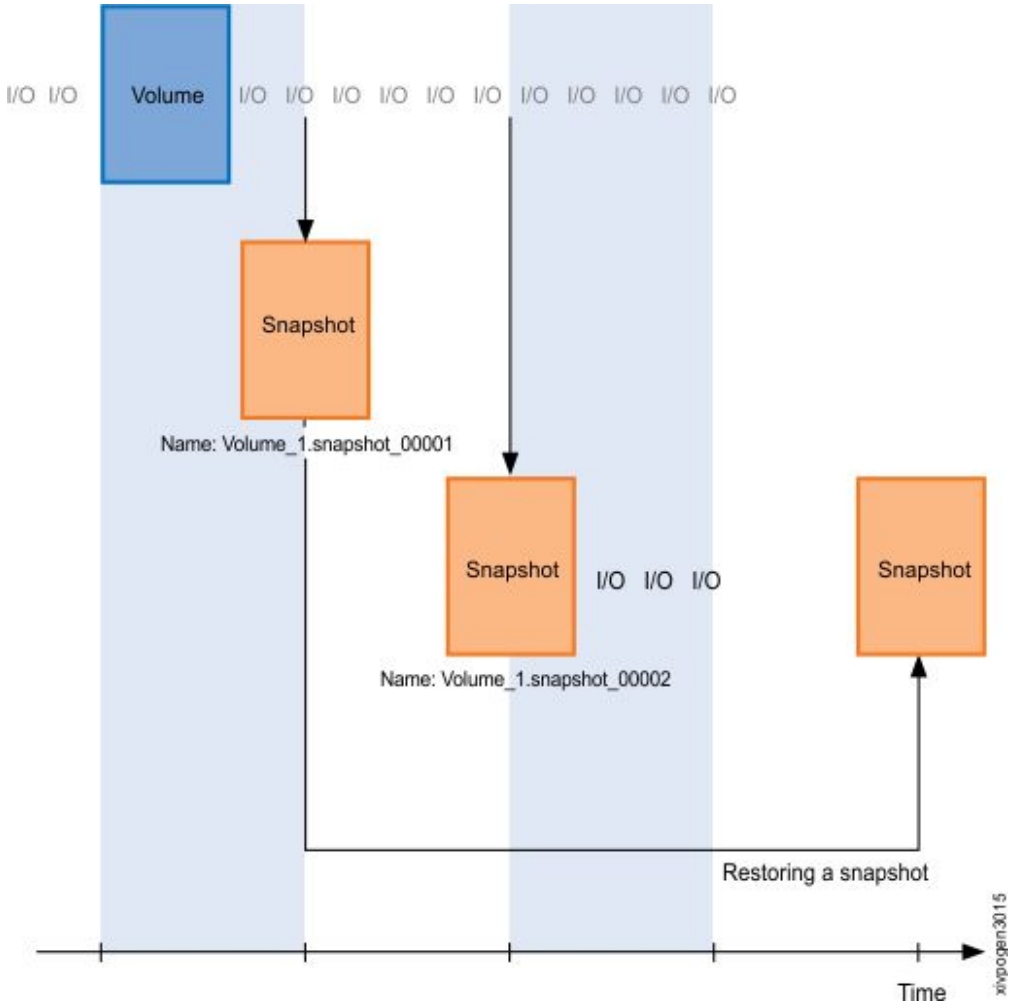


Figure 9. Restoring snapshots

Chapter 3. Storage pools

The storage space of the IBM XIV Storage System is portioned into *storage pools*, where each volume belongs to a specific storage pool.

Storage pools provide the following benefits:

Improved management of storage space
Specific volumes can be grouped together in a storage pool. This enables you to control the allocation of a specific storage space to a specific group of volumes. This storage pool can serve a specific group of applications, or the needs of a specific department.

Improved regulation of storage space
Snapshots can be automatically deleted when the storage capacity that is allocated for snapshots is fully consumed. This automatic deletion is performed independently on each storage pool. Therefore, when the size limit of the storage pool is reached, only the snapshots that reside in the affected storage pool are deleted. For more information, see "Additional snapshot attributes" on page 16.

Facilitating thin provisioning
Thin provisioning is enabled by storage pools.

Storage pools as logical entities

A storage pool is a logical entity and is not associated with a specific disk or module. All storage pools are equally spread over all disks and all modules in the system.

As a result, there are no limitations on the size of storage pools or on the associations between volumes and storage pools. For example:

- The size of a storage pool can be decreased, limited only by the space consumed by the volumes and snapshots in that storage pool.
- Volumes can be moved between storage pools without any limitations, as long as there is enough free space in the target storage pool.

Note: For the size of the storage pool, please refer to the IBM XIV Storage System data sheet.

All of the above transactions are accounting transactions, and do not impose any data copying from one disk drive to another. These transactions are completed instantly.

Moving volumes between storage pools

For a volume to be moved to a specific storage pool, there must be enough room for it to reside there. If a storage pool is not large enough, the storage pool must be resized, or other volumes must be moved out to make room for the new volume.

A volume and all its snapshots always belong to the same storage pool. Moving a volume between storage pools automatically moves all its snapshots together with the volume.

Protecting snapshots at a storage pool level

Snapshots that participate in the mirroring process can be protected in case of pool space depletion.

This is done by attributing both snapshots (or snapshot groups) and the storage pool with a deletion priority. The snapshots are attributed with a deletion priority between 0 - 4 and the storage pool is configured to disregard snapshots whose priority is above a specific value. Snapshots with a lower delete priority (higher number) than the configured value might be deleted by the system whenever the pool space depletion mechanism implies so, thus protecting snapshots with a priority equal or higher than this value.

Thin provisioning

The IBM XIV Storage System supports thin provisioning, which provides the ability to define logical volume sizes that are much larger than the physical capacity installed on the system. Physical capacity needs only to accommodate written data, while parts of the volume that have never been written to do not consume physical space.

This chapter discusses:

- Volume hard and soft sizes
- System hard and soft sizes
- Pool hard and soft sizes
- Depletion of hard capacity

Volume hard and soft sizes

Without thin provisioning, the size of each volume is both seen by the hosts and reserved on physical disks. Using thin provisioning, each volume is associated with the following two sizes:

Hard volume size
This reflects the total size of volume areas that were written by hosts. The hard volume size is not controlled directly by the user and depends only on application behavior. It starts from zero at volume creation or formatting and can reach the volume soft size when the entire volume has been written. Resizing of the volume does not affect the hard volume size.

Soft volume size
This is the logical volume size that is defined during volume creation or resizing operations. This is the size recognized by the hosts and is fully configurable by the user. The soft volume size is the traditional volume size used without thin provisioning.

System hard and soft size

Using thin provisioning, each IBM XIV Storage System is associated with a *hard system size* and *soft system size*. Without thin provisioning, these two are equal to the system's capacity. With thin provisioning, these concepts have the following meaning:

Hard system size
This is the physical disk capacity that was installed. Obviously, the system's hard capacity is an upper limit on the total hard capacity of all

the volumes. The system's hard capacity can only change by installing new hardware components (disks and modules).

Soft system size
This is the total limit on the soft size of all volumes in the system. It can be set to be larger than the hard system size, up to 79TB. The soft system size is a purely logical limit, but should not be set to an arbitrary value. It must be possible to upgrade the system's hard size to be equal to the soft size, otherwise applications can run out of space. This requirement means that enough floor space should be reserved for future system hardware upgrades, and that the cooling and power infrastructure should be able to support these upgrades. Because of the complexity of these issues, the setting of the system's soft size can only be performed by IBM XIV support.

Pool hard and soft sizes

The concept of storage pool is also extended to thin provisioning. When thin provisioning is not used, storage pools are used to define capacity allocation for volumes. The storage pools control if and which snapshots are deleted when there is not enough space.

When thin provisioning is used, each storage pool has a soft pool size and a hard pool size, which are defined and used as follows:

Hard pool size
This is the physical storage capacity allocated to volumes and snapshots in the storage pool. The hard size of the storage pool limits the total of the hard volume sizes of all volumes in the storage pool and the total of all storage consumed by snapshots. Unlike volumes, the hard pool size is fully configured by the user.

Soft pool size
This is the limit on the total soft sizes of all the volumes in the storage pool. The soft pool size has no effect on snapshots.

Thin provisioning is managed for each storage pool independently. Each storage pool has its own soft size and hard size. Resources are allocated to volumes within this storage pool without any limitations imposed by other storage pools. This is a natural extension of the snapshot deletion mechanism, which is applied even without thin provisioning. Each storage pool has its own space, and snapshots within each storage pool are deleted when the storage pool runs out of space regardless of the situation in other storage pools.

The sum of all the soft sizes of all the storage pools is always the same as the system's soft size and the same applies to the hard size.

Storage pools provide a logical way to allocate storage resources per application or per groups of applications. With thin provisioning, this feature can be used to manage both the soft capacity and the hard capacity.

Depletion of hard capacity

Thin provisioning creates the potential risk of depleting the physical capacity. If a specific system has a hard size that is smaller than the soft size, the system will run out of capacity when applications write to all the storage space that is mapped to hosts. In such situations, the system behaves as follows:

Snapshot deletion
Snapshots are deleted to provide more physical space for volumes. The snapshot deletion is based on the deletion priority and creation time.

Volume locking
If all snapshots have been deleted and more physical capacity is still required, **all the volumes in the storage pool are locked and no write commands are allowed**. This halts any additional consumption of hard capacity.

Note: Space that is allocated to volumes that is unused (that is, the difference between the volume's soft and hard size) can be used by snapshots in the same storage pool.

The thin provisioning implementation in the IBM XIV Storage System manages space allocation per storage pool. Therefore, one storage pool cannot affect another storage pool. This scheme has the following advantages and disadvantages:

Storage pools are independent
Storage pools are independent in respect to the aspect of thin provisioning. **Thin provisioning volume locking on one storage pool does not create a problem in another storage pool.**

Space cannot be reused across storage pools
Even if a storage pool has free space, this free space is never reused for another storage pool. This creates a situation where volumes are locked due to the depletion of hard capacity in one storage pool, while there is available capacity in another storage pool.

Important: If a storage pool runs out of hard capacity, all of its volumes are locked to all write commands. Although write commands that overwrite existing data can be technically serviced, they are blocked to ensure consistency.

Instant space reclamation

The IBM XIV Storage System instant space reclamation continuously recycles reusable IBM XIV storage space that is released by the host operating system, without any performance or management impact, and with measurable results.

Using instant space reclamation, storage and host administrators increase their systems capacity use and reduce the need for thin provisioning. Upon notification from the host, the IBM XIV frees any space that is no longer in use by writing zeroes into it. See more here: “Writing zeroes” on page 46.

Communicating with the host in order to determine whether an allocated space is not in use evolves the following:

- Getting the allocation status from the host
- Matching this status with provisioning thresholds and reporting the findings
- Detecting space that is suitable for reclamation
- Freeing this space

The instant space reclamation feature skips volumes with temporary functionality and relations, such as:

- Off-line initialization of asynchronous mirroring
- Data migration of all kinds

The support of instant space reclamation for a mirrored pair of volumes is limited to synchronous mirroring and provided that both systems support the feature (that is, both systems are of version 11.2.0 and up).

Supported platforms

The following vendors have announced that by the end of 2012 their OS will support instant space reclamation:

- Microsoft Windows Server 8
- VMware
- RedHat
- Symantec SSF

Activating the instant space reclamation feature

Instant space reclamation can be globally enabled on manufacturing the IBM XIV Storage System, and disabled - and further enabled - by a technician.

Chapter 4. Consistency groups

A consistency group is a group of volumes of which a snapshot can be made at the same point in time, therefore ensuring a consistent image of all volumes within the group at that time.

The concept of a consistency group is common among storage systems in which it is necessary to perform concurrent operations collectively across a set of volumes so that the result of the operation preserves the consistency among volumes. For example, effective storage management activities for applications that span multiple volumes, or creating point-in-time backups, is not possible without first employing consistency groups.

The consistency between the volumes in the group is important for maintaining data integrity from the application perspective. By first grouping the application volumes into a consistency group, it is possible to later capture a consistent state of all volumes within that group at a specified point-in-time using a special snapshot command for consistency groups.

Consistency groups can be used to take simultaneous snapshots of multiple volumes, thus ensuring consistent copies of a group of volumes. Creating a synchronized snapshot set is especially important for applications that use multiple volumes concurrently. A typical example is a database application, where the database and the transaction logs reside on different storage volumes, but all of their snapshots must be taken at the same point in time.

A consistency group is also an administrative unit that facilitates simultaneous snapshots of multiple volumes, mirroring of volume groups, and administration of volume sets.

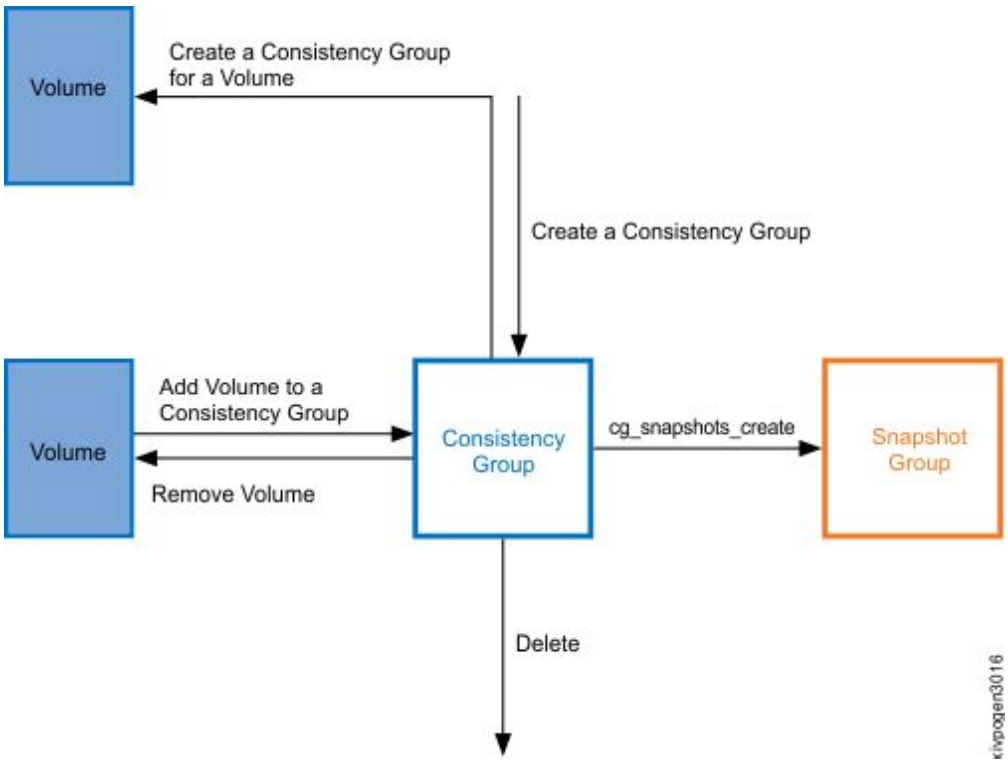


Figure 10. Consistency group creation and options

All volumes in a consistency group must belong to the same storage pool.

Snapshot of a consistency group

Taking a snapshot for the entire consistency group means that a snapshot is taken for each volume of the consistency group at the same point-in-time. These snapshots are grouped together to represent the volumes of the consistency group at a specific point in time.

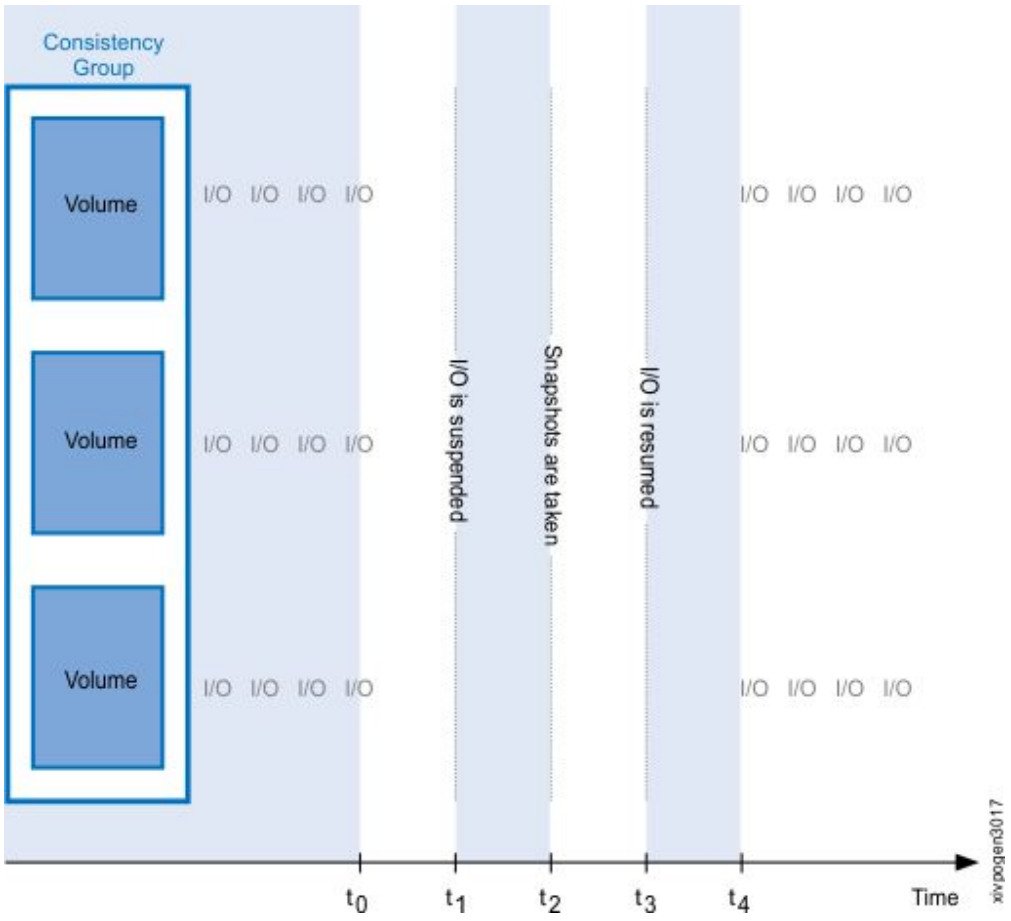


Figure 11. A snapshot is taken for each volume of the consistency group

In Figure 11, a snapshot is taken for each of the consistency group's volumes in the following order:

Time = t_0
Prior to taking the snapshots, all volumes in the consistency group are active and being read from and written to.

Time = t_1
When the command to snapshot the consistency group is issued, I/O is suspended .

Time = t_2
Snapshots are taken at the same point in time.

Time = t_3
I/O is resumed and the volumes continue their normal work.

Time = t_4
After the snapshots are taken, the volumes resume active state and continue to be read from and written to.

Most snapshot operations can be applied to each snapshot in a grouping, known as a *snapshot set*. The following items are characteristics of a snapshot set:

- A snapshot set can be locked or unlocked. When you lock or unlock a snapshot set, all snapshots in the set are locked or unlocked.

- A snapshot set can be duplicated.
- A snapshot set can be deleted. When a snapshot set is deleted, all snapshots in the set are also deleted.

A snapshot set can be disbanded which makes all the snapshots in the set independent snapshots that can be handled individually. The snapshot set itself is deleted, but the individual snapshots are not.

Consistency group snapshot lifecycle

Most snapshot operations can be applied to snapshot groups, where the operation affects every snapshot in the group.

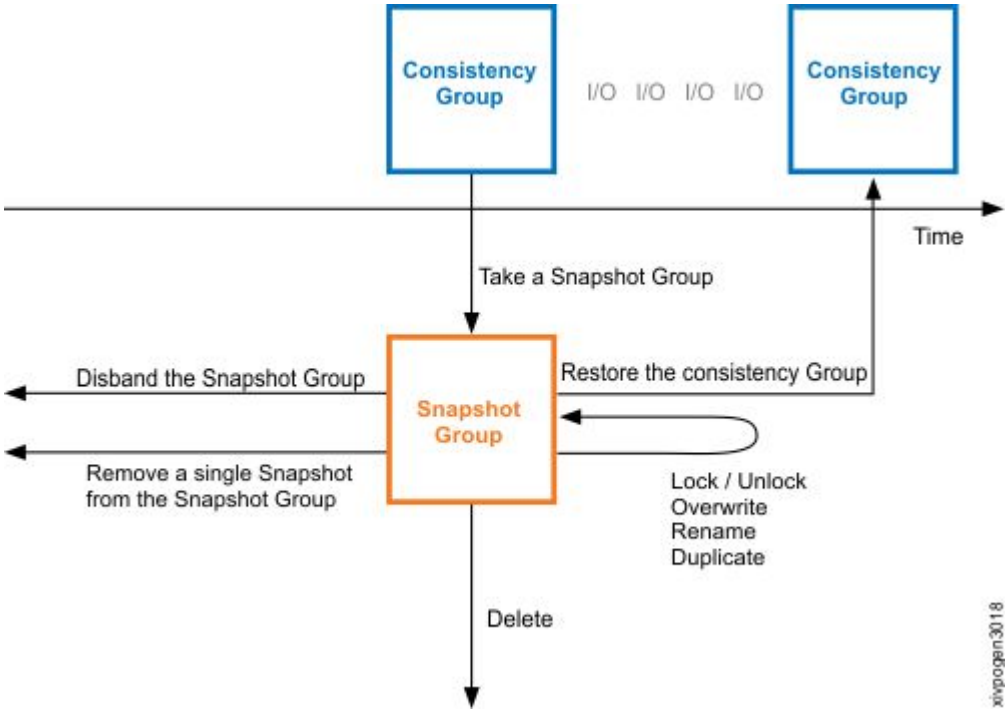


Figure 12. Most snapshot operations can be applied to snapshot groups

Taking a snapshot group
Creates a snapshot group. .

Restoring consistency group from a snapshot group
The main purpose of the snapshot group is the ability to restore the entire consistency group at once, ensuring that all volumes are synchronized to the same point in time.

Restoring a consistency group is a single action in which every volume that belongs to the consistency group is restored from a corresponding snapshot that belongs to an associated snapshot group.

Not only does the snapshot group have a matching snapshot for each of the volumes, all of the snapshots have the same time stamp. This implies that the restored consistency group contains a consistent picture of its volumes as they were at a specific point in time.

Note: A consistency group can only be restored from a snapshot group that has a snapshot for each of the volumes. If either the consistency group or the snapshot group has changed after the snapshot group is taken, the restore action does not work.

Listing a snapshot group
This command lists snapshot groups with their consistency groups and the time the snapshots were taken.

Note: All snapshots within a snapshot group are taken at the same time.

Lock and unlock
Similar to unlocking and locking an individual snapshot, the snapshot group can be rendered writable, and then be written to. A snapshot group that is unlocked cannot be further used for restoring the consistency group, even if it is locked again.

The snapshot group can be locked again. At this stage, it cannot be used to restore the master consistency group. In this situation, the snapshot group functions like a consistency group of its own.

Overwrite
The snapshot group can be overwritten by another snapshot group.

Rename
The snapshot group can be renamed.

Restricted names
Do not prefix the snapshot group's name with any of the following:

1. most_recent
2. last_replicated

Duplicate
The snapshot group can be duplicated, thus creating another snapshot group for the same consistency group with the time stamp of the first snapshot group.

Disbanding a snapshot group
The snapshots that comprise the snapshot group are each related to its volume. Although the snapshot group can be rendered inappropriate for restoring the consistency group, the snapshots that comprise it are still attached to their volumes. Disbanding the snapshot group detaches all snapshots from this snapshot group but maintains their individual connections to their volumes. These individual snapshots cannot restore the consistency group, but they can restore its volumes individually.

Changing the snapshot group deletion priority
Manually sets the deletion priority of the snapshot group.

Deleting the snapshot group
Deletes the snapshot group along with its snapshots.

Chapter 5. QoS performance classes

The Quality of Service (QoS) feature allows the IBM XIV Storage System to deliver different service levels to hosts that are connected to the same XIV system.

The QoS feature favors performance of critical business applications that run concurrently with noncritical applications. Because the XIV disk and cache are shared among all applications and all hosts are attached to the same resources, division of these resources among both critical and noncritical applications might have an unintended adverse performance effect on critical applications. QoS can address this by limiting the rate, based on bandwidth and IOPS, for non-critical applications. Limiting performance resources for non-critical applications means that the remaining resources are available without limitation for the business-critical applications.

The QoS feature is managed through the definition of performance classes and then associating hosts with a performance class. The feature was extended in the XIV Storage Software Version 11.5 and can also be set by XIV domains and XIV storage pools. Each performance class is now implicitly one of two types: host type or pool/domain type.

The QoS feature possibilities and limitations can be summarized as follows:

- Up to 500 performance classes are configurable.
- QoS is applicable to host, domain, pool and restricted combinations of these entities. For instance, hosts cannot be specified for a performance class that already contains a domain or pool
- Limits can be defined as *Total*, meaning for XIV system as a whole, or *Per Interface*.
- Limits are specified as IOPS or bandwidth.
- Limit calculation is based on preferred practices for setup and zoning.

  The limited I/O processes are expected to always come through all active XIV interface nodes (equal to active interface modules). For example, on a 9-module partial rack XIV, where 4 interface modules are active, the total I/O or bandwidth rate would be divided by 4 (the number active interface modules). If a limit total of 3,000 I/Os is specified, it would result to a limitation of 750 I/Os per interface module.

  In addition, in the case of the 9-module XIV, if the limited I/Os are coming through only two of the four interface modules (as a result of the SAN zoning), the effective limitation will be 2 x 750 I/Os = 1,500 I/Os rather than the expected 3,000 I/O limitation.

Note: If more than one host, domain, or pool is added to a performance class, all hosts, domains, or pools in this performance class share the limitations defined on that performance class. For example, if two or more entities are added to a 10,000 IOPS performance class, the total number of all contained entities IOPS is limited to 10,000. Therefore, it is a good practice to create one performance class per domain and one performance class per pool.

Max bandwidth limit attribute

The host rate limitation group has a max bandwidth limit attribute, which is the number of blocks per second. This number could be either:

- A value between *min_rate_limit_bandwidth_blocks_per_sec* and *max_rate_limit_bandwidth_blocks_per_sec* (both are available from the storage system's configuration).
- Zero (0) for unlimited bandwidth.

Chapter 6. Connectivity with hosts

The storage system connectivity is provided through the following interfaces:

- Fibre Channel for host-based I/O
- Gigabit Ethernet for host-based I/O using the iSCSI protocol
- Gigabit Ethernet for management (GUI or CLI) connectivity
- Remote access interfaces:
  - Call-home connection - connecting the IBM XIV Storage System to an IBM trouble-ticketing system.
  - Modem - for incoming calls only. The customer has to provide telephone line and number. The modem provides secondary means for providing remote access for IBM Support.

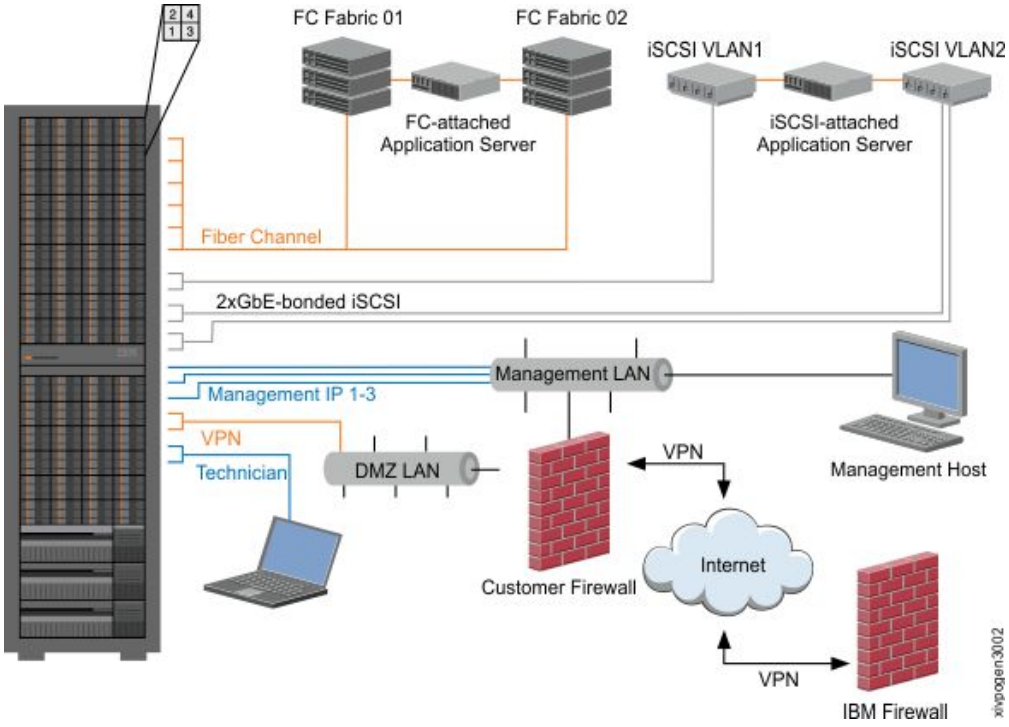


Figure 13. The IBM XIV Storage System interfaces

The following subsections provide information about different connectivity aspects.

IP and Ethernet connectivity

The following topics provide a basic explanation of the various Ethernet ports and IP interfaces that can be defined and various configurations that are possible within the IBM XIV Storage System.

The IBM XIV Storage System IP connectivity provides:

- iSCSI services over IP or Ethernet networks
- Management communication

Ethernet ports

The following three types of Ethernet ports are available:

iSCSI service ports
These ports are used for iSCSI over IP or Ethernet services. A fully equipped rack is configured with six Ethernet ports for iSCSI service. These ports should connect to the user's IP network and provide connectivity to the iSCSI hosts. The iSCSI ports can also accept management connections.

Management ports
These ports are dedicated for IBM XIV command-line interface (XCLI) and IBM XIV Storage Management GUI communications, as well as being used for outgoing SNMP and SMTP connections. A fully equipped rack contains three management ports.

Field technician ports
These ports are used for incoming management traffic only (usage is both XCLI and IBM XIV Storage Management GUI access). The ports are utilized only for the field technician's laptop computer and must not be connected to the user's IP network.

IPv6 certification

The IBM XIV Storage System supports IPv6 and IPSec technology adoption as described in this topic.

The IBM XIV Storage System supports IPv6 through stateless autoconfiguration and full IPSec (IKE2, transport, and tunnel mode) for Management and VPN ports.

Not supported

- There is no IPv6 support for technician notebook port.
- iSCSI ports are not supported.

Enabling and disabling IPv6

The IBM XIV Storage System supports IPv4 and IPv6 addresses out of the box. As the feature is enabled, stateless autoconfiguration is automatically enabled as well, and the system interfaces are getting ready to work with IPv6. Thus, looking for DNS addresses, the system also looks for AAAA entries.

Programs that are using connections on the Management and VPN ports must support IPv6 addresses. Each IP interface in the system may now have several IP addresses: static IPv4 address, static IPv6 address, and the stateless configuration link and site local IPv6 addresses. Where multiple IPv6 static addresses are assigned for each interface, the system supports only one address per interface.

The IPv6 addressing feature can be disabled if wanted.

Management connectivity

Management connectivity is used for the following functions:

- Executing XCLI commands through the IBM XIV command-line interface (XCLI)
- Controlling the IBM XIV Storage System through the IBM XIV Storage Management GUI
- Sending e-mail notification messages and SNMP traps about event alerts

To ensure management redundancy in case of module failure, the IBM XIV Storage System management function is accessible from three different IP addresses. Each of the three IP addresses is handled by a different hardware module. The various IP addresses are transparent to the user and management functions can be performed through any of the IP addresses. These addresses can be accessed simultaneously by multiple clients. Users only need to configure the IBM XIV Storage Management GUI or XCLI for the set of IP addresses that are defined for the specific system.

Note: All management IP interfaces must be connected to the same subnet and use the same network mask, gateway, and MTU.

XCLI and IBM XIV Storage Management GUI management

The IBM XIV Storage System management connectivity system allows users to manage the system from both the XCLI and IBM XIV Storage Management GUI. Accordingly, both can be configured to manage the system through iSCSI IP interfaces. Both XCLI and IBM XIV Storage Management GUI management is run over TCP port 7778. With all traffic encrypted through the Secure Sockets Layer (SSL) protocol.

System-initiated IP communication

The IBM XIV Storage System can also initiate IP communications to send event alerts as necessary. Two types of system-initiated IP communications exist:

Sending e-mail notifications through the SMTP protocol
E-mails are used for both e-mail notifications and for SMS notifications through the SMTP to SMS gateways.

Sending SNMP traps

Note: SMPT and SNMP communications can be initiated from any of the three IP addresses. This is different from XCLI and IBM XIV Storage Management GUI, which are user initiated. Accordingly, it is important to configure all three IP interfaces and to verify that they have network connectivity.

Field technician ports

The IBM XIV Storage System supports two Ethernet ports. These ports are dedicated for the following reasons:

- Field technician use
- Initial system configuration
- Direct connection for service staff when they can not connect to customer network
- Directly manage the IBM XIV Storage System through a laptop computer

Laptop connectivity - configuring using DHCP

Two field technician ports are provided for redundancy purposes. This ensures that field technicians will always be able to connect a laptop to the IBM XIV Storage System. These two ports use a Dynamic Host Configuration Protocol (DHCP) server. The DHCP server will automatically assign IP addresses to the user's laptop and connects the laptop to the IBM XIV Storage System network. A laptop connected to any of the field technician ports is assigned an IP address and the

IBM XIV Storage Management GUI or IBM XIV command-line interface (XCLI) will typically use the predefined configuration *direct-technician-port*.

Note: The two field technician laptop ports are used only to connect directly to the IBM XIV Storage System and should never be connected to the customer's network.

Laptop connectivity - configuring without DHCP

If the technician's laptop is not setup to receive automatic IP configuration information through DHCP, the laptop should be defined using these parameters:

IP address:
14.10.202.1

Netmask:
255.255.255.0

Gateway:
none

MTU:
1536

The field technician ports accept both XCLI and IBM XIV Storage Management GUI communications. SNMP and SMTP alerts are not sent through these ports.

Note: Each of the field technician ports is connected to a different module. Therefore, if a module fails, the port will become inoperative. When this happens, the laptop should be connected to the second port.

Configuration guidelines summary

When shipped, the IBM XIV Storage System does not have any IP management configurations. Accordingly, the following procedures should be performed when first setting up the system:

- Connecting a laptop to one of the field technician laptop ports on the patch panel
- Configuring at least one management IP interface
- Continuing the configuration process from either the technician port or from the configured IP interface

Note: It is important to define all three management IP interfaces and to test outgoing SNMP and SMTP connections from all three interfaces.

Host system attachment

The IBM XIV Storage System attaches to hosts of various operating systems.

The IBM XIV Storage System attaches to hosts through a set of Host Attachment Kits and complementary utilities.

Note: The term *host system attachment* was previously known as *host connectivity* or *mapping*. These terms are obsolete.

Balanced traffic without a single point of failure

All host traffic (I/O) is served through up to six interface modules (modules 4-9). Although the IBM XIV Storage System distributes the traffic across all system modules, the storage administrator is responsible for ensuring that host I/O operations are equally distributed among the different interface modules.

The workload balance should be watched and reviewed when host traffic patterns change. The IBM XIV Storage System does not automatically balance incoming host traffic. The storage administrator is responsible for ensuring that host connections are made redundantly in such a way that a single failure, such as in a module or HBA, will not cause all paths to the machine to fail. In addition, the storage administrator is responsible for making sure the host workload is adequately spread across the different connections and interface modules.

Dynamic rate adaptation

The IBM XIV Storage System provides a mechanism for handling insufficient bandwidth and external connections for the mirroring process.

The mirroring process replicates a local site on a remote site (See the Chapter 8, "Synchronous remote mirroring," on page 57 and Chapter 9, "Asynchronous remote mirroring," on page 75 chapters later on this document). To accomplish this, the process depends on the availability of bandwidth between the local and remote storage systems.

The mirroring process' sync rate attribute determines the bandwidth that is required for a successful mirroring. Manually configuring this attribute, the user takes into account the availability of bandwidth for the mirroring process, where the IBM XIV Storage System adjusts itself to the available bandwidth. Moreover, in some cases the bandwidth is sufficient, but external I/Os latency causes the mirroring process to fall behind incoming I/Os, thus to repeat replication jobs that were already carried out, and eventually to under-utilize the available bandwidth even if it was adequately allocated.

The IBM XIV Storage System prevents I/O timeouts through continuously measuring the I/O latency. Excess incoming I/Os are pre-queued until they can be submitted. The mirroring rate dynamically adapts to the number of pre-queued incoming I/Os, allowing for a smooth operation of the mirroring process.

Attaching volumes to hosts

While the IBM XIV Storage System identifies volumes and snapshots by name, hosts identify volumes and snapshots according to their logical unit number (LUN).

A *LUN* is an integer that is used when attaching a system's volume to a registered host. Each host can access some or all of the volumes and snapshots on the storage system, up to a set maximum. Each accessed volume or snapshot is identified by the host through a LUN.

For each host, a LUN identifies a single volume or snapshot. However, different hosts can use the same LUN to access different volumes or snapshots.

Excluding LUN0

Do not use LUN 0 as a normal LUN.

LUN0 can be mapped to a volume just like other LUNs. However, when no volume is mapped to LUN0, the HAK is using it to discover the LUN array. Hence, we recommend not to use LUN0 as a normal LUN.

Advanced host attachment

The IBM XIV Storage System provides flexible host attachment options.

The following host attachment options are available:

- Definition of different volume mappings for different ports on the same host
- Support for hosts that have both Fibre Channel and iSCSI ports. Although it is not advisable to use these two protocols together to access the same volume, a dual configuration can be useful in the following cases:
  - As a way to smoothly migrate a host from Fibre Channel to iSCSI
  - As a way to access different volumes from the same host, but through different protocols

CHAP authentication of iSCSI hosts

The MS-CHAP extension enables authentication of initiators (hosts) to the IBM XIV Storage System and vice versa in unsecured environments.

When CHAP support is enabled, hosts are securely authenticated by the IBM XIV Storage System. This increases overall system security by verifying that only authenticated parties are involved in host-storage interactions.

Definitions

The following definitions apply to authentication procedures:

CHAP Challenge Handshake Authentication Protocol

CHAP authentication
An authentication process of an iSCSI initiator by a target through comparing a secret hash that the initiator submits with a computed hash of that initiator's secret which is stored on the target.

Initiator
The host.

Oneway (unidirectional CHAP)
CHAP authentication where initiators are authenticated by the target, but not vice versa.

Supported configurations

CHAP authentication type
Oneway (unidirectional) authentication mode, meaning that the Initiator (host) has to be authenticated by the IBM XIV Storage System.

MDS CHAP authentication utilizes the MDS hashing algorithm.

Access scope
CHAP-authenticated Initiators are granted access to the IBM XIV Storage System via mapping that may restrict access to some volumes.

Authentication modes

The IBM XIV Storage System supports the following authentication modes:

None (default)
In this mode, an initiator is not authenticated by the IBM XIV Storage System.

CHAP (one way)
In this mode, an initiator is authenticated by the IBM XIV Storage System based on the pertinent initiator's submitted hash, which is compared to the hash computed from the initiator's secret stored on the IBM XIV storage system.

Changing the authentication mode from None to CHAP requires an authentication of the host. Changing the mode from CHAP to None doesn't require an authentication.

Complying with RFC 3720

The IBM XIV storage system CHAP authentication complies with the CHAP requirements as stated in RFC 3720. on the following Web site:http://tools.ietf.org/html/rfc3720

Secret length
The secret has to be between 96 bits and 128 bits; otherwise, the system fails the command, responding that the requirements are not fulfilled.

Initiator secret uniqueness
Upon defining or updating an initiator (host) secret, the system compares the entered secret's hash with existing secrets stored by the system and determines whether the secret is unique. If it is not unique, the system presents a warning to the user, but does not prevent the command from completing successfully.

Clustering hosts into LUN maps

To enhance the management of hosts, the IBM XIV Storage System allows clustering them together, where the clustered hosts are provided with identical mappings. The mapping of volumes to LUN identifiers is defined per cluster and applies to all of the hosts in the cluster.

Adding and removing hosts to a cluster are done as follows:

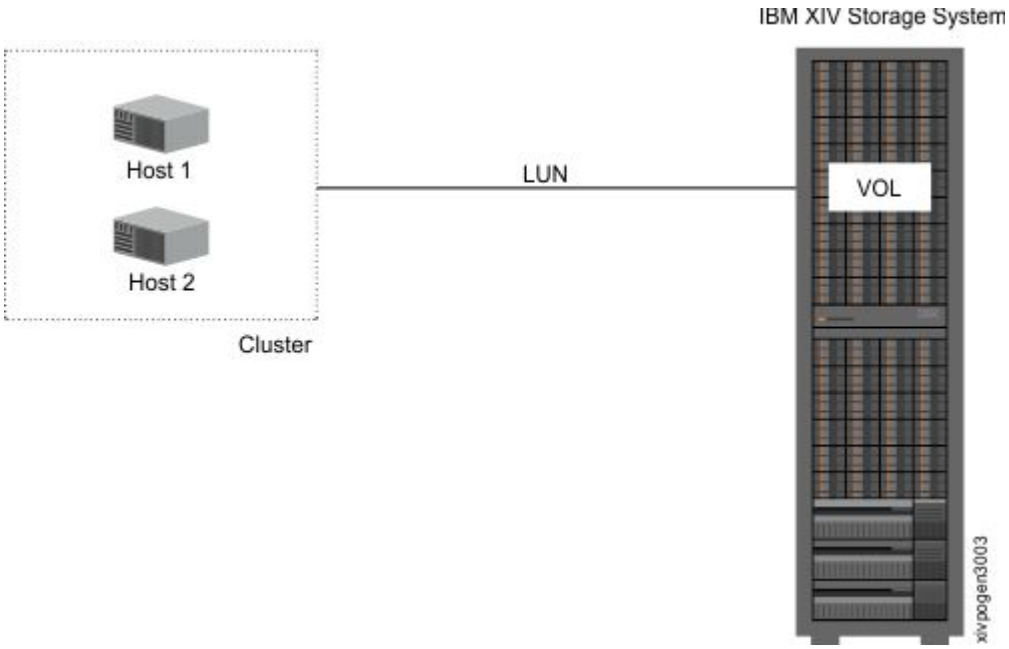


Figure 14. A volume, a LUN and clustered hosts

Adding a host to a cluster

Adding a host to a cluster is a straightforward action in which a host is added to a cluster and is connected to a LUN:

- Changing the host's mapping to the cluster's mapping.
- Changing the cluster's mapping to be identical to the mapping of the newly added host.

Removing a host from a cluster

The host is disbanded from the cluster, maintaining its connection to the LUN:

- The host's mapping remains identical to the mapping of the cluster.
- The mapping definitions do not revert to the host's original mapping (the mapping that was in effect before the host was added to the cluster).
- The host's mapping can be changed.

Notes:

- The IBM XIV Storage System defines the same mapping to all of the hosts of the same cluster. No hierarchy of clusters is maintained.
- Mapping a volume to a LUN that is already mapped to a volume.

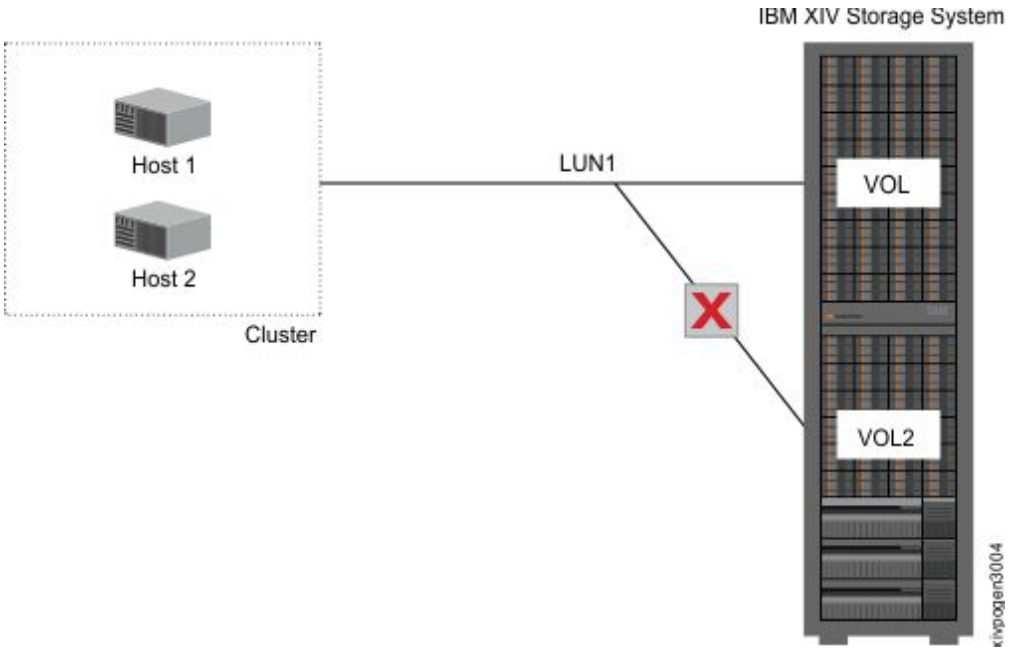


Figure 15. You cannot map a volume to a LUN that is already mapped

- Mapping an already mapped volume to another LUN.

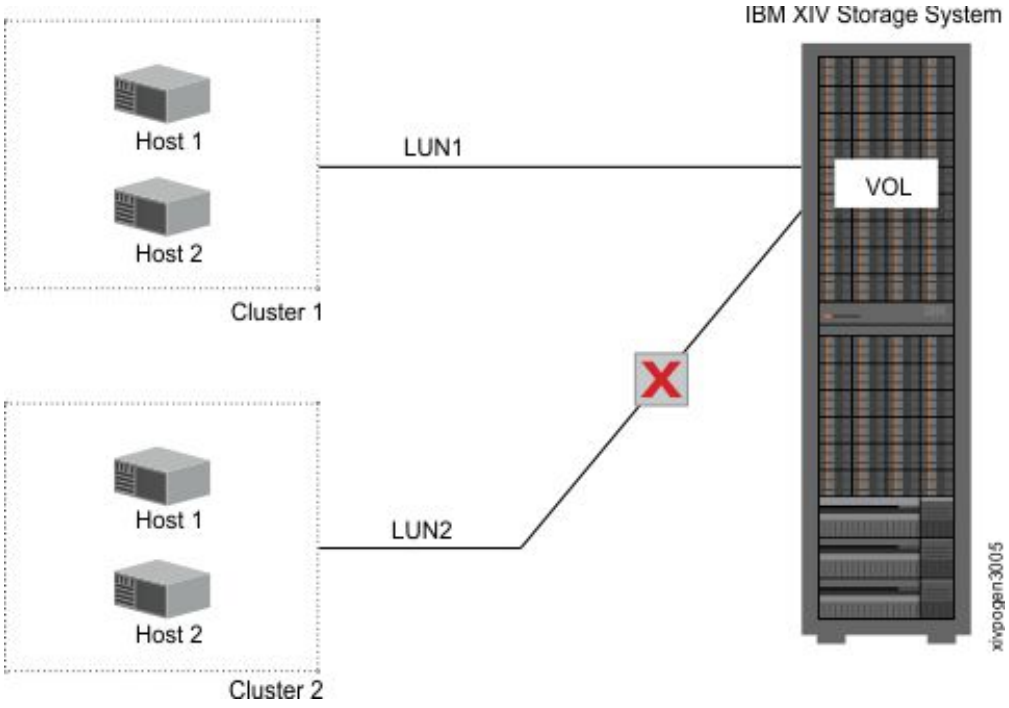


Figure 16. You cannot map a volume to a LUN, if the volume is already mapped.

Volume mapping exceptions

The IBM XIV Storage System facilitates association of cluster mappings to a host that is added to a cluster. The system also facilitates easy specification of mapping exceptions for such host; such exceptions are warranted to accommodate cases where a host must have a mapping that is not defined for the cluster (e.g., Boot from SAN).

Mapping a volume to a host within a cluster
It is impossible to map a volume or a LUN that are already mapped.

For example, the host *host1* belongs to the cluster *cluster1* which has a mapping for the volume *vol1* to *lun1*:

1. Mapping *host1* to *vol1* and *lun1* **fails** as both volume and LUN are already mapped.
2. Mapping *host1* to *vol2* and *lun1* **fails** as the LUN is already mapped.
3. Mapping *host1* to *vol1* and *lun2* **fails** as the volume is already mapped.
4. Mapping *host1* to *vol2* and *lun2* **succeeds** with a warning that the mapping is host-specific.

Listing volumes that are mapped to a host/cluster
Mapped hosts that are part of a cluster are listed (that is, the list is at a host-level rather than cluster-level).

Listing mappings
For each host, the list indicates whether it belongs to a cluster.

Adding a host to a cluster
Previous mappings of the host are removed, reflecting the fact that the only relevant mapping to the host is the cluster's.

Removing a host from a cluster
The host regains its previous mappings.

Supporting VMware extended operations

The IBM XIV Storage System supports VMware extended operations that were introduced in VMware ESX Server 4 (VMware vStorage API).

The purpose of the VMware extended operations is to offload operations from the VMware server onto the storage system. The IBM XIV Storage System supports the following operations:

Full copy
The ability to copy data from one storage array to another without writing to the ESX server.

Block zeroing
Zeroing-out a block as a means for freeing it and make it available for provisioning.

Hardware-assisted locking
Allowing for locking volumes within an atomic command.

Writing zeroes

The Write Zeroes command allows for zeroing large storage areas without sending the zeroes themselves.

Whenever an new VM is created, the ESXi server creates a huge file full of zeroes and sends it to the storage system. The Write Zeroes command is a way to tell a storage controller to zero large storage areas without sending the zeroes. To meet this goal, both VMware's generic driver and our own plug-in utilizes the WRITE SAME 16 command.

This method differs from the former method where the host used to write and send a huge file full of zeroes.

Note: The write zeroes operation is not a thin provisioning operation, as its purpose is not to allocate storage space.

Hardware-assisted locking

The hardware-assisted locking feature utilizes VMware Compare and Write command for reading and writing the volume's metadata within a single operation.

Upon the replacement of SCSI2 reservations mechanism with Compare and Write by VMware, the IBM XIV Storage System provides a faster way to change the metadata specific file, along with eliminating the necessity to lock all of the files during the metadata change.

The legacy VMware SCSI2 reservations mechanism is utilized whenever the VM server performs a management operation, that is to handle the volume's metadata. This method has several disadvantages, among them the mandatory overall lock of access to all volumes, which implies that all other servers are refrained from accessing their own files. In addition, the SCSI2 reservations mechanism entails performing at least four SCSi operations (reserve, read, write, release) in order to get the lock.

The introduction of the new SCSi command, called Compare and Write (SBC-3, revision 22), results with a faster mechanism that is displayed to the volume as an atomic action that does not require to lock any other volume.

Note: The IBM XIV Storage System supports single-block Compare and Write commands only. This restriction is carried out in accordance with VMware.

Backwards compatibility

The IBM XIV Storage System maintains its compatibility with older ESX versions as follows:

- Each volume is capable of connecting legacy hosts, as it still supports SCSi reservations.
- Whenever a volume is blocked by the legacy SCSi reservations mechanism, it is not available for an arriving COMPARE AND WRITE command.
- The Admin is expected to phase out legacy VM servers to fully benefit from the performance improvement rendered by the hardware-assisted locking feature.

Fast copy

The Fast Copy functionality allows for VN cloning on the storage system without going through the ESX server.

The Fast copy functionality speeds up the VM cloning operation by copying data inside the storage system, rather than issuing READ and WRITE requests from the host. This implementation provide a great improvement in performance, since it saves host to storage system intra-storage system communication. Instead, the functionality utilizes the huge bandwidth within the storage system.

Chapter 7. IBM Real-time Compression with XIV

This section introduces the IBM® Real-time Compression™ (RtC) feature of IBM XIV storage systems.

IBM XIV storage system implementation of IBM RtC is a software-only feature that leverages the original XIV hardware design. IBM RtC, based on Random Access Compression Engine (RACE) technology, is field-proven since June 2012, starting with SVC and Storwize v7000 systems.

IBM Real-time Compression (RtC) was introduced with XIV storage system version 11.6 as an optional software feature for models 114 and 214. On the XIV Gen3 model 314, IBM RtC is available in the base license and enabled by default.

By doubling the XIV Gen3 RAM and CPU resources and dedicating the added resources to Real-time "Turbo" Compression, XIV model 314 further increases the high utilization and power efficiencies of XIV 4 TB and 6 TB disk drives, delivering outstanding data economics to your high-end storage.

Starting from version 11.6.1, you can uniformly and centrally manage all software licenses for storage deployments that are built with IBM Spectrum Accelerate software under one enterprise license agreement (ELA), including those for XIV Storage System Gen3. The application of Spectrum Accelerate software licenses to XIV storage system Gen3 hardware is supported.

Starting from version 11.6.2, XIV storage system also offers user-configurable soft capacity of up to 2 PB, and a reduced minimum compressible volume size from 103 GB (in version 11.6) to 51 GB.

IBM Real-time Compression with the IBM XIV storage system Gen3 effectively and efficiently answers a key requirement that typically challenges traditional approaches to data reduction. It reduces capacity while maintaining high performance of the storage system.

XIV implementation of IBM Real-time Compression highlights include:

- Substantial capacity savings across a versatile range of enterprise workloads.
- Use of the IBM Random Access Compression Engine (RACE) technology, which was purpose-built for real-life primary application workloads. IBM Real-time Compression takes advantage of data temporal locality to maximize data savings and system performance.
- Ease of use. An administrator can simply select the **Compressed** check box to create a new compressed volume. For an existing uncompressed volume,an accurate estimation of the potential compression savings is displayed in the XIV GUI. Assessing potential savings in the XIV GUI before compressing data requires considerably less time and effort than what it takes with external tools. Furthermore, non-disruptive conversion of uncompressed volumes to compressed volumes for existing volumes can provide an easy way to reclaim capacity and accelerate ROI.

  Note: Compression is enabled by default on XIV Gen3 model 314 systems.
- Benefiting from the XIV architecture, compression compute resources are evenly distributed across the system, thereby increasing performance and efficiency.

- Due to the system's ability to preserve high performance consistency with compression, IBM Real-time Compression can be used with active primary data. Therefore, it supports workloads that are not candidates for compression in other solutions.

Turbo Compression in model 314

IBM XIV storage system Gen3 model 314 delivers Turbo Compression with larger effective capacity and guaranteed better performance:

- 2 x 6-core CPUs per module (versus 1 x 6-core CPU per module in model 214) - 1x 6-core CPU is dedicated to Real-time-Compression
- 96 GB RAM per module (versus 48 GB RAM per module in model 214) - 48 GB of RAM is dedicated to Real-time-Compression
- 1-2 PB of effective capacity without performance degradation
- Improved IOPS per compressed capacity

Benefits of IBM Real-time Compression

IBM Real-time Compression uses the reliable, field-proven, and patented IBM Random Access Compression Engine (RACE) technology to achieve a valuable combination of high performance and compression efficiencies. Data compression reduces the required storage capacity for a given amount of data, resulting in significantly lower Total Cost of Ownership (TCO).

Among the benefits of using IBM Real-time Compression are:

- Lower effective capacity requirements of a volume typically up to 1/5 of the uncompressed capacity.
- No additional hardware is required to use IBM Real-time Compression.
- Reduced cost for both software and hardware that is licensed by capacity because less physical storage is required for compressed data.
- If you already have 214, save Capital Expenditure (CAPEX) by only purchasing an IBM XIV RtC software license, leverage your current XIV Gen3 investment, and apply compression to existing data and new data.
- Compression is transparent to the applications and can be enabled or disabled on any volume, at any time, non-disruptively.
- Compressed volumes can be mirrored like other XIV volumes. The required bandwidth for a compressed volume is significantly reduced since the replicated data is compressed.

  For the same reason, mirroring and Hyper-Scale Mobility are faster and require less bandwidth because less data is transferred.

  Remote volume copies are always compressed if the source is compressed.

Planning for compression

Before implementing IBM Real-time Compression on your system, assess the current types of data and volumes that are used on your system.

Note: If you are using IBM XIV storage system software version 11.6.x or later, and IBM XIV Management Tools version 4.6 or later, you can see storage saving estimates (even if compression is not licensed or is just disabled). The decision to compress volumes can be based on the expected storage savings of the compressed data and the expected effect on performance.

Understanding compression rates, ratios and savings

Consider a use case where the original capacity required to hold data was 100 TB, but 20 TB after compression (100 TB = 20 TB + 80 TB saved).

The following values help to clarify these terms:

Compression rate = 80%

Compression savings rate = 80%

Compression savings = 80 TB

Compression ratio = original size (100 TB) *divided by* the size on disk after compression (20 TB) = 5:1

When you consider savings, it is easiest to use the compression rate.

The compression ratio helps in understanding how much effective data you can store on your system. So, when you have a 5:1 compression ratio, you will be able to store 500 TB of data on 100 TB of physical capacity.

Prerequisites and limitations

The following prerequisites and limitations are for XIV storage system models 114 or 214, with XIV Storage software version 11.6 or later, and models 214 or 314 with XIV storage software version 11.6.1 or later.

Tip: Consider using version 11.6.1 with model 214 to benefit from improvements like compressed volume size.

- Compressed volumes must be created in thin-provisioned pools.
- To convert a mirrored volume from uncompressed to compressed (or vice versa), the mirroring relationship for that volume must first be removed and then recreated after the conversion.
- Snapshots cannot be converted from compressed to uncompressed and vice versa. Snapshots that already existed before a volume was converted from non-compressed to compressed are not converted and are not available with the converted volume.
- Space requirements:
  - Prior to enabling compression, the system must have a minimum of 17 GB of free hard space available. Enabling compression reserves 17 GB from the available system hard capacity. It is reserved for internal system use only.
  - Before the compression process, there must be enough space for both compressed and uncompressed versions of the volume.
  - For models 114 and 214 with version 11.6.0: Volume size must be at least 103 GB before compression.

    For models 214 and 314 with version 11.6.1: Volume size must be at least 51 GB before compression.

The following is a partial list of limitations.

- Up to 1024 volumes and snapshots can be compressed.
- The following limits apply to compression capacity:
  - System must have a minimum of 17 GB of free hard space to enable IBM Real-time Compression.

  - Thin pools require a minimum of 17 GB of free hard space available to convert or transform volumes from uncompressed to compressed.
  - Thin pools require a minimum free soft space that is at least as large as the volume size that is being converted from uncompressed to compressed.
  - When you decompress a compressed volume, you must have both free hard space at least the size of the uncompressed volume and free soft space.

    It is a good practice to have free soft space at least the size of the uncompressed volume.
  - The Storage Admin can modify the system soft capacity.

    Tip: Over-provisioning with Real-time Compression is safe, since compression ratios are predictable and stable.
- To compress an uncompressed volume in a thick pool, it must be moved to a thin-provisioned pool with compression enabled.
- Only one conversion process can be active at any time.
- Adding a module, rebuilding a disk, or upgrading the system suspends and then resumes the conversion process.

For the most current information about the limitations, refer to the Limitations section of the IBM XIV storage system Gen3 Release Notes, versions 11.6.0, 11.6.1 and 11.6.2.

Estimating compression savings

Compressible data can be identified and expected compression ratios can be estimated even before using compression.

On an XIV system supporting compression, the compression ratio for all uncompressed volumes in the system is continuously estimated, even before enabling compression. The decision to use compression can be based on the expected storage savings of the compressed data and the expected effect on performance (throughput and latency) of the compression processing overhead.

Information on compression usage can also be monitored using the XIV GUI to determine the potential savings to your storage capacity when uncompressed volumes are compressed. You can view the total percentage and total size of capacity savings when compression is used on the system. Compression savings across individual domains, pools and volumes can also be monitored. These compression values can be used to determine which volumes have achieved the highest compression savings. See the *IBM XIV Management Tools Operations Guide* for more information on monitoring and using compression.

Note: Keep in mind that the expected storage savings can vary from 5% higher or lower than the actual compression ratio. A negative estimated compression ratio can be due to metadata that consumes storage space on a volume, even when a 0% estimated compression ratio is received from data that cannot be compressed.

Effective capacity

Effective capacity is the amount of storage that is virtually allocated to applications.

Using thin-provisioned storage architectures, the effective capacity is larger than the array usable capacity. This is made possible by over-committing capacity, or by

compressing the served data. Compression is the preferred method to apply thin-provisioning to usable capacity, since the compression ratio is highly predictable.

Hard capacity denotes usable, non-compressed capacity, whereas *soft capacity* denotes the nominal capacity that is assigned to volumes, and reported to any hosts mapped to those volumes. Thin provisioning denotes committing more soft capacity than hard capacity. Soft capacity is assigned at a pool level. In the case of compression, thin-provisioning is obvious: a compressed volume will forever use less hard capacity than soft capacity.

In XIV terms, the effective capacity is allocated out of the system soft capacity, and is the sum of the sizes of all the allocated volumes.

In an XIV System Gen3 Turbo Compression model 314, the maximal, non-compressed hard capacity supported in a single XIV frame is 485 TB (15 modules, 6 TB drives). However, an XIV frame can effectively accommodate up to 2 PB of real written data - when the data is compressed. With very high compression rates, filling a system up to 2 PB of soft capacity may not require a lot of usable capacity.

The maximum *soft capacity* that can be allocated to volumes in XIV is 2 PB. Considering the compression ratio for typical data profiles on XIV systems, the effective soft capacity leveraged within a 15-module frame will range from 1 PB to 2 PB. To maximize the utilization of XIV hard and soft capacity with compressed data, and avoid over-sizing the system, it is important to assess the expected compression ratio for the stored data. The Comprestimator command-line host-based utility can be used for that purpose. For more information on Comprestimator, see “Estimating compression savings using IBM Comprestimator utility” on page 55.

The maximum effective capacity is reached when all the soft capacity has been allocated to volumes (that is, 2 PB).

For more information on how to fine-tune soft and hard capacities, refer to "Additional space utilization guidance" in the Real-time Compression with IBM XIV Storage System Model 314 *(REDP-5306)* Redbook.

General compression saving guidelines

The best candidates for data compression are data types that are not already compressed. Compressed data types are used by many workloads and applications, such as databases, character/ASCII-based data, email systems, server virtualization, CAD/CAM, software development systems, and vector data.

The following examples represent workloads and data that are already compressed and are, therefore, not good candidates for compression.

- Compressed audio, video, and image file formats -

  File types such as JPEG, PNG, MP3, medical imaging (DICOM), and MPEG2
- Compressed user productivity file formats -

  Microsoft Office 2007 newer formats (.pptx, .docx, .xlsx, and so on), PDF files, Microsoft Windows executable files (.exe), and so on
- Compressed file formats

  File types such as .zip, .gzip, .rar, .cab, and .tgz

IBM Real-time Compression is best suited for data that has an estimated compression savings of 25% or higher. There are various configuration items that affect the performance of compression on the system. Different data types have different compression ratios, and it is important to determine the compressible data currently on your system.

The IBM Comprestimator, a host-based utility, can be used to estimate expected compression rates. Compressing selectively, based on saving estimates, optimizes both capacity use and performance. For more information on Comprestimator, see "Estimating compression savings using IBM Comprestimator utility" on page 55.

The following table shows the compression ratio for common applications and data types:

Table 1. Compression ratios for different data types

Data Types/Applications	Compression Ratios
Productivity	Up to 75%
Databases	Up to 80%
CAD/CAM	Up to 70%
Virtualization	Up to 75%

Note: The required capacity reserve is equal to the size of the volume (not the used capacity, but the volume size).

Estimating compression savings using XIV GUI

The XIV software provides a built-in comprestimator function from the XIV Management Tools GUI on an XIV system supporting Real-time Compression.

From the XIV GUI, compressible data can be identified and expected compression ratios can be estimated even before using compression. Compression does not even have to be enabled to view compression saving estimates. Continuous saving estimates are visible at all times for uncompressed volumes. And the compression ratio for all uncompressed volumes in the system is continuously estimated in a cyclical manner. That is, the potential savings estimations appear continuously and are updated every few hours.

The decision to use compression can be based on the expected storage savings of the compressed data and the expected effect on performance (throughput and latency) of the compression processing overhead.

Figure 17 on page 55 displays the compression savings (in both percentage and GB values) of compressed volumes and uncompressed volumes with *estimates* of potential savings, should the uncompressed volumes be compressed. These potential compression savings are constantly being updated.

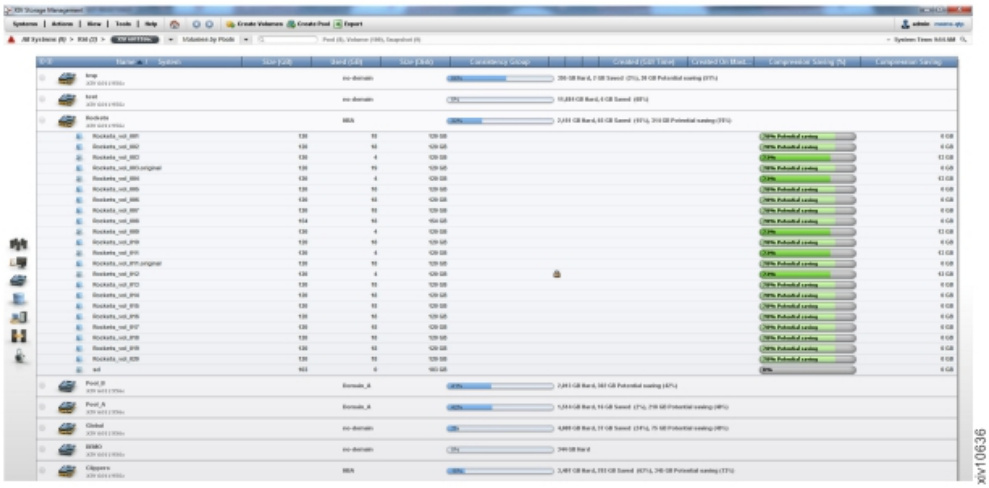


Figure 17. Compression savings in the Volumes by Pools view

Compression Saving and *Compression Saving (%)* appear on the following views:

- Storage Pools
- Volumes by Pools
- Volumes and Snapshots
- Consistency Groups
- Domains
- Systems list

Estimating compression savings using IBM Comprestimator utility

Comprestimator is a stand-alone tool that can be used to estimate compression savings for data that is either not on XIV storage, or on an XIV Gen2 or Gen3 storage system with system software earlier than 11.6.x.

Comprestimator is a command-line host-based utility that can be used to estimate the expected compression rate for block-devices. The utility uses advanced mathematical and statistical algorithms to perform sampling and analysis efficiently. The utility also displays its accuracy level by showing the compression accuracy range of the results that are achieved based on the formulas it uses, deviating plus or minus 5 percent based on the formulas that are used by the RACE implementation.

The utility runs on a host that has access to the devices to be analyzed. It runs only read operations, so it has no effect on the data that is stored on the device. The following links provide useful information about installing Comprestimator on a host and using it to analyze devices on that host: Comprestimator Utility and Comprestimator Utility Version 1.5.2.2.

For more information on Comprestimator, refer to the *IBM Real-time Compression on the IBM XIV Storage System (REDP-5215)* Redpaper at http://www.redbooks.ibm.com/redpieces/abstracts/redp5215.html?Open&pdfbookmark.

Chapter 8. Synchronous remote mirroring

IBM XIV Storage System features synchronous and asynchronous remote mirroring for disaster recovery. Remote mirroring can be used to replicate the data between two geographically remote sites. The replication ensures uninterrupted business operation if there is a total site failure.

Remote mirroring provides data protection for the following types of site disasters:

Site failure
When a disaster happens to a site that is remotely connected to another site, the second site takes over and maintains full service to the hosts connected to the first site. The mirror is resumed after the failing site recovers.

Split brain
After a communication loss between the two sites, each site maintains full service to the hosts. After the connection is resumed, the sites complement each other's data to regain mirroring.

Synchronous and asynchronous remote mirroring

The two distinct methods of remote mirroring - synchronous and asynchronous - are described on this chapter and on the following chapter. Throughout this chapter, the term *remote mirroring* refers to *synchronous* remote mirroring, unless clearly stated otherwise.

Remote mirroring basic concepts

Synchronous remote mirroring provides continuous availability of critical information in the case of a disaster scenario.

A typical remote mirroring configuration involves the following two sites:

Primary site
The location of the master storage system.

A local site that contains both the data and the active servers.

Secondary site
The location of the secondary storage system.

A remote site that contains a copy of the data and standby servers. Following a disaster at the master site, the servers at the secondary site become active and start using the copy of the data.

Master
The volume or storage system which is mirrored. The master volume or storage system is usually at the master site.

Slave The volume or storage system to which the master is mirrored. The slave volume or storage system is usually at the Secondary site.

One of the main goals of remote mirroring is to ensure that the secondary site contains the same (consistent) data as the master site. With remote mirroring, services are provided seamlessly by the hosts and storage systems at the secondary site.

The process of ensuring that both storage systems contain identical data at all times is called *remote mirroring*. Synchronous remote mirroring is performed during each write operation. The write operation issued by a host is sent to both the master and the slave storage systems.

To ensure that the data is also written to the secondary system, acknowledgment of the write operation is only issued after the data has been written to both storage systems. This ensures the consistency of the secondary storage system to the master storage system except for the contents of any last, unacknowledged write operations. This form of mirroring is called synchronous mirroring.

In a remote mirroring system, reading is performed from the master storage system, while writing is performed on both the master and the slave storage systems, as previously described.

The IBM XIV Storage System supports configurations where server pairs perform alternate master or secondary roles with respect to their hosts. As a result, a server at one site might serve as the master storage system for a specific application, while simultaneously serving as the secondary storage system for another application.

Remote mirroring operation

Remote mirroring operations involve *configuration*, *initialization*, *ongoing operation*, *handling of communication failures*, and *role switching* activities.

The following list defines the remote mirroring operation activities:

Configuration
Local and remote replication peers are defined by an administrator who specifies the primary and secondary volumes. For each coupling, several configuration options can be defined.

Initialization
Remote mirroring operations begin with a master volume that contains data and a formatted slave volume. The first step is to copy the data from the master volume to the slave volume. This process is called *initialization*. Initialization is performed once in the lifetime of a remote mirroring coupling. After it is performed, both volumes are considered synchronized.

Ongoing Operation
After the initialization process is complete, remote mirroring is activated. During this activity, all data is written to the master volume and then to the slave volume. The write operation is complete after an acknowledgment from the slave volume. At any point, the master and slave volumes are identical except for any unacknowledged (pending) writes.

Handling of Communication Failures
From time to time the communication between the sites might break down, and it is usually preferable for the primary site to continue its function and to update the secondary site when communication resumes. This process is called *synchronization*.

Role Switching
When needed, a replication peer can change its role from master to slave

or vice versa, either as a result of a disaster at the primary site, maintenance operations, or because of a drill that tests the disaster recovery procedures.

Configuration options

The remote mirroring configuration process involves configuring volumes and volume pair options.

When a pair of volumes point to each other, it is referred to as a *coupling*. In a *coupling relationship*, two volumes participate in a remote mirroring system with the slave peer serving as the backup for the master peer. The coupling configuration is identical for both master volumes and slave volumes.

Table 2. Configuration options for a volume

Name	Values	Definition
Role	None, Master, Slave	Role of a volume. (Primary and Secondary are designations.)
Peer	Remote target identification and the name of the volume on the remote target.	Identifies the peer volume.

Table 3. Configuration options for a coupling

Name	Values	Definition
Activation	Active, Stand-by.	Activates or deactivates remote mirroring.

Volume configuration

The role of each volume and its peer volumes on the IBM XIV Storage System must be defined for it to function within the remote mirroring process.

The following concepts are to be configured for volumes and the relations between them:

- Volume role
- Peer

The *volume role* is the current function of the volume. The following volume roles are available:

None The volume is created using normal volume creation procedures and is not configured as part of any remote mirroring configuration.

Master volume
The volume is part of a mirroring coupling and serves as the master volume.

All write operations are made to this master volume. It ensures that write operations are made to the slave volume before acknowledging their success.

Slave volume
This volume is part of a mirroring coupling and serves as a backup to the master volume.

Data is read from the slave volume, but cannot be written to it.

A *peer* is a volume that is part of a coupling. A volume with a role other than none has to have a peer designation, and a corresponding master or slave volume assigned to it.

Configuration errors

In some cases, configuration on both sides might be changed in a non-compatible way. This is defined as a *configuration error*. For example, switching the role of only one side when communication is down causes a configuration error when connection resumes.

Mixed configuration

The volumes on a single storage system can be defined in a mixture of configurations.

For example, a storage system can contain volumes whose role is defined as master, as well as volumes whose roles are defined as slave. In addition, some volumes might not be involved in a remote mirroring coupling at all.

The roles assigned to volumes are *transient*. This means a volume that is currently a master volume can be defined as a slave volume and vice versa. The term *local* refers to the master volume, and *remote* refers to the slave volume for processes that switch the master and slave assignments.

Communication errors

When the communication link to the secondary volume fails or the secondary volume itself is not usable, processing to the volume continues as usual. The following occurs:

- The system is set to an unsynchronized state.
- All changes to the master volume are recorded and then applied to the slave volume after communication is restored.

Coupling activation

Remote mirroring can be manually activated and deactivated per coupling. When it is activated, the coupling is in *Active* mode. When it is deactivated, the coupling is in *Standby* mode.

These modes have the following functions:

Active Remote mirroring is functioning and the data is being written to both the master and the slave volumes.

Standby
Remote mirroring is deactivated. The data is not being written to the slave volume, but it is being recorded in the master volumes which will later synchronize the slave volume.

Standby mode is used mainly when maintenance is performed on the secondary site or during communication failures between the sites. In this mode, the master volumes do not generate alerts that the mirroring has failed.

The coupling lifecycle has the following characteristics:

- When a coupling is created, it is always initially in Standby mode.

- Only a coupling in Standby mode can be deleted.
- Transitions between the two states can only be performed from the UI and on the volume.

Synchronous mirroring statuses

The status of the synchronous remote mirroring volume represents the state of the storage volume in regard to its remote mirroring operation.

The state of the volume is a function of the status of the communication link and the status of the coupling between the master volume and the slave volume. “Link status” describes the various statuses of a synchronous remote mirroring volume during remote mirroring operations.

Table 4. Synchronous mirroring statuses

Entity	Name	Values	Definition
Link	Status	• Up • Down	Specifies if the communications link is up or down.
Coupling	Operational status	• Operational • Non-operational	Specifies if remote mirroring is working.
	Synchronization status	• Initialization • Synchronized • Unsynchronized • Consistent • Inconsistent	Specifies if the master and slave volumes are consistent.
	Last-secondary-timestamp	Point-in-time date	Time stamp for when the secondary volume was last synchronized.
	Synchronization process progress	Synchronization status	Amount of data remaining to be synchronized between the master and slave volumes due to non-operational coupling.
	Secondary locked	Boolean	True, if secondary was locked for writing due to lack of space; otherwise false. This can happen during the synchronization process when there is not enough space for the last-consistent snapshot.
	Configuration error	Boolean	True, if the configuration of the master and secondary slave is inconsistent.

Link status

The status of the communication link can be either *up* or *down*. The link status of the master volume is, of course, also the link status of the slave volume.

Operational status

The coupling between the master and slave volumes is either *operational* or *non-operational*. To be operational, the link status must be up and the coupling must be activated. If the link is down or if the remote mirroring feature is in Standby mode, the operational status is non-operational.

Synchronization status

The synchronization status reflects the consistency of the data between the master and slave volumes. Because the purpose of the remote mirroring feature is to ensure that the slave volume is an identical copy of the master volume, this status indicates whether this objective is currently attained.

The possible synchronization statuses for the master volume are:

Initialization
The first step in remote mirroring is to create a copy of the data from the master volume to the slave volume, at the time when the mirroring was set to place. During this step, the coupling status remains initialization.

Synchronized (master volume only)
This status indicates that all data that was written to the primary volume and acknowledged has also been written to the secondary volume. Ideally, the primary and secondary volumes should always be synchronized. This does not imply that the two volumes are identical because at any time, there might be a limited amount of data that was written to one volume, but was not yet written to its peer volume. This means that their write operations have not yet been acknowledged. These are also known as pending writes.

Unsynchronized (primary volume only)

After a volume has completed the initialization stage and achieved the synchronized status, a volume can become unsynchronized.
This occurs when it is not known whether all the data that was written to the primary volume was also written to the secondary volume. This status occurs in the following cases:

- **Communications link is down** - As a result of the communication link going down, some data might have been written to the primary volume, but was not yet written to the secondary volume.
- **Secondary system is down** - This is similar to communication link errors because in this state, the primary system is updated while the secondary system is not.
- **Remote mirroring is deactivated** - As a result of the remote mirroring deactivation, some data might have been written to the primary volume and not to the secondary volume.

It is always possible to reestablish the synchronized status when the link is reestablished or the remote mirroring feature is reactivated, no matter what was the reason for the unsynchronized status.

Because all updates to the primary volume that are not written to the secondary volume are recorded, these updates are written to the secondary volume. The synchronization status remains unsynchronized from the time that the coupling is not operational until the synchronization process is completed successfully.

Synchronization progress status

During the synchronization process while the secondary volumes are being updated with previously written data, the volumes have a dynamic synchronization process status.

This status is comprised of the following sub-statuses:

Size to complete
The size of data that requires synchronization.

Part to synchronize
The size to synchronize divided by the maximum size-to-synchronize since the last time the synchronization process started. For coupling initialization, the size-to-synchronize is divided by the volume size.

Time to synchronize
Estimate of the time, which is required to complete the Synchronization process and achieve synchronization, based on past rate.

Last secondary timestamp

A timestamp is taken when the coupling between the primary and secondary volumes becomes non-operational.

This timestamp specifies the last time that the secondary volume was consistent with the primary volume. This status has no meaning if the coupling's synchronization state is still *initialization*. For synchronized coupling, this timestamp specifies the current time. Most importantly, for an unsynchronized coupling, this timestamp denotes the time when the coupling became non-operational.

The timestamp is returned to current only after the coupling is operational and the primary and secondary volumes are synchronized.

I/O operations

I/O operations are performed on the primary and secondary volumes across various configuration options.

I/O on the primary

Read All data is read from the primary (local) site regardless of whether the system is synchronized.

Write

- If the coupling is operational, data is written to both the primary and secondary volumes.
- If the coupling is non-operational, an error is returned.

The error reflects the type of problem that was encountered. For example, remote mirroring has been deactivated, there is a locked secondary error, or there is a link error.

I/O on the secondary

A secondary volume can have LUN maps and hosts associated with it, but it is only accessible as a read-only volume. These maps are used by the backup hosts when a switchover is performed. When the secondary volume becomes the

primary volume, hosts can write to it on the remote site. When the primary volume becomes a secondary volume, it becomes read-only and can be updated only by the new primary volume.

Read Data is read from the secondary volume like from any other volume.

Write An attempt to write on the secondary volume results in a volume read-only SCSI error.

Synchronization process

When a failure condition has been resolved, remote mirroring begins the process of synchronizing the coupling. This process updates the secondary volume with all the changes that occurred while the coupling was not operational.

This section describes the process of synchronization.

State diagram

Couplings can be in the *Initialization*, *Synchronized*, *Timestamp*, or *Unsychronized* state.

The following diagram shows the various coupling states that the IBM XIV Storage System assumes during its lifetime, along with the actions that are performed in each state.

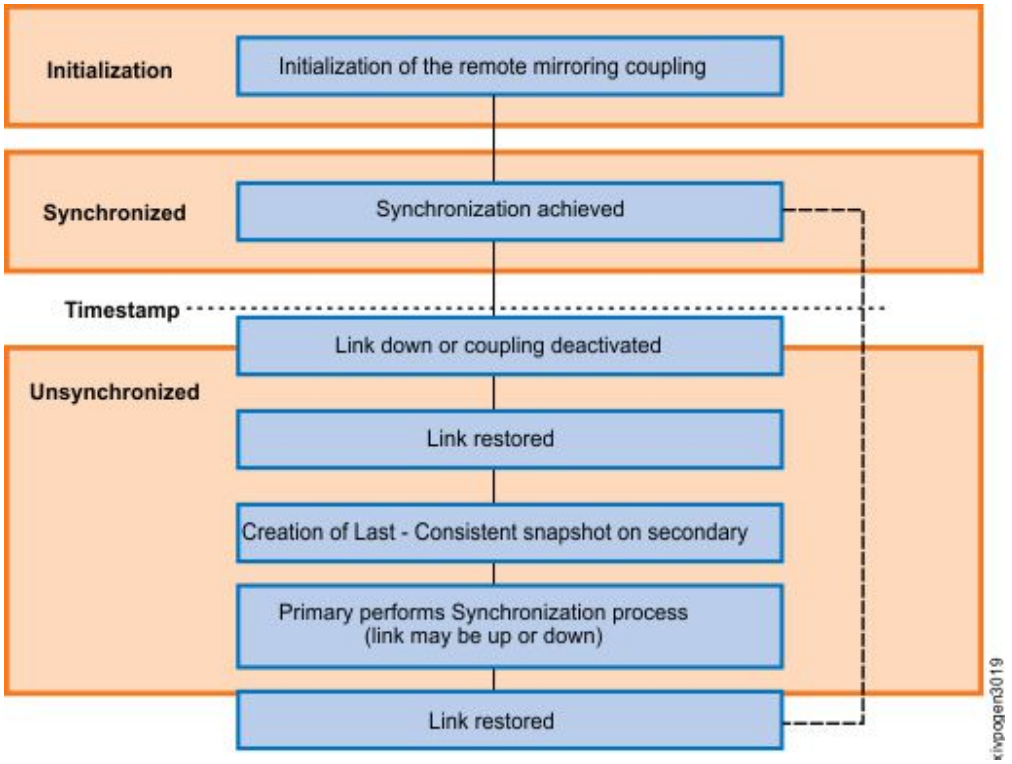


Figure 18. Coupling states and actions

The following list describes each coupling state:

Initialization
The secondary volume has a Synchronization status of Initialization. During this state, data from the primary volume is copied to the secondary volume.

Synchronized
This is the working state of the coupling, where both the primary and secondary volumes are consistent.

Timestamp
Remote mirroring has become non-operational so a time stamp is recorded. During this status, the following actions take place:

1. Coupling deactivation, or the link is down
2. Coupling reactivation, or the link is restored.

Unsynchronized
Remote mirroring is non-operational because of a communications failure or because remote mirroring was deactivated. Therefore, the primary and secondary volumes are not synchronized. When remote mirroring resumes, steps are taken to return to the Synchronized state.

Coupling recovery

Remote mirroring recovers from non-operational coupling.

When remote mirroring recovers from a non-operational coupling, the following actions take place:

- If the secondary volume is in the Synchronized state, a last-consistent snapshot of the secondary volume is created and named with the string `secondary-volume-time-date-consistent-state`.
- The primary volume updates the secondary volume until it reaches the Synchronized state.
- The primary volume deletes the special snapshot after all couplings that mirror volumes between the same pair of systems are synchronized.

Uncommitted data

When the coupling is in an Unsynchronized state, for best-effort coupling, the system must track which data in the primary volume has been changed, so that these changes can be committed to the secondary when the coupling becomes operational again.

The parts of the primary volume that must be committed to the secondary volume and must be marked are called *uncommitted data*.

Note: There is only metadata that marks the parts of the primary volume that must be written to the secondary volume when the coupling becomes operational.

Constraints and limitations

Coupling has constraints and limitations.

The following constraints and limitations exist:

- The Size, Part, or Time-to-synchronize are relevant only if the Synchronization status is Unsynchronized.
- The last-secondary-time stamp is only relevant if the coupling is Unsynchronized.

Last-consistent snapshots

Before the synchronization process is initiated, a snapshot of the secondary volume is created. This snapshot is created to ensure the usability of the secondary volume in case of a primary site disaster during the synchronization process.

If the primary volume is destroyed before the synchronization is completed, the secondary volume might be inconsistent because it may have been only partially updated with the changes that were made to the primary volume. The reason for this possible inconsistency is the fact that the updates were not necessarily performed in the same order in which they were written by the hosts.

To handle this situation, the primary volume always creates a snapshot of the last-consistent secondary volume after re-connecting to the secondary machine, and before starting the synchronization process.

The last consistent snapshot

The Last Consistent snapshot (LCS) is created by the system on the Slave peer in synchronous mirroring just before mirroring resynchronization needs to take place. Mirroring resynchronization takes place after link disruption, or a manual mirroring deactivation. In both cases the Master will continue to accept host writes, yet will not replicate them onto the Slave as long as the link is down, or the mirroring is deactivated.

Once the mirroring is restored and activated, the system takes a snapshot of the Slave (LCS), which represents the data that is known to be mirrored, and only then the non yet mirrored data written to the Master is replicated onto the Slave through a resynchronization process.

The LCS is deleted automatically by the system once the resynchronization is complete for all mirrors on the same target , but if the Slave peer role is changed during resynchronization ⌂ this snapshot will not be deleted.

The external last consistent snapshot

Prior to the introduction of the external last consistent snapshot, whenever the peer's role was changed back to Slave and sometime whenever a new resynchronization process had started, the system would detect an LCS on the peer and would not create a new one. If, during such an event, the peer was not part of a mirrored consistency group (mirrored CG) it would meant that not all volumes have the same LCS timestamp. If the peer was part of a mirrored consistency group, we would have a consistent LCS but not as current as possibly expected. This situation is avoided thanks to the introduction of the external last consistent snapshot.

Whenever the role of a Slave with an LCS is changed to Master while mirroring resynchronization is in progress (in the system/target not specific to this volume), the LCS is renamed external last consistent (ELCS). The ELCS retains the LCS deletion priority of 0. If the peer's role is later changed back to Slave and sometime afterwards a new resynchronization process starts, a new LCS will be created.

Subsequently changing the Slave role again will rename the existing external last consistent snapshot to external last consistent x (x being the first available number starting from 1) and will rename the LCS to external last consistent. The deletion priority of external last consistent will be 0, but the deletion priority of the new

external last consistent x will be the system default (1), and can thus be deleted automatically by the system upon pool space depletion.

It is crucial to validate whether the LCS or an ELCS (or even ELC x) should serve as a restore point for the Slave peer if resynchronization cannot be completed. While snapshots with deletion priority 0 are not automatically deleted by the system to free space, the external last consistent and external last consistent x snapshots can be manually deleted by the administrator if so required. As the deletion of such snapshots might leave an inconsistent peer without a consistent snapshot to be restored from (in case the resynchronization cannot complete due to Master unavailability) ⌂ it should generally be avoided even when pool space is depleted, unless the peer is guaranteed to be consistent.

Manually deleting the last consistent snapshot

- Only the XIV support team can delete the last consistent snapshot.
- The XIV support team can also configure a mirroring so that it does not create the last consistent snapshot. This is required when the system that contains the secondary volume is fully utilized and an additional snapshot cannot be created.

Timestamp

A timestamp is taken when the coupling between the primary and secondary volumes becomes non-operational. This timestamp specifies the last time that the secondary volume was consistent with the primary volume.

This status has no meaning if the coupling's synchronization state is still *Initialization*. For synchronized couplings, this timestamp specifies the current time. Most importantly, for unsynchronized couplings, this timestamp denotes the time when the coupling became non-operational.

This table provides an example of a failure situation and describes the time specified by the timestamp.

Table 5. Example of the last consistent snapshot time stamp process

Time	Status of coupling	Action	Last consistent timestamp
Up to 12:00	Operational and synchronized		Current
12:00 - 1:00	Failure caused the coupling to become non-operational. The coupling is Unsynchronized.	Writing continues to the primary volume. Changes are marked so that they can be committed later.	12:00
13:00	Connectivity resumes and remote mirroring is operational. Synchronization begins. The coupling is still Unsynchronized.	All changes since the connection was broken are committed to the secondary volume, as well as current write operations.	12:00
13:15	Synchronized		Current

Secondary locked error status

While the synchronization process is in progress, there is a period in which the secondary volume is not consistent with the primary volume, and a last-consistent snapshot is maintained. While in this state, I/O operations to the secondary

volume can fail because there is not enough space. There is not enough space because every I/O operation potentially requires a copy-on-write of a partition.

Whenever I/O operations fail because there is not enough space, all couplings in the system are set to the secondary-locked status and the coupling becomes non-operational. The administrator is notified of a critical event, and can free space on the system containing the secondary volume.

If this situation occurs, contact an IBM XIV field engineer. After there is enough space, I/O operations resume and remote mirroring can be reactivated.

Role switchover

Role switchover when remote mirroring is operational

Role switching between primary and secondary volumes can be performed from the IBM XIV Storage Management GUI or the XCLI after the remote mirroring function is operational. After role switchover occurs, the primary volume becomes the secondary volume and vice versa.

There are two typical reasons for performing a switchover when communications between the volumes exist. These are:

Drills Drills can be performed on a regular basis to test the functioning of the secondary site. In a drill, an administrator simulates a disaster and tests that all procedures are operating smoothly.

Scheduled maintenance
To perform maintenance at the primary site, switch operations to the secondary site on the day before the maintenance. This can be done as a preemptive measure when a primary site problem is known to occur.

This switchover is performed as an automatic operation acting on the primary volume. This switchover cannot be performed if the primary and secondary volumes are not synchronized.

Role switchover when remote mirroring is nonoperational

A more complex situation for role switching is when there is no communication between the two sites, either because of a network malfunction, or because the primary site is no longer operational.

Switchover procedures differ depending on whether the primary and secondary volumes are connected or not. As a general rule, the following is true:

- When the coupling is deactivated, it is possible to change the role on one side only, assuming that the other side will be changed as well before communication resumes.
- If the coupling is activated, but is either unsynchronized, or nonoperational due to a link error, an administrator must either wait for the coupling to be synchronized, or deactivate the coupling.
- On the secondary volume, an administrator can change the role even if coupling is active. It is assumed that the coupling will be deactivated on the primary volume and the role switch will be performed there as well in parallel. If not, a configuration error occurs.

Switch secondary to primary

The role of the secondary volume can be switched to primary using the IBM XIV Storage Management GUI or the XCLI. After this switchover, the following is true:

- The secondary volume is now the primary.
- The coupling has the status of unsynchronized.
- The coupling remains in standby mode, meaning that the remote mirroring is deactivated. This ensures an orderly activation when the role of the other site is switched.

The new primary volume starts to accept write commands from local hosts. Because coupling is not active, in the same way as any primary volume, it maintains a log of which write operations should be sent to the secondary when communication resumes.

Typically, after switching the secondary to the primary volume, an administrator also switches the primary to the secondary volume, at least before communication resumes. If both volumes are left with the same role, a configuration error occurs.

Secondary consistency

Switching the secondary volume to primary, when the last-consistent snapshot is no longer available

If the user is switching the secondary to a primary volume, and a snapshot of the last_consistent state exists, then the link was broken during the process of synchronizing. In this case, the user has a choice between using the most-updated version, which might be inconsistent, or reverting to the last_consistent snapshot. Table 6 outlines this process.

Table 6. Disaster scenario leading to a secondary consistency decision

Time	Status of remote mirroring	Action
Up to 12:00	Operational	Volume A is the primary volume and volume B is the secondary volume.
12:00	Non-operational because of communications failure	Writing continues to volume A and volume A maintains the log of changes to be committed to volume B.
13:00	Connectivity resumes and remote mirroring is operational	A last_consistent snapshot is generated on volume B. After that, volume A starts to update volume B with the write operations that occurred since communication was broken.
13:05	Primary site is destroyed and all information is lost	
13:10		Volume B is becoming the primary. The operators can choose between using the most-updated version of volume B, which contains only parts of the write operations committed to volume A between 12:00 and 13:00, or use the last-consistent snapshot, which reflects the state of volume B at 12:00.

If a last-consistent snapshot exists and the role is changed from secondary to primary, the system automatically generates a snapshot of the volume. This snapshot is named *most_updated snapshot*. It is generated to enable a safe reversion to the latest version of the volume, when recovering from user errors. This snapshot can only be deleted by the IBM XIV Storage System support team.

If the coupling is still in the initialization state, switching cannot be performed. In the extreme case where the data is needed even though the initial copy was not completed, a volume copy can be used on the primary volume.

Switch primary to secondary

When coupling is inactive, the primary machine can switch roles. After such a switch, the primary volume becomes the secondary one.

Because the primary volume is inactive, it is also in the unsynchronized state, and it might have an uncommitted data list. All these changes will be lost. When the volume becomes secondary, this data must be reverted to match the data on the peer volume, which is now the new primary volume. In this case, an event is created, summarizing the size of the changes that were lost.

The uncommitted data list has now switched its semantics, and instead of being a list of updates that the local volume (old primary, new secondary) needs to update on the remote volume (old secondary, new primary), the list now represents the updates that need to be taken from the remote to the local volume.

Upon reestablishing the connection, the local volume (current secondary, which was the primary) will update the remote volume (new primary) with this uncommitted data list to update, and it is the responsibility of the new primary volume to synchronize these lists to the local volume (new secondary).

Resumption of remote mirroring after role change

When the communication link is resumed after a switchover of roles in which both sides were switched, the coupling now contains one secondary and one primary volume.

Note: After a role switchover, the coupling is in standby. The coupling can be activated before or after the link resumes.

Table 7 describes the system when the coupling becomes operational, meaning after the communications link has been resumed and the coupling has been reactivated. When communications is resumed, the new primary volume (old secondary) might be in the unsynchronized state, and have an uncommitted data list to synchronize.

The new secondary volume (old primary), might have an uncommitted data list to synchronize from the new primary volume. These are write operations that were written after the link was broken and before the role of the volume was switched from primary to secondary. These changes must be reverted to the content of the new primary volume. Both lists must be used for synchronization by the new primary volume.

Table 7. Resolution of uncommitted data for synchronization of the new primary volume

Time	Status of remote mirroring	Action
Up to 12:00	Operational and synchronized	Volume A is the primary volume and volume B is the secondary volume.
12:00 to 12:30	Communication failure, coupling becomes non-operational	Volume A still accepts write operations from the hosts and maintains an uncommitted data list marking these write operations. For example, volume A accepted a write operation to blocks 1000 through 2000, and marks blocks 1000 through 2000 as ones that need to be copied to volume B after reconnection.

Table 7. Resolution of uncommitted data for synchronization of the new primary volume (continued)

Time	Status of remote mirroring	Action
12:30	Roles changed on both sides	Volume A is now secondary and volume B is primary. Volume A should now revert the changes done between 12:00 and 12:30 to their original values. This data reversion is only performed after the two systems reconnect. For now, volume A reverts the semantics of the uncommitted data list to be data that needs to be copied from volume B. For example, blocks 1000 through 2000 need to be copied now from volume B.
12:30 to 13:00	Volume B is primary, volume A is secondary, coupling is non-operational	Volume A does not accept changes because it is a secondary in a nonoperational coupling. volume B is now a primary in a nonoperational coupling, and maintains its own uncommitted data list of the write operations that were performed since it was defined as the primary. For example, if the hosts wrote blocks 1500 through 2500, volume B marks these blocks to be copied to volume A.
13:00	Connectivity resumes	Volume B and volume A communicate and volume B merges the lists of uncommitted data. Volume B copies to volume A both the blocks that changed in volume B between 12:30 and 13:00, as well as the blocks that changed in volume A between 12:00 and 12:30. For example, volume B could copy to volume A blocks 1000 through 2500, where blocks 1000 through 1500 would revert to their original values at 12:00 and blocks 1500 through 2500 would have the values written to volume B between 12:30 and 13:00. Changes written to volume A between 12:00 and 12:30 are lost.

Reconnection when both sides have the same role

What happens when one side was switched while the link was down?

Situations where both sides are configured to the same role can only occur when one side was switched while the link was down. This is a user error, and the user must follow these guidelines to prevent such a situation:

- Both sides need to change roles before the link is resumed.
- As a safety measure, it is recommended to first switch the primary to secondary.

If the link is resumed and both sides have the same role, the coupling will not become operational.

To solve the problem, the user must use the role switching mechanism on one of the volumes and then activate the coupling.

In this situation, the system behaves as follows:

- If both sides are configured as secondary volumes, a minor error is issued.
- If both sides are configured as primary volumes, a critical error is issued. Both volumes will be updated locally with remote mirroring being nonoperational until the condition is solved.

Remote mirroring

Remote mirroring and consistency groups

The consistency group has to be compatible with mirroring.

The following assumptions ensure that consistency group procedures are compatible with the remote mirroring function:

- All volumes in a consistency group are mirrored on the same system (as all primaries on a system are mirrored on the same system).
- All volumes in a consistency group have the same role.
- The last_consistent snapshot is created and deleted system-wide, and therefore, it is consistent across the consistency group.

Note: An administrator can incorrectly switch the roles of some of the volumes in a consistency group, while keeping others in their original role. This is not prevented by the system and is detected at the application level.

Using remote mirroring for media error recovery

If a media error is discovered on one of the volumes of the coupling, the peer volume is then used for a recovery.

Supported configurations

Synchronous mirroring supports the following configurations.

- Either Fibre Channel or iSCSI can serve as the protocol between the primary and secondary volumes. A volume accessed through one protocol can be synchronized using another protocol.
- The remote system must be defined as an XIV in the remote-target connectivity definitions.
- All the peers of volumes that belong to the same consistency group on a system must reside on a single remote system.
- Primary and secondary volumes must contain the same number of blocks.

I/O performance versus synchronization speed optimization

The synchronization rate can be adjusted, so as to prevent resource exhaustion.

The IBM XIV Storage System has two global parameters, controlling the maximum rate used for initial synchronization and for synchronization after nonoperational coupling.

These rates are used to prevent a situation where synchronization uses too many of the system or communication line resources.

This configuration parameter can be changed at any time. These parameters are set by the IBM XIV Storage System technical support representative.

Implications regarding volume and snapshot management

Using synchronous mirroring incurs several implications on volume and snapshot management.

- Renaming a volume changes the name of the last_consistent and most_updated snapshots.
- Deleting all snapshots does not delete the last_consistent and most_updated snapshot.
- Resizing a primary volume resizes its secondary volume.
- A primary volume cannot be resized when the link is down.
- Resizing, deleting, and formatting are not permitted on a secondary volume.

- A primary volume cannot be formatted. If a primary volume must be formatted, an administrator must first deactivate the mirroring, delete the mirroring, format both the secondary and primary volumes, and then define the mirroring again.
- Secondary or primary volumes cannot be the target of a copy operation.
- Locking and unlocking are not permitted on a secondary volume.
- Last_consistent and most_updated snapshots cannot be unlocked.
- Deleting is not permitted on a primary volume.
- Restoring from a snapshot is not permitted on a primary volume.
- Restoring from a snapshot is not permitted on a secondary volume.
- A snapshot cannot be created with the same name as the last_consistent or most_updated snapshot.

Chapter 9. Asynchronous remote mirroring

Asynchronous mirroring enables you to attain high availability of critical data through a process that asynchronously replicates data updates that are recorded on a primary storage peer to a remote, secondary peer.

The relative merits of asynchronous and synchronous mirroring are best illustrated by examining them in the context of two critical objectives:

- Responsiveness of the storage system
- Currency of mirrored data

With synchronous mirroring, host writes are acknowledged by the storage system only after being recorded on both peers in a mirroring relationship. This yields high currency of mirrored data (both mirroring peers have the same data), yet results in less than optimal system responsiveness because the local peer cannot acknowledge the host write until the remote peer acknowledges it. This type of process incurs latency that increases as the distance between peers increases.

XIV features both asynchronous mirroring and synchronous mirroring. Asynchronous mirroring is advantageous in various use cases. It represents a compelling mirroring solution in situations that warrant replication between distant sites because it eliminates the latency inherent to synchronous mirroring, and might lower implementation costs. Careful planning of asynchronous mirroring can minimize the currency gap between mirroring peers, and can help to realize better data availability and cost savings.

With synchronous mirroring (first image below), response time (latency) increases as the distance between peers increases, but both peers are synchronized. With asynchronous mirroring (second image below), response time is not sensitive to distance between peers, but the synchronization gap between peers is sensitive to the distance.

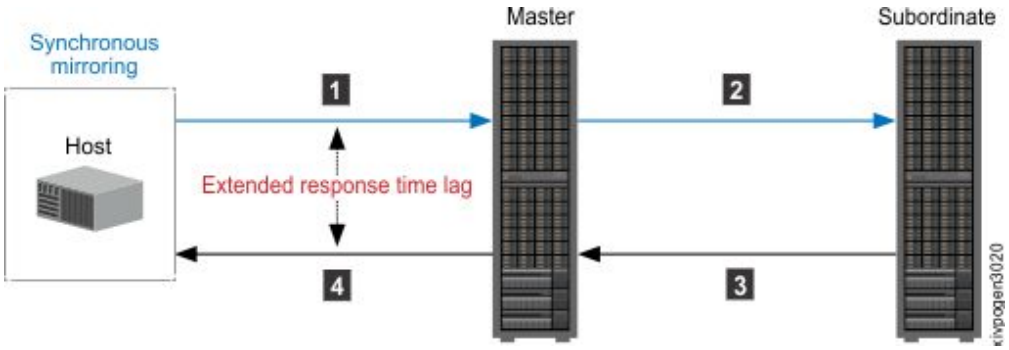


Figure 19. Synchronous mirroring extended response time lag

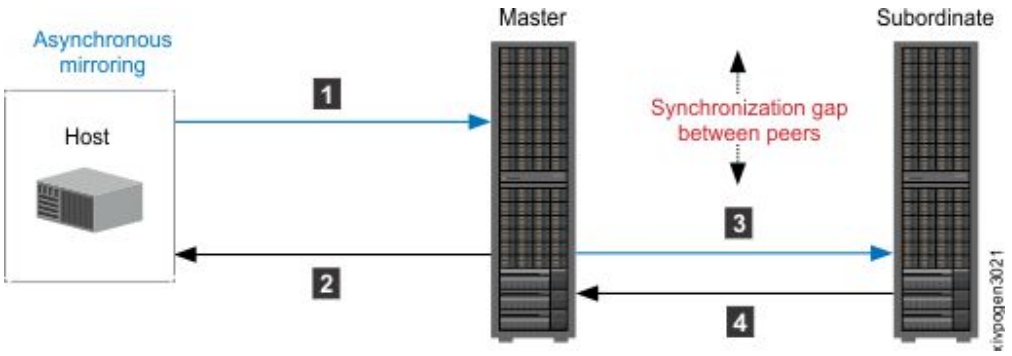


Figure 20. Asynchronous mirroring - no extended response time lag

Note: Synchronous mirroring is covered in Chapter 8, "Synchronous remote mirroring," on page 57.

Features

The IBM XIV Storage System blends existing and new XIV technologies to produce an advanced mirroring solution with unique strengths.

The following are highlights of IBM XIV Storage System asynchronous mirroring:

Advanced snapshot-based technology
IBM XIV asynchronous mirroring is based on XIV snapshot technology, which streamlines implementation while minimizing impact on general system performance. The technology leverages functionality that has previously been effectively employed with synchronous mirroring and is designed to support mirroring of complete systems – translating to hundreds or thousands of mirrors.

Mirroring of consistency groups
IBM XIV supports definition of mirrored consistency groups, which is highly advantageous to enterprises, facilitating easy management of replication for all volumes that belong to a single consistency group. This enables streamlined restoration of consistent volume groups from a remote site upon unavailability of the primary site.

Automatic and manual replication
Asynchronous mirrors can be assigned a user-configurable schedule for automatic, interval-based replication of changes, or can be configured to replicate changes upon issuance of a manual (or scripted) user command. Automatic replication allows you to establish crash-consistent replicas, whereas manual replication allows you to establish application-consistent replicas, if required. The XIV implementation allows you to combine both approaches because you can define mirrors with a scheduled replication and you can issue manual replication jobs for these mirrors as needed.

Multiple RPOs and multiple schedules
IBM XIV asynchronous mirroring enables each mirror to be specified a different RPO, rather than forcing a single RPO for all mirrors. This can be used to prioritize replication of some mirrors over others, potentially making it easier to accommodate application RPO requirements, as well as bandwidth constraints.

Flexible and independent mirroring intervals
IBM XIV asynchronous mirroring supports schedules with intervals

ranging between 20 seconds and 12 hours. Moreover, intervals are independent from the mirroring RPO. This enhances the ability to fine tune replication to accommodate bandwidth constraints and different RPOs.

Flexible pool management
IBM XIV asynchronous mirroring enables the mirroring of volumes and consistency groups that are stores in thin provisioned pools. This applies to both mirroring peers.

Bi-directional mirroring
IBM XIV systems can host multiple mirror sources and targets concurrently, supporting over a thousand mirrors per system. Furthermore, any given IBM XIV Storage System can have mirroring relationships with several other IBM XIV systems. This enables enormous flexibility when setting mirroring configurations.

The number of systems with which the Storage System can have mirroring relationships is specified out side this document (see the IBM XIV Storage System Data Sheet).

Concurrent synchronous and asynchronous mirroring
The IBM XIV Storage System can concurrently run synchronous and asynchronous mirrors.

Easy transition between peer roles
IBM XIV mirror peers can be easily changed between master and slave.

Easy transition from independent volume mirrors into consistency group mirror
The IBM XIV Storage System allows for easy configuration of consistency group mirrors, easy addition of mirrored volumes into a mirrored consistency group, and easy removal of a volume from a mirrored consistency group while preserving mirroring for such volume.

Control over synchronization rates per target
The asynchronous mirroring implementation enables administrators to configure different system mirroring rates with each target system.

Comprehensive monitoring and events
IBM XIV systems generate events and monitor critical asynchronous mirroring-related processes to produce important data that can be used to assess the mirroring performance.

Easy automation via scripts
All asynchronous mirroring commands can be automated through scripts.

Asynchronous remote mirroring terminology

Mirror coupling (sometimes referred to as *coupling*)
A pairing of storage peers (either volumes or consistency groups) that are engaged in a mirroring relationship.

Master and slave
The roles that correspond with the source and target storage peers for data replication in a mirror coupling. These roles can be changed by a system administrator after a mirror is created to accommodate customer needs and support failover and failback scenarios. A valid mirror can have only one master peer and only one slave peer.

Peer designation
A user-configurable mirroring attribute that describes the designation associated with a coupling peer. The master is designated by default as primary and the slave is designated by default as secondary. These values

serve as a reference for the original peer's designation regardless of any role change issued after the mirror is created, but should not be mistaken for the peer's role (which is either master or slave).

Last replicated snapshot
A snapshot that represents the latest state of the master that is confirmed to be replicated to the slave.

Most recent snapshot
A snapshot that represents the latest synchronized state of the master that the coupling can revert to in case of disaster.

Sync job
The mirroring process responsible for replicating any data updates recorded on the master since the last replicated snapshot was taken. These updates are replicated to the slave.

Schedule
An administrative object that specifies how often a sync job is created for an associated mirror coupling.

Interval
A schedule parameter that indicates the duration between successive sync jobs.

RPO Recovery Point Objective – an objective for the maximal data synchronization gap acceptable between the master and the slave. An indicator for the tolerable data loss (expressed in time units) in the event of failure or unavailability of the master.

RTO Recovery Time Objective - an objective for the maximal time to restore service after failure or unavailability of the master.

Specifications

The following specifications apply to the mirroring operation:

Minimum link bandwidth
10Mbps.

Recommended link bandwidth
20Mbps and up.

Maximum round trip latency
250ms.

Attaching XIV systems for mirroring
The connection between two XIV systems has to pass via SAN.

Technological overview

The IBM XIV Storage System asynchronous mirroring blends existing and new technologies.

The asynchronous mirroring implementation is based on snapshots and features the ability to establish automatic and manual mirroring with the added flexibility to assign each mirror coupling with a different RPO. The ability to specify a different schedule for each mirror independently from the RPO helps accommodate special mirroring prioritization requirements without subjecting all mirrors to the same mirroring parameters. The paragraphs below detail the following IBM XIV asynchronous mirroring aspects, technologies, and concepts:

- The replication scheme
- The snapshot-based technology
- IBM XIV asynchronous mirroring special snapshots
- Initializing IBM XIV asynchronous mirroring
- The mirroring replication unit: the sync job
- Mirroring schedules and intervals
- The manual (ad-hoc) sync job
- Determining mirror state through the RPO
- Mirrored consistency groups
- IBM XIV asynchronous mirroring and pool space depletion

Replication scheme

IBM XIV asynchronous mirroring supports establishing mirroring relationships between an IBM XIV Storage System and other XIV systems.

Each of these relationships can be either synchronous or asynchronous and a system can concurrently act as a master in one relationship and act as the slave in another relationship. There are also no practical distance limitations for asynchronous mirroring – mirroring peers can be located in the same metropolitan area or in separate continents.

Each IBM XIV Storage System can have mirroring relationships with other XIV storage systems. Multiple concurrent mirroring relationships are supported with each target.

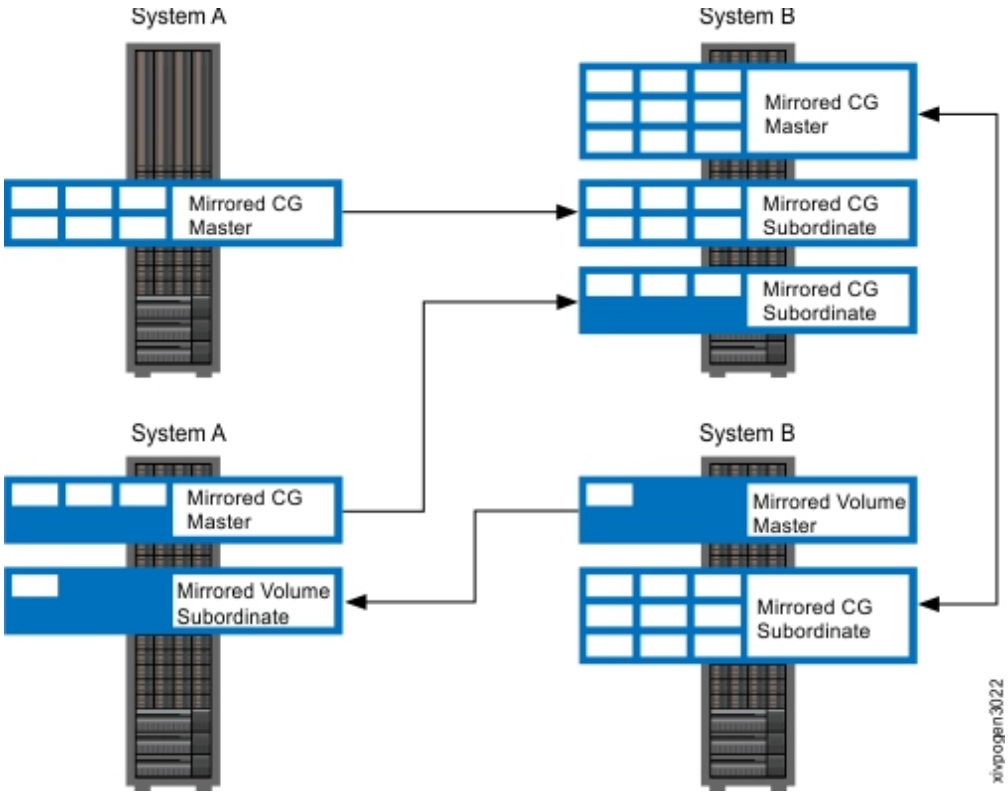


Figure 21. The replication scheme

Snapshot-based technology

IBM XIV features an innovative snapshot-based technology for asynchronous mirroring that facilitates concurrent mirrors with different recovery objectives.

With IBM XIV asynchronous mirroring, write order on the master is not preserved on the slave. As a result, a snapshot taken of the slave at any moment is most likely inconsistent and therefore not valid. To ensure high availability of data in the event of a failure or unavailability of the master, it is imperative to maintain a consistent replica of the master that can ensure service continuity.

This is achieved through XIV snapshots. XIV asynchronous mirroring employs snapshots to record the state of the master, and calculates the difference between successive snapshots to determine the data that needs be copied from the master to the slave as part of a corresponding replication process. Upon completion of the replication process, a snapshot is taken of the slave and reflects a valid replica of the master.

Below are select technological properties that explain how the snapshot-based technology helps realize effective asynchronous mirroring:

- XIV's support for a practically unlimited number of snapshots facilitates mirroring of complete systems with practically no limitation on the number of mirrored volumes supported
- XIV implements memory optimization techniques that further maximize the performance attainable by minimizing disk access.

Mirroring-special snapshots

The *status* and scope of the synchronization process is determined through the use of snapshots.

The following two special snapshots are used:

most_recent snapshot (MRS)
: This snapshot is the most recent snapshot taken of the master (being either a volume or consistency group), prior to the creation of a new replication process (Sync Job – see below). This snapshot exists only on the master.

last_replicated snapshot (LRS)
: This is the most recent snapshot that is confirmed to have been fully replicated to the slave. Both the master and the slave have this snapshot. On the slave, the snapshot is taken upon completion of a replication process, and replaces any previous snapshot with that name. On the master, the most_recent snapshot is renamed last_replicated after the slave is confirmed to have a corresponding replica of the master's most_recent snapshot.

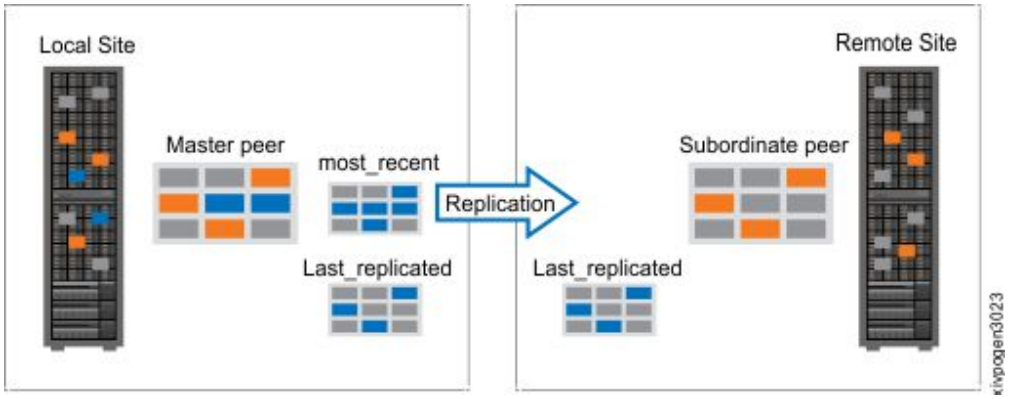


Figure 22. Location of special snapshots

XIV maintains three snapshots per mirror coupling: two on the master and one on the slave. A valid (recoverable) state of the master is captured in the last_replicated snapshot, and on an identical snapshot on the slave. The most_recent snapshot represents a recent state of the master that needs be replicated next to the slave. The system determines the data to replicate by comparing the master's snapshots.

Initializing the mirroring

An XIV mirror is easily created using the CLI or GUI. First, the mirror is created and activated, then an initialization phase starts.

XIV mirrors are created in standby state and must be explicitly activated. During the Initialization phase, the system generates a valid replica of the state of the master on the slave. Until the Initialization is over, there is no valid replica on the slave to help recover the master (in a case of disaster). Once the Initialization phase ends, a snapshot of the slave is taken. This snapshot represents a valid replica of the master and can be used to restore a consistent state of the master in disaster recovery scenarios.

The Initialization takes the following steps (all part of an atomic operation):

The master start initializing the slave
When a new mirror is defined, a snapshot of the master is taken. This snapshot represents the initial state of the master prior to the issuing of the mirror. The objective of the Initialization is to reflect this state on the slave.

Initialization finishes
Once the Initialization finishes, an ongoing mirroring commences through a sync job.

Acknowledgment
The slave acknowledges the completion of the Initialization to the master.

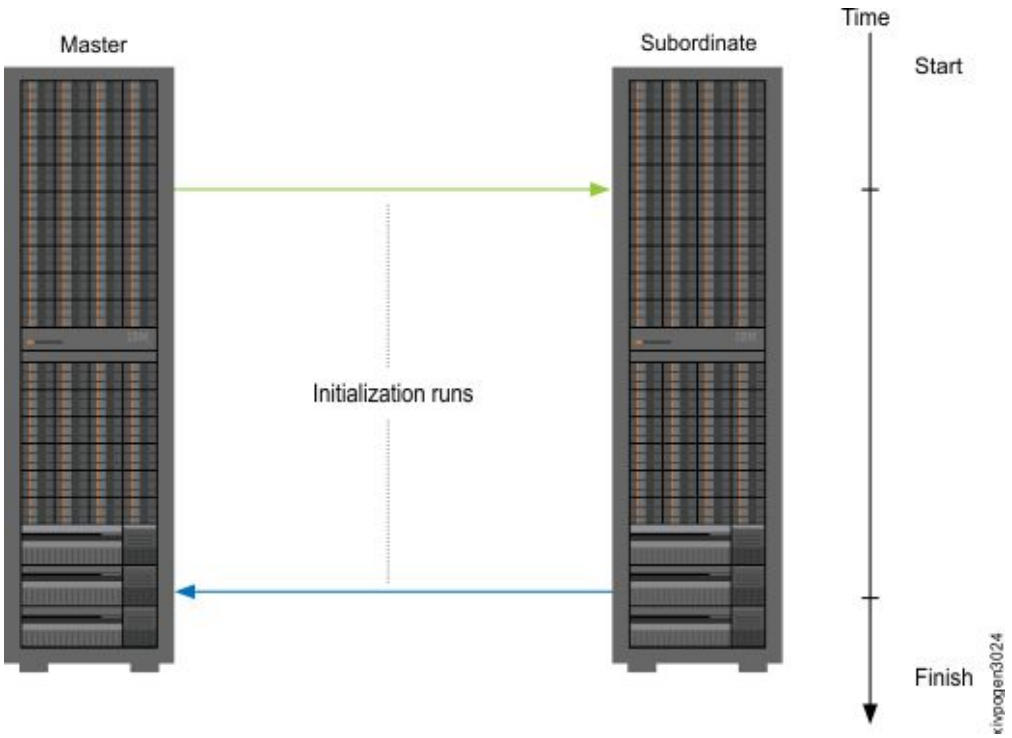


Figure 23. Asynchronous mirroring over-the-wire initialization

Off-line replicating the master onto the slave

The IBM XIV Storage System allows for this volume replica to be transferred off-line to the slave.

At the beginning of the Initialization of the mirror, the user states which volume will be replicated from the master to the slave. This replica of the master is typically much larger than the schedule-based replicas that accumulate differences that are made during a small amount of time. The IBM XIV Storage System allows for this volume replica to be transferred off-line to the slave. This method of transfer is sometimes called "Truck Mode" and is accessible through the *mirror_create* command.

Off-line initialization of the mirror replicates the master onto the slave without being required to utilize the inter-site link. The off-line replication requires:

- Specifying the volume to be mirrored.
- Specifying the initialization type to the mirror creation command.
- Activating the mirroring.

Meeting the above requirements, the system takes a snapshot of the master, and compares it with the slave volume. Only areas where differences are found are replicated as part of the initialization. Then, the slave peer's content is checked against the master and not just automatically considered a valid replica. This check optimizes the initialization time through taking into consideration the available bandwidth between the master and slave and whether the replica is identical to the master volume (that is, there where no writes to the master during the initialization).

The sync job

Data synchronization between the master and slave is achieved through a process run by the master called a sync job.

The sync job updates the slave with any data that was recorded on the master since the latest sync job was created. The process can either be started automatically based on a user-configurable schedule, or manually based on a user-issued command.

When the sync Job is started, a snapshot of the master's state at that time is taken (the most_recent_snapshot).

After any outstanding sync job are completed, the system calculates the data differences between this snapshot and the most recent master snapshot that corresponds with a consistent replica on the slave (the last_replicated_snapshot). This difference constitutes the data to be replicated next by the sync job.

The replication is very efficient because it only copies data differences between mirroring peers. For example, if only a single block was changed on the master, only a single block will be replicated to the slave.

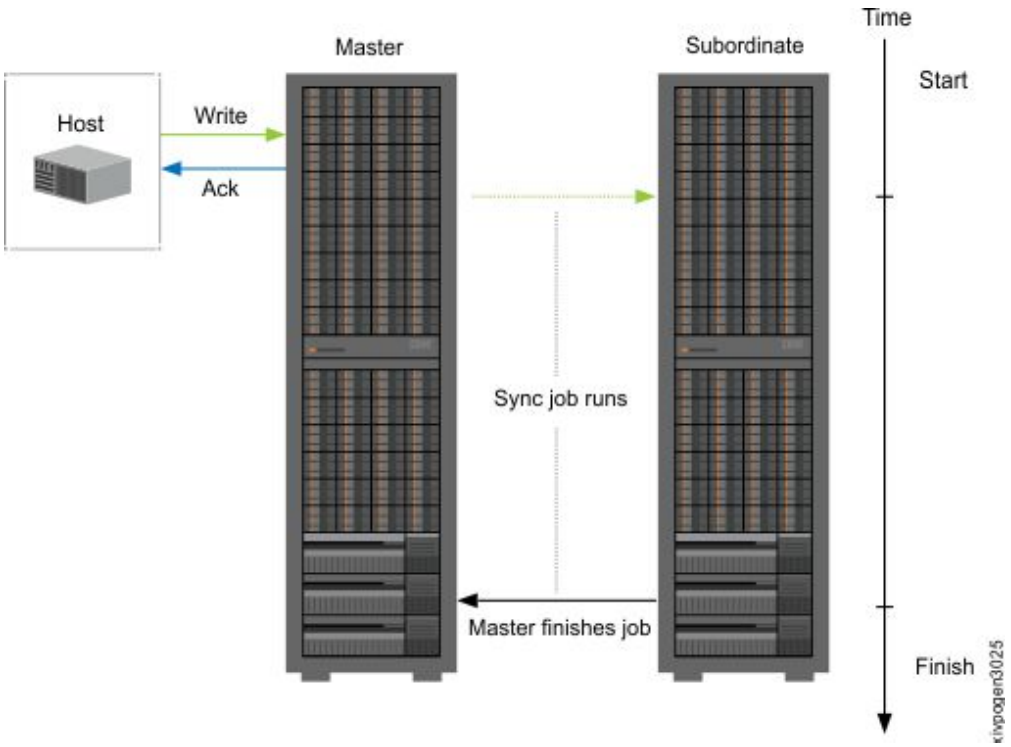


Figure 24. The asynchronous mirroring sync job

Mirroring schedules and intervals

The IBM XIV Storage System implements a scheduling mechanism that is used to drive a recurring asynchronous mirroring process.

Each asynchronous mirror has a specified schedule, and the schedule's interval indicates how often a sync job is created for that mirror.

Asynchronous mirroring has the following features:

- The schedule concept. A *schedule* specifies an interval for automatic creation of sync jobs; a new sync job is normally created at the arrival of a new interval.
- A sync job is not created if another scheduled sync job is running when a new interval arrives
- Custom schedules can be created by users
- Schedule intervals can be set to any of the following values: 30 seconds, 1 min, 2 min, 5 min, 10 min, 15 min , 30 min , 1 hour, 2 hours, 3 hours, 6 hours, 8 hours, 12 hours. The schedule start hour is 00:00.

  Note: The IBM XIV Storage System offers a built-in, non-configurable schedule called min_interval with a 20s interval. It is only possible to specify a 20s schedule using this predefined schedule.
- When creating a mirror, two schedules are specified - one per peer. The slave's schedule can help streamline failover scenarios - controlled by either XIV or a 3rd party process.
- A single schedule can be referenced by multiple couplings on the same system.
- Sync job creation for mirrors with the same schedule takes place at exactly the same time. This is in contrast with mirrors having different schedules with the same interval. Despite having the same interval, sync jobs for these types of mirrors are not guaranteed to take place at the same time.
- A unique schedule called Never is provided to indicate that no sync jobs are automatically created for the pertinent mirror (see below).

Schedules are local to a single XIV system

Schedules are local to the XIV system where they are defined and are set independently of system-to-system relationships. A given source-to-target replication schedule does not mandate an identical schedule defined on the target for reversed replication. To maintain an identical schedule for reverse replication (if the master and slave roles need be changed), independent identical schedules must be defined on both peers.

Schedule sensitivity to timezone difference

The schedules of the peers of a mirroring couple have to be defined in a way that won't be impacted from timezone differences. For example, if the timezone difference between the master and slave sites is two hours, the interval is 3 hours and the schedule on one peer is (12PM, 3AM, 6AM,...), the schedule on the other peer needs to be (2AM, 5AM, 8AM). Although some cases do not call for shifted schedules (for example, a timezone difference of 2 hours and an interval of one hour), this issue can't be overlooked.

The master and the slave also have to have their clocks synchronized (for example using NTP). Avoiding such a synchronization could hamper schedule-related measurements, mainly RPO.

The Never schedule

The system features a special, non-configurable schedule called Never that denotes a schedule with no interval. This schedule indicates that no sync jobs are automatically created for the mirror so it is only possible to issue replication for the mirror through a designated manual command.

Note: A manual snapshot mirror can be issued to every mirror that is assigned a user-defined schedule.

The mirror snapshot (ad-hoc sync job)

You can manually issue a dedicated command to run a mirror snapshot, in addition to using the schedule-based option.

This type of mirror snapshot can be issued for a coupling regardless of whether it has a schedule. The command creates a new snapshot on the master and manually initiates a sync job that is queued behind outstanding sync jobs.

The mirror snapshot:

- Accommodates a need for adding manual replication points to a scheduled replication process.
- Creates application-consistent replicas (in cases where consistency is not achieved via the scheduled replication).

The following characteristics apply to the manual initiation of the asynchronous mirroring process:

- Multiple mirror snapshot commands can be issued – there is no maximum limit on the number of mirror snapshots that can be issued manually.
- A mirror snapshot running when a new interval arrives delays the start of the next interval-based mirror scheduled to run, but does not cancel the creation of this sync job.
  - The interval-based mirror snapshot will be canceled only if the running snapshot mirror (ad-hoc) has never finished.

Other than these differences, the manually initiated sync job is identical to a regular interval-based sync job.

Determining replication and mirror states

The mirror state indicates whether the master is mirrored according to objectives that are specified by the user.

As asynchronous mirroring endures a gap that may exist between the master and slave states, the user must specify a restriction objective for the mirror – the RPO – or Recovery Point Objective. The system determines the mirror state by examining if the master's replica on the slave. The mirror state is considered to be OK only if the master replica on the slave is newer than the objective that is specified by the RPO.

RPO and RTO

The evaluation of the synchronization status is done based on the mirror's RPO value. Note the difference between RPO and RTO.

RPO Stands for Recovery Point Objective and represents a measure of the maximum data loss that is acceptable in the event of a failure or unavailability of the master.

RPO units

Each mirror must be set an RPO by the administrator, expressed in time units. Valid RPO values range between 30 seconds and 24 hours. An RPO of 60 seconds indicates that the slave's state should not be older than the master's state by more than 60 seconds. The

system can be instructed to alert the user if the RPO is missed, and the system's internal prioritization process for mirroring is also adjusted.

RTO Stands for Recovery Target Objective and represents the amount of time it takes the system to recover from a failure or unavailability of the master.

The mirror's RTO is not administered in XIV.

Mirror status values

The mirror status is determined based on the mirror state and the mirroring status.

During the progress of a sync job and until it completes, the slave replica is inconsistent because write order on the master is not preserved during replication. Instead of reporting this state as inconsistent, the mirror state is reported based on the timestamp of the slave's last_replicated snapshot as one of the following:

RPO_OK
Synchronization exists and meets its RPO objective.

RPO_Lagging
Synchronization exists but lags behind its RPO objective.

Initializing
Mirror is initializing.

Definitions of mirror state and status:

The mirror status is determined based on the mirror state and the mirroring status.

Mirror state
During the progress of a sync job and until it completes, the slave replica is inconsistent because write order on the master is not preserved during replication. Instead of reporting this state as inconsistent, the mirror state is reported based on the timestamp of the slave's last_replicated snapshot as one of the following:

Definition of RPO_OK
Synchronization exists and meets its RPO objective.

Definition of RPO_Lagging
Synchronization exists but lags behind its RPO objective.

Initializing
Mirror is initializing.

Mirroring status
The mirroring status denotes the status of the replication process and reflects the activation state and the link state.

Effective recovery currency
Measures as the delta between the current time and the timestamp of the last_replicated_snapshot

Declaring on RPO_OK
The effective recovery currency is positive.

RPO_OK

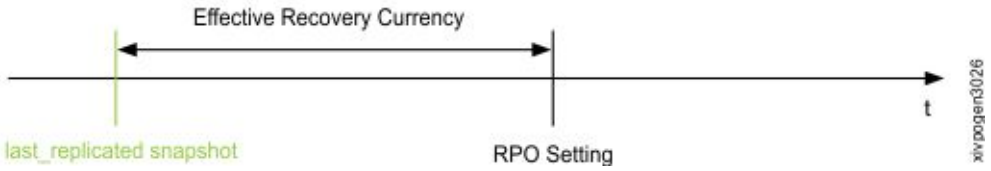


Figure 25. The way RPO_OK is determined

Declaring on RPO_Lagging
The effective recovery currency is negative.

RPO_Lagging

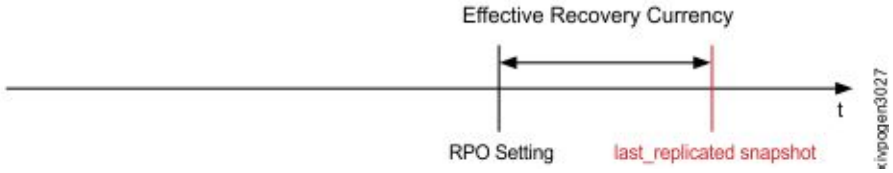


Figure 26. The way RPO_Lagging is determined

Determining mirror status:

The mirroring status denotes the status of the replication process and reflects the activation state and the link state.

The following example portrays the way the mirroring status is determined. The time axis denotes time and the schedule of the sync jobs (t_0 - t_5). Both RPO states are displayed in red and green at the upper section of the image.

First sync job - RPO is OK

Time: t_0
A sync job starts. RPO_OK is maintained as long as the sync job ends before t_1.

Time: t_a
As the sync job ends at t_a, prior to t_1, the status is RPO_OK.

Effective recovery currency
During the sync job run the value of effective recovery currency (the black graph on the upper section of the image) changes. This value goes up as we are getting farther from t_0, goes down - to the RPO setting - once the sync job complete, and does not resume climbing as long as the next schedule has arrived.

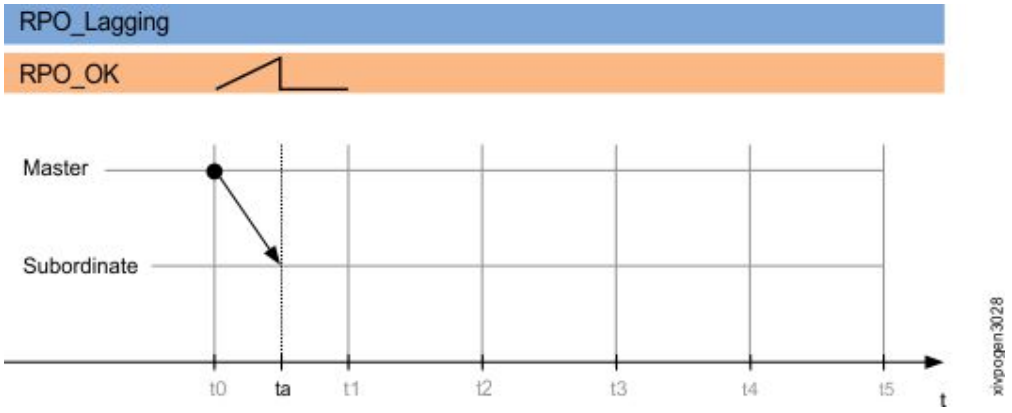


Figure 27. Determining the asynchronous mirroring status – example part 1

Second sync job - RPO is lagging

Time: t_1
A sync job starts. RPO_OK is maintained as long as the sync job ends before t_2.

Time: t_2
The sync job should have ended at this point, but it is still running.

The sync job the was scheduled to run on this point in time is cancelled.

Time: t_b
As the sync job ends at t_b, which is after t_2, the status is RPO_Lagging.

Effective recovery currency
The value of effective recovery currency k.jpg climbing as long as the next sync job hasn't finished.

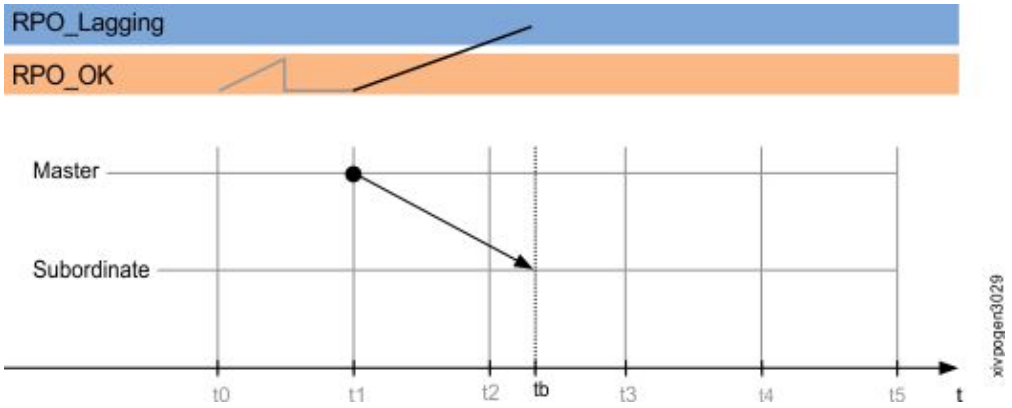


Figure 28. Determining the asynchronous mirroring status – example part 2

Third sync job - RPO is OK

Time: t_3
A new sync job starts. At this point the status is RPO_Lagging.

Time: t_c
As the sync job ends prior to t_4, the status is RPO_OK.

Effective recovery currency
The value of effective recovery currency k.jpg climbing until the next sync

job has finished (this happens at t_c). This value immediately returns to the RPO setting until the time of the next schedule.

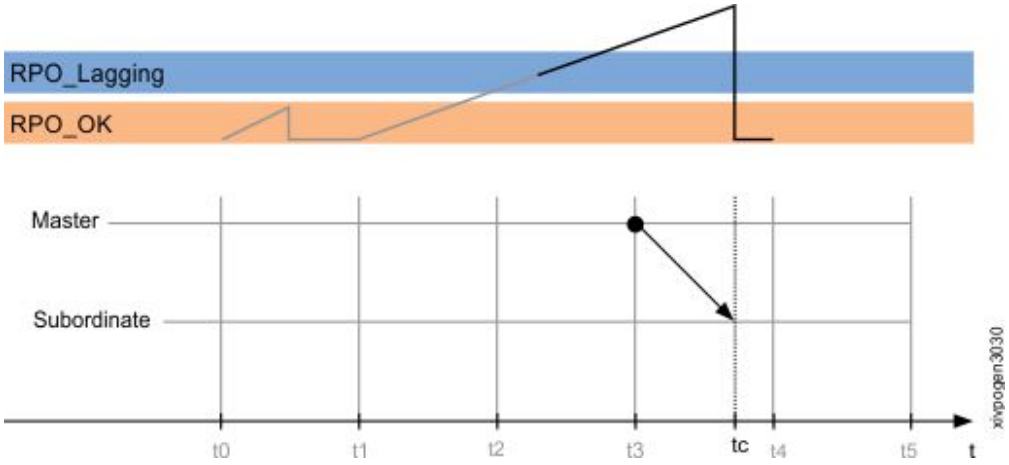


Figure 29. Determining Asynchronous mirroring status – example part 3

The added-value of multiple RPOs

The system bases its internal replication prioritization on the mirror RPO; hence, support for multiple RPOs corresponding with true recovery objectives helps optimize the available bandwidth for replication.

The added-value of multiple Schedules

You can attain a target RPO using multiple schedule interval options. A variable schedule that is decoupled from the RPO helps optimize the replication process to accommodate RPO requirements without necessarily modifying the RPO.

Mirrored consistency groups

IBM XIV enables mirrored consistency groups and mirroring of volumes to facilitate the management of mirror groups.

Asynchronous mirroring of consistency groups is accomplished by taking a snapshot group of the master consistency group in the same manner employed for volumes, either based on schedules, or manually through a dedicated command option.

The peer synchronization and status are managed on a consistency group level, rather than on a volume level. This means that administrative operations are carried out on the whole consistency group, rather than on a specific volume within the consistency group. This includes operations such as activation, and mirror-wide settings such as a schedule.

The synchronization status of the consistency group reflects the combined status of all mirrored volumes pertaining to the consistency group. This status is determined by examining the (system-internally-kept) status of each volume in the consistency group. Whenever a replication is complete for all volumes and their state is RPO_OK, the consistency group mirror status is also RPO_OK. On the other hand, if the replication is incomplete or any of the volumes in a mirrored consistency group has the status of RPO_Lagging, the consistency group mirror state is also RPO_Lagging.

Storage space required for the mirroring

IBM XIV enables to manage the storage required for the mirroring on thin-provisioned pools on both the master and the slave.

Throughout the course of the mirroring, the last_replicated and most_recent snapshots may exceed the space allocated to the volume and its snapshots. The lack of sufficient space on the master can prevent host writes, where the lack of space on the slave can disrupt the mirroring process itself.

IBM XIV enables to manage the storage required for the mirroring on thin-provisioned pools. This way, The IBM XIV Storage System manages and allocates space according to the schemes described in the“Thin provisioning” on page 24 chapter.

Upon depletion of space on each of the peers, the “Pool space depletion” mechanism takes effect.

Pool space depletion

Pool space depletion is a mechanism that takes place whenever the mirroring can no longer be maintained due to lack of space for incoming write requests issued by the host.

Whenever a pool does not have enough free space to accommodate the storage requirements warranted by a new host write, the system runs a multi-step procedure that progressively deletes snapshots within that pool until enough space is made available for a successful completion of the write request.

This multi-step procedure is progressive, meaning that the system proceeds to the next step only if following the execution of the current step, there is still insufficient space to support the write request.

Protecting snapshots using deletion priority:

Protected snapshots have precedence over other snapshots during the pool space depletion process.

The concept of protected snapshots assigns the storage pool with an attribute that is compared with the snapshots' auto-deletion priority attribute. Whenever a snapshot has a deletion priority that is higher that the pool's attribute, it is considered protected.

For example, if the deletion priority of the depleting storage is set to 3, the system will delete snapshots with the deletion priority of 4. Snapshots with priority level 1, 2 and 3 will not be deleted.

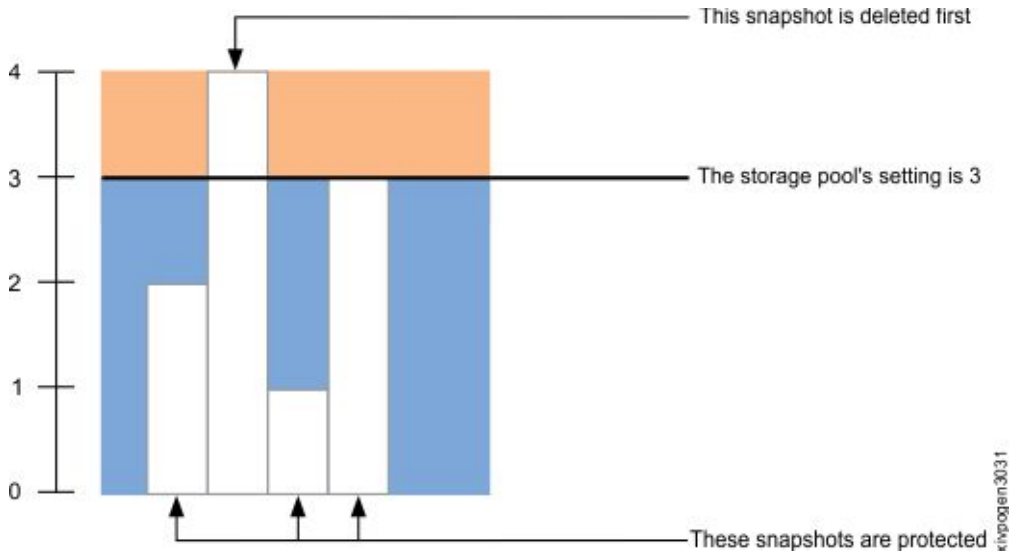


Figure 30. The deletion priority of the depleting storage is set to 3

If the deletion priority of the depleting storage is set to 4, the system will deactivate mirroring before deleting any snapshots.

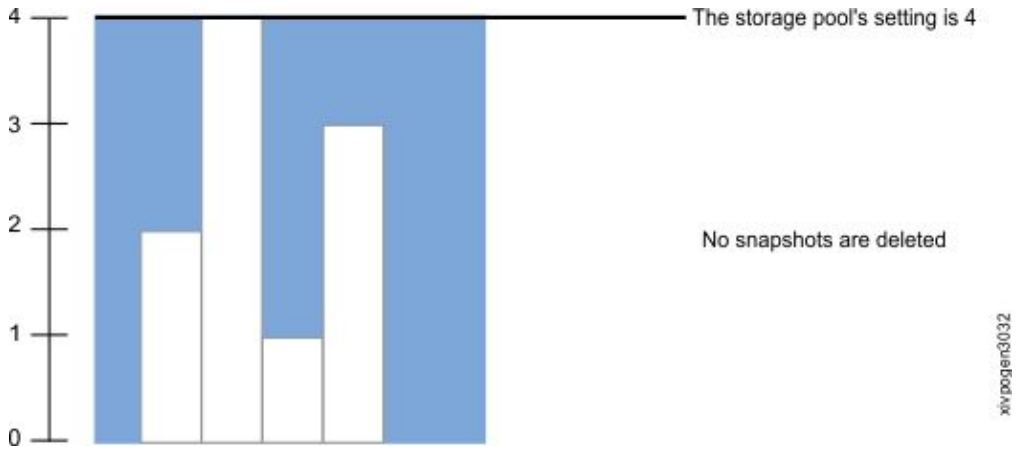


Figure 31. The deletion priority of the depleting storage is set to 4

If the deletion priority of the depleting storage is set to 0, the system can delete any snapshot regardless of deletion priority.

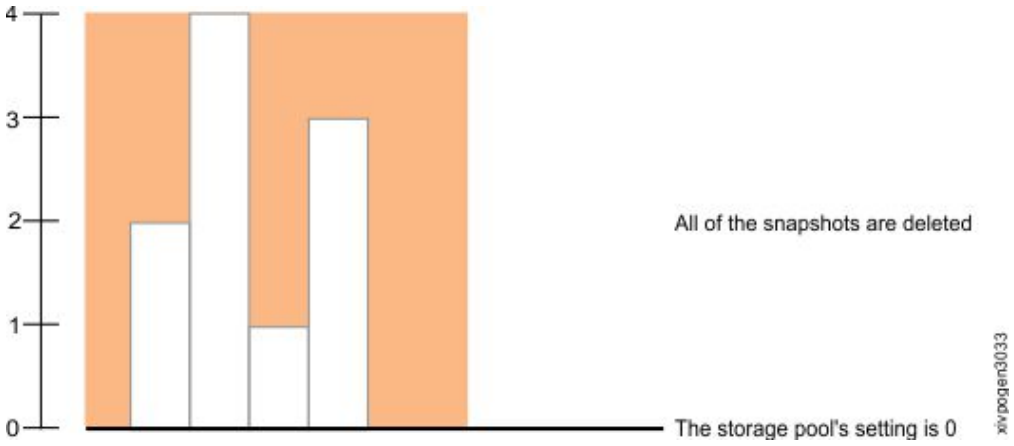


Figure 32. The deletion priority of the depleting storage is set to 0

Deletion priority conventions

Protecting the deletion priority of the last replicated snapshot
The deletion priority of mirror-related snapshots is set implicitly by the system and cannot be customized by the user (see below).

Last replicated and most recent snapshots
The deletion priority of the asynchronous mirroring last_replicated and most_recent snapshots on the master is set to 1.

Last replicated snapshot on the slave
The deletion priority of the last_replicated snapshot on the slave and the is set to 0 (see below).

Default value of the snapshot protecting CLI
By default, the value of the protected_snapshot_priority parameter of the pool_config_snapshots command is 0.

Changing this value
If the protected_snapshot_priority parameter is changed, the system and user-created snapshots with a deletion priority nominally equal or lower than the protected setting will be deleted only after the internal mirroring snapshots are .

For example, if the protected_snapshot_priority is changed to 1, all system and user⌂created snapshots with deletion priority 1 (which includes ALL snapshots created by the user, assuming that their deletion priority was not changed) will be protected and will be deleted only after internal mirroring snapshots are.

Other snapshots
Non mirroring-related snapshots are created by default with a deletion priority 1.

Protecting the last replicated snapshots

The last replicated snapshots represent a consistent replica of the master in asynchronous mirroring. Both the master and the slave have a last replicated snapshot, however, these two snapshots are protected differently.

LRS_{slave}
The slave must have an available consistent copy of the master at all times, where the master does not have to have such availability (as the LRS_{master}

itself is regarded consistent). As a result, this snapshot is never deleted. Upon pool space depletion on the slave, whenever there is no space for the mirroring process, the pool will be locked.

The deletion priority of the LRS on the slave is 0.

LRS_{master}
The last replicated snapshot on the master is available for deletion during pool space depletion.

The deletion priority of the LRS on the master is 1.

Pool space depletion on the master:

The depletion procedure on the master takes the following steps.

Step 1 - deletion of unprotected snapshots

The following snapshots are deleted:

- Regular (not related to mirroring) snapshots
- Snapshots of the mirroring processes that are no longer active
- The snapshot of any snapshot mirror (ad hoc sync job) that has not started yet

The deletion is subject to the deletion priority of the individual snapshot. In the case of deletion priority clash, older snapshots are deleted first.

Success criteria:
The user reattempts operation, re-enables mirroring and resumes replication. If this fails, the system proceeds to step 2 (below).

Step 2 - deletion of the snapshot of any outstanding (pending) scheduled sync job
If replication still does not resume after the actions taken on step 1:

The following snapshots are deleted:

- All snapshots that were not deleted in step 1.

Success criteria:
The system reattempts operation, re-enables mirroring and resumes replication.

Step 3 - automatic deactivation of the mirroring and deletion of the snapshot designated as the mirror most_recent snapshot

If the replication still does not resume:

The following takes place:

- An automatic deactivation of the mirroring
- Deletion of the most_recent snapshot
- An event is generated.

Ongoing ad-hoc sync job
The snapshot created during the ad-hoc sync job is considered as a most_recent snapshot, although it is not named as such and not suplicated with a snapshot in that name. Following the completion of the ad-hoc sync job, and only after this completion, the snapshot is duplicated and the duplicate is named last_replicated.

Upon a manual reactivation of the mirroring process:

1. The mirroring activation state changes to Active

2. A most_recent snapshot is created
3. A new sync job starts

Step 4 - deletion of the last_replicated snapshot

If more space is still required:

The following takes place:

- Deletion of the last_replicated snapshot (on the master)
- An event is generated.

Following the deletion:

1. The mirroring remains deactivated, and must be manually reactivated.
2. The mirroring changes to *change tracking* state. Host I/O to the master are tracked but not replicated
3. The system marks storage areas that were written into since the last_replicated snapshot was created

Step 5 - deletion of the most_recent snapshot that is created when activating the mirroring in *Change Tracking* **state**

If more space is still required:

The following takes place:

- Deletion of the most_recent snapshot (on the master).
- An event is generated.

Following the deletion:

Deletion of this most_recent snapshot in this state leaves the master with neither a snapshot nor a bitmap, mandating full initialization. To minimize the likelihood for such deletion, this snapshot is automatically assigned a special (new) deletion priority level. This deletion priority implies that the system should delete the snapshot only after all other last_replicated snapshots in the pool were deleted. Note that the new priority level will only be assigned to a mirror with a consistent Slave replica and not to a mirror that was just created (whose first state is also initialization).

Step 6 - deletion of protected snapshots

If more space is still required:

The following takes place:

- An event is generated.
- Deletion of all protected snapshots, regardless of the mirroring. These snapshots are deleted according to their deletion priority and age.

Following the deletion:

- The master's state changes to **Init** (distinguished from the Initialization phase mirrors start with)
- The system stops marking new writes
- A most_recent snapshot is created
- The system creates and runs a sync job encompassing all of the tracked changes tracked

- following the completion of this sync job, a last_replicated snapshot is created on the master, and the mirror state changes to rpo_ok or rpo_lagging, as warranted by the effective RPO

If pool space depletes during the Init:

- The master's state remains Initialization
- An event is generated
- The mirroring is deactivated
- The most_recent snapshot is deleted (mandating a Full Initialization)

Upon manual mirroring activation during the Init:

- The master's state remains Initialization
- A most_recent snapshot is created
- The system starts a Full Initialization based on the most_recent snapshot

Pool space depletion on the slave:

Pool space depletion on the slave means that there is no room available for the last_replicated snapshot. In this case, the mirroring is deactivated.

Snapshots with a deletion priority of 0 are special snapshots that are created by the system on the slave peer and are not automatically deleted to free space, regardless of the pool space depletion process. The asynchronous mirroring slave pear has one such snapshot: the last_replicated snapshot.

Asynchronous mirroring process walkthrough

This section walks you through creating an asynchronous mirroring relationship, starting from the initialization all the way through completing the first scheduled sync job.

Step 1

Time is 01:00 when the command to create a new mirror is issued. In this example, an RPO of 120 minutes and a schedule of 60 minutes are specified for the mirror.

The mirroring process must first establish a baseline for ensuing replication. This warrants an Initialization process during which the current state of the master is replicated to the slave peer. This begins with the host writes being briefly blocked (1). The state of the master peer can then be captured by taking a snapshot of the master state: the most_recent snapshot (2), which serves as a baseline for ensuing schedule-based mirroring. After this snapshot is created, host writes are no longer blocked and continue to update the storage system (3). At this time, no snapshot exists on the slave yet.

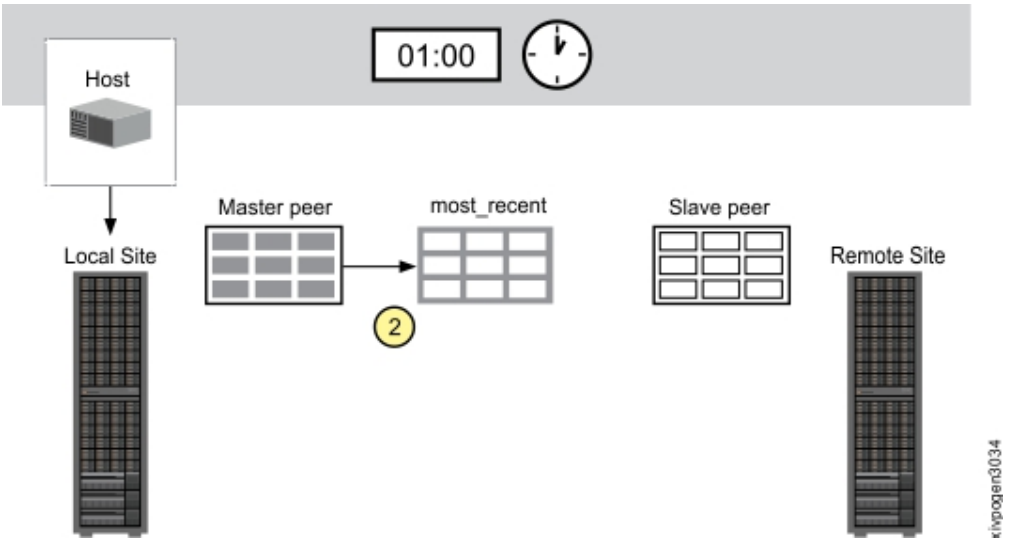


Figure 33. Asynchronous mirroring walkthrough – Part 1

Step 2

After the state of the master is captured, the data that needs to be replicated as part of the Initialization process is calculated. In this example, the master's most_recent snapshot represents the data to be replicated through the first sync job (4).

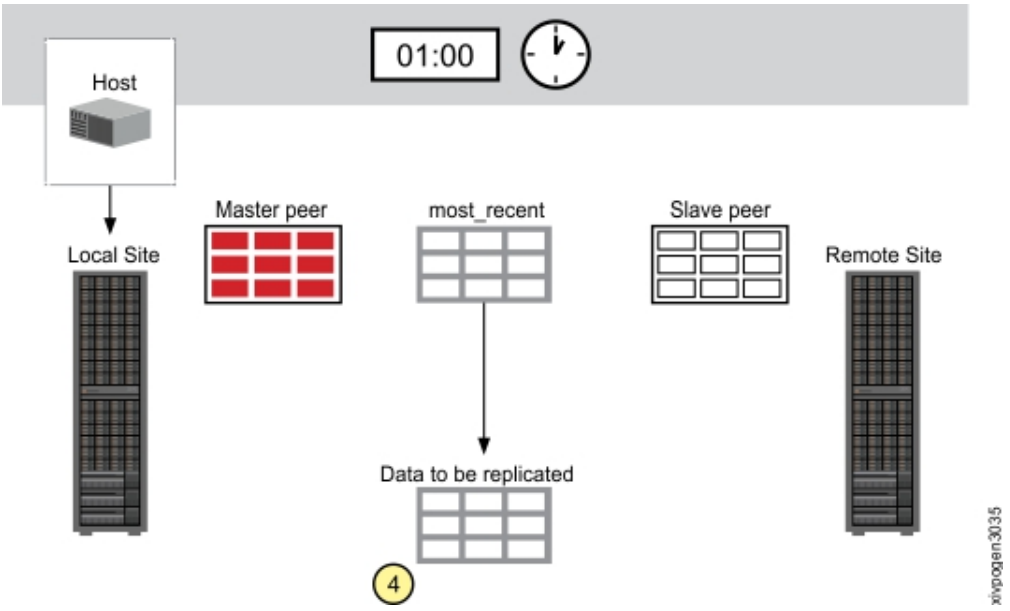


Figure 34. Asynchronous mirroring walkthrough – Part 2

Step 3

During this step, the Initialization Sync Job is well in progress. The master continues to be updated with host writes – the updates are noted in the order they are written – first 1, then 2 and finally 3. The initialization sync job replicates the initial master peer's state to the slave peer (5).

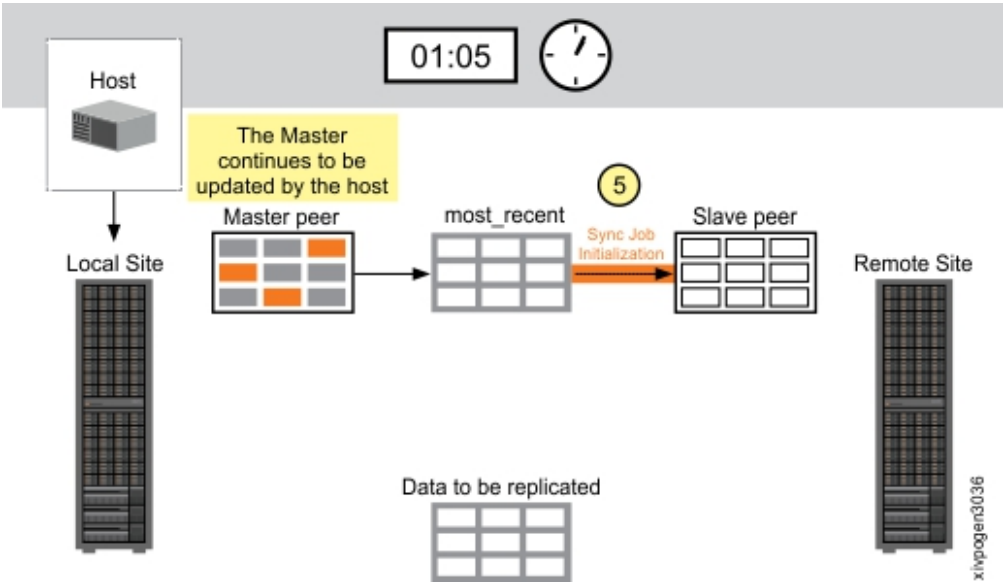


Figure 35. Asynchronous mirroring walkthrough – Part 3

Step 4

Moments later, the initialization sync job completes. After it completes, the slave's state is captured by taking a snapshot: the last_replicated snapshot (6). This snapshot reflects the state of the master as captured in the most_recent snapshot. In this example, it is the state just before the initialization phase started.

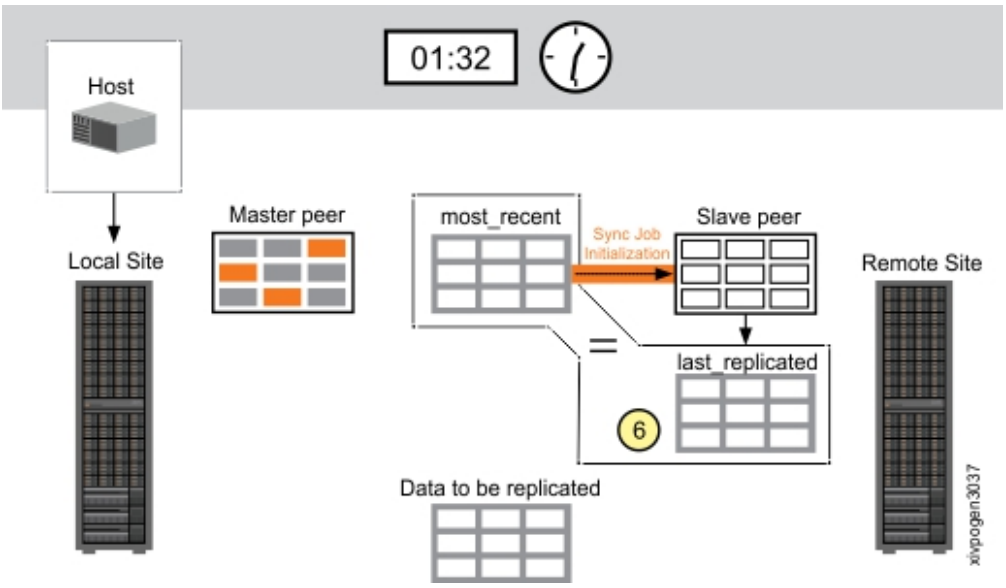


Figure 36. Asynchronous mirroring walkthrough – Part 4

Step 5

During this step, the master's last_replicated snapshot is created. The most_recent snapshot on the master is renamed last_replicated (7) and represents the most recent point-in-time that the master can be restored if needed (because this state is captured in the slave's corresponding snapshot).

When the initialization phase ends, the master and slave peers have an identical restore time point, to which they can be reverted, if needed.

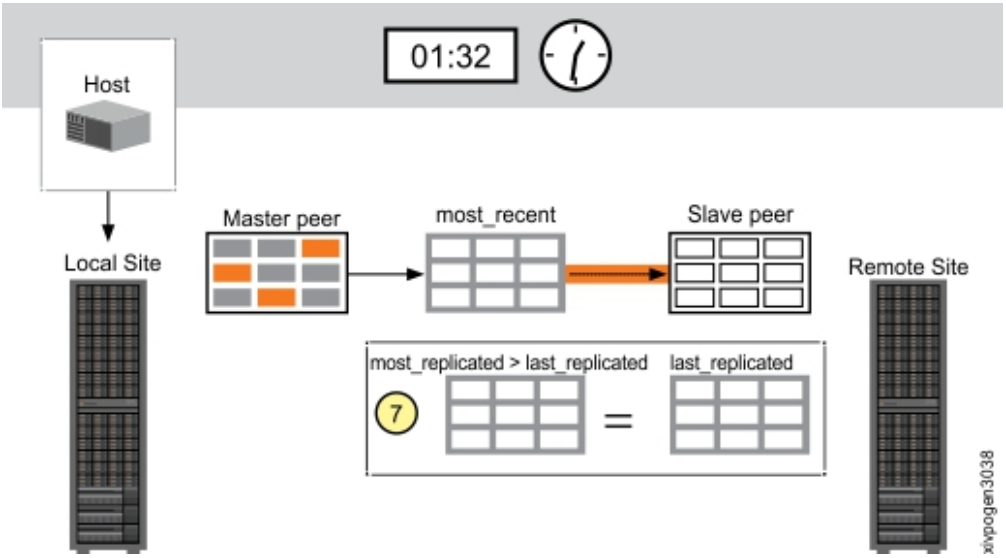


Figure 37. Asynchronous mirroring walkthrough – Part 5

Step 6

Based on the mirror's schedule, a new interval arrives in a manner similar to the Initialization phase: host writes are blocked (1), and a new master most_recent snapshot is created (2), reflecting the master peer's state at this time.

Then, host writes are no longer blocked (3).

The update number (4) occurs after the snapshot is taken and is not reflected in the next sync job. This is shown by the color-shaded cells in the most_recent snapshot figure.

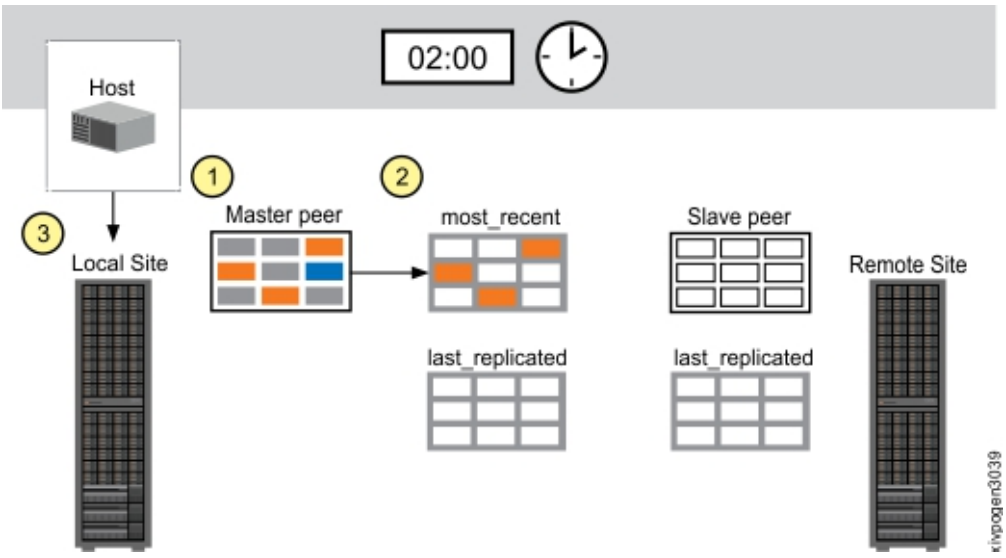


Figure 38. Asynchronous mirroring walkthrough – Part 6

Step 7

A new sync job is set. The data to be replicated is calculated based on the difference between the master's most_recent snapshot and the last_replicated snapshot (4).

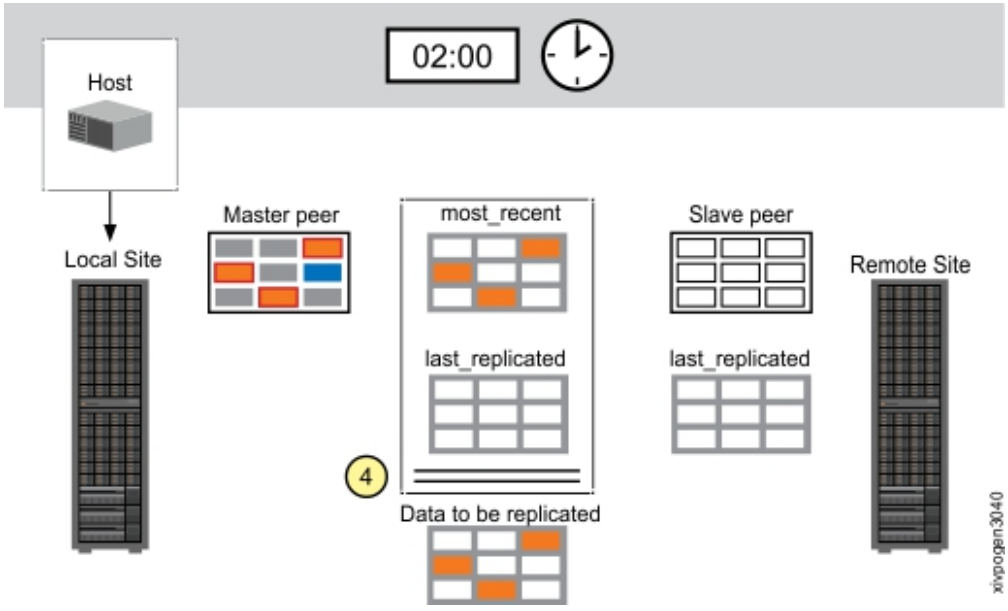


Figure 39. Asynchronous mirroring walkthrough – Part 7

Step 8

The sync job is in process. During the sync job, the master continues to be updated with host writes (update 5).

The sync job data is not replicated to the slave in the order by which it was recorded at the master – the order of updates on the slave is different.

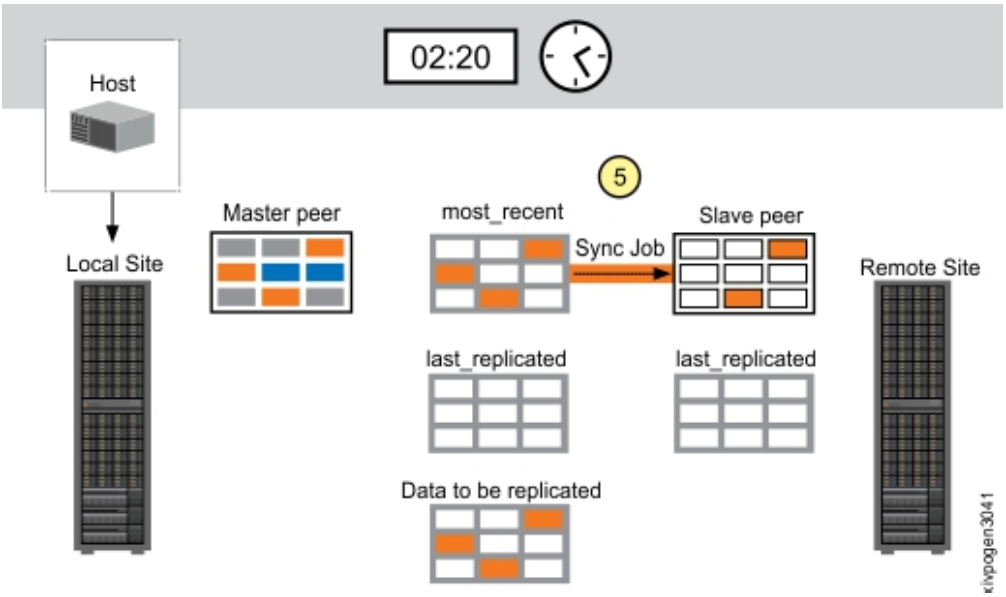


Figure 40. Asynchronous mirroring walkthrough – Part 8

Step 9

The sync job is completed. The slave's last_replicated snapshot is deleted (6) and replaced (in one atomic operation) by a new last_replicated snapshot.

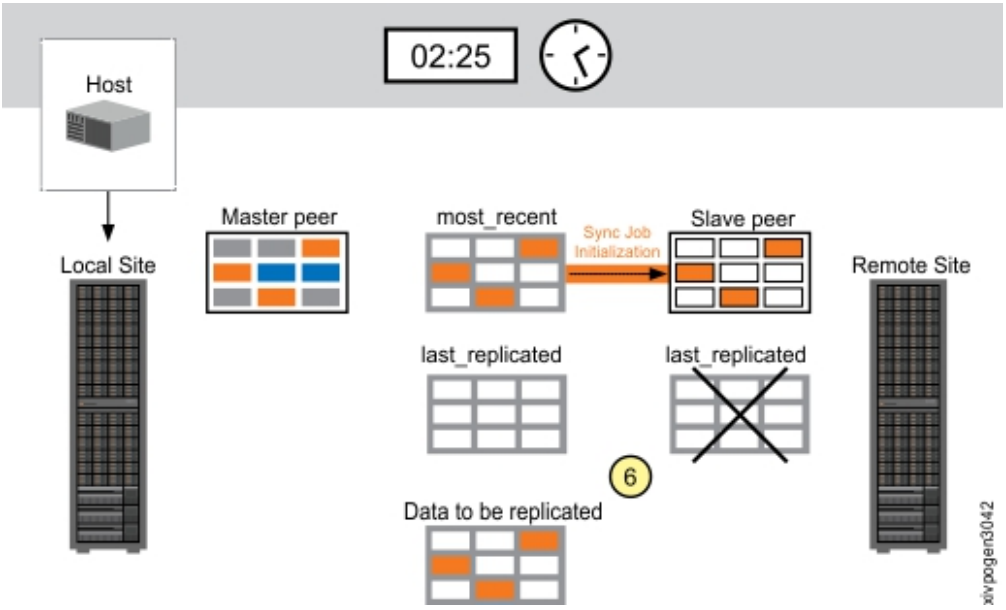


Figure 41. Asynchronous mirroring walkthrough – Part 9

Step 10

The sync job is completed with a new last_replicated snapshot representing the updated slave's state (7).

The slave's last_replicated snapshot reflects the master's state as captured in the most_recent snapshot. In this example, it is the state at the beginning of the mirror schedule's interval.

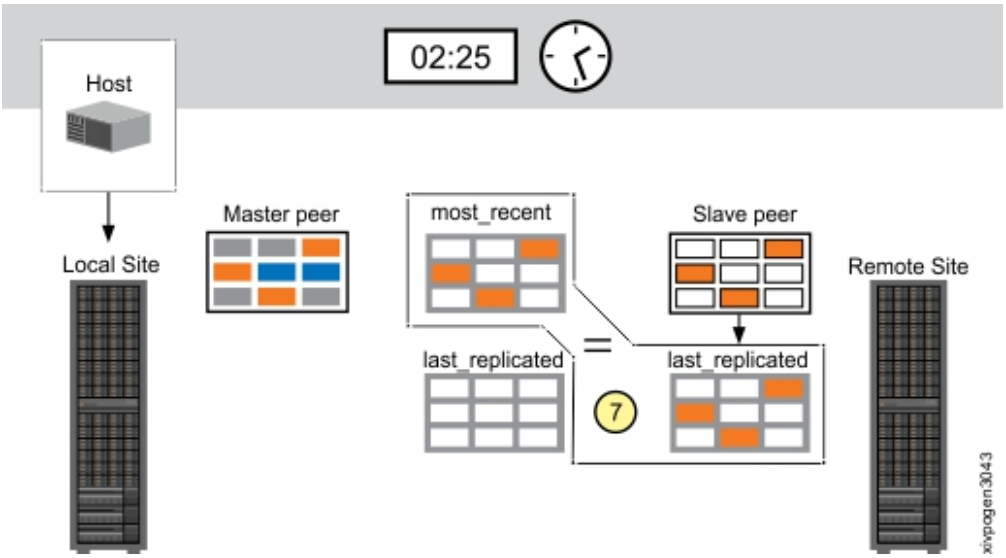


Figure 42. Asynchronous mirroring walkthrough – Part 10

Step 11

A new master last_replicated snapshot created. In one transaction - the current last_replicated snapshot on the master is deleted (8) and the most_recent snapshot is renamed the last_replicated (9).

The interval sync process is now complete - the master and slave both have an identical restore time point to which they can be reverted if needed.

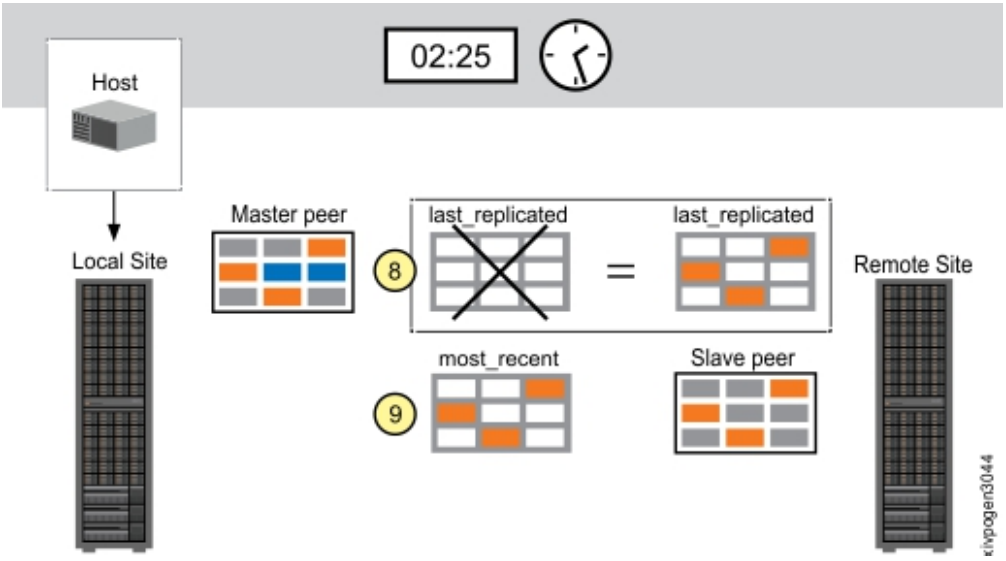


Figure 43. Asynchronous mirroring walkthrough – Part 11

Peers roles

Peers' statuses denote their roles within the coupling definition.

After creation, a coupling has exactly one peer that is set to be the master peer, and exactly one peer that is set to be the slave peer. Each of the peers can have the following available statuses:

None The peer is not part of a coupling definition.

Master
The actual source peer in a replication coupling. This type of peer serves host requests, and is the source for synchronization updates to the slave. A master peer can be changed to a slave directly while in asynchronous mirroring.

Slave The actual target peer in a replication coupling. This type of peer does not serve host requests, and accepts synchronization updates from a corresponding master. A slave can be changed to a master directly while in asynchronous mirroring.

Activating the mirroring

The state of the mirroring is derived from the state of its components.

The remote mirroring process hierarchically manages the states of the entities that participate in the process. It manages the states for the mirroring based on the states of the following components:

- Link
- Activation

The following mirroring states are possible:

Non-operational
The coupling state is defined as non-operational if at least one of the following conditions is met:
- The activation state is standby.
- The link state is error.
- The slave peer is locked.

Operational
All of the following conditions must be met for the coupling to be defined as operational:
- The activation state is active.
- The link is OK.
- The peers have different roles.
- The slave peer is not locked.

Link states

The link state is one of the factors determining the coupling operational status.

The *link state* reflects the connection from the master to the slave. A failed link or a failed slave system both manifest as a link error. The link state is one of the factors determining the coupling operational status.

The available link states are:

OK The link is up and functioning.

Error The link is down.

Activation states

When the coupling is created, its activation is in standby state. When the coupling is enabled, its activation is in active state.

Standby
When the coupling is created, its activation is in standby state.

The synchronization is disabled:

- Sync jobs do not run.
- No data is copied.
- The coupling can be deleted.

Active The synchronization is enabled:

- Sync jobs can be run.
- Data can be copied between peers.

Regardless of the activation state:

- The mirroring type can be changed to synchronous.
- Peer roles can change.

Deactivating the coupling

Deactivating the coupling stops the mirroring process.

The mirroring is terminated by deactivating the coupling, causing the system to:

- Terminate, or delete the mirroring
- Stop the mirroring process as a result of:
  - A planned network outage
  - An application to reduce network bandwidth
  - A planned recovery test

The deactivation pauses a running sync job and no new sync jobs will be created as long as the active state of the mirroring is not restored. However, the deactivation does not cancel the interval-based status check by the master and the slave. The synchronization status of the deactivated coupling is calculated on the start of each interval, as if the coupling was active.

Deactivating a coupling while a sync job is running, and not changing that state before the next interval begins, leads to the synchronization status becoming RPO_Lagging, as described in the following outline. Upon the deactivation:

On the master
The activation state changes to standby; replication pauses (and records where it paused); replication resumes upon activation.

Note: An ongoing sync job resumes upon activation, no new sync job will be created until the next interval.

On the slave
Not available.

Regardless of the state of the coupling:

- Peer roles can be changed

Note: For consistency group mirroring: deactivation pauses all running sync jobs pertaining to the consistency group. It is impossible to deactivate a single volume sync job within a consistency group.

Mirroring consistency groups

Grouping volumes into a consistency group provides a means to maintain a consistent snapshot of the group of volumes at the secondary site.

The following assumptions make sure that consistency group semantics work with remote mirroring:

Consistency group-level management
Mirroring of consistency groups is managed on a consistency group level, rather than on a volume level. For example, the synchronization status of the consistency group is determined after examining all mirrored volumes that pertain to the consistency group.

Starting with an empty consistency group
Only an empty consistency group can be defined as a mirrored consistency group. If you want to define an existing non-empty consistency group as mirrored, the volumes within the consistency group must first be removed from the consistency group and added back only after the consistency group is defined as mirrored.

Adding a volume to an already consistency group
Only mirrored volumes can be added into a mirrored consistency group. This operations requires the following:

- Volume peer is on the same system as the peers of the consistency group
- Volume replication type is identical to the type used by the consistency group. For example, async_interval.
- Volume belongs to the same storage pool of the consistency group
- Volume has the same schedule as the consistency group
- Volume has the same RPO as the consistency group
- Volume and consistency group are in the same synchronization status (SYNC_BEST_EFFORT for synchronous mirroring, RPO OK for asynchronous mirroring)

If the mirrored consistency group is configured with a user-defined schedule, meaning not using the Never schedule:
Mirrored consistency group or volume should not have non-started snapshot mirrors, non-finished snapshot mirrors (ad hoc sync jobs), or both.

If the mirrored consistency group is configured with a Never schedule:
Mirrored consistency group or volume should not have non-started, non-finished snapshot mirrors, non-finished snapshot mirrors (ad hoc sync jobs), or both. The status of the mirrored consistency group shall be Initialization until the next sync job is completed.

Adding a mirrored volume to a non-mirrored consistency group
It is possible to add a mirrored volume to a non-mirrored consistency group, and it will retain its mirroring settings.

A single sync job for the entire consistency group
The mirrored consistency group has a single sync job for all pertinent mirrored volumes within the consistency group.

Location of the mirrored consistency group
All mirrored volumes in a consistency group are mirrored on the same system.

Retaining mirroring attributes of a volume upon removing it from a mirrored consistency group
When removing a volume from a mirrored consistency group, the corresponding peer volume is removed from the peer consistency group. Mirroring is retained (same configuration as the consistency group from which it was removed). Peer volume is also removed from peer consistency group. Ongoing consistency group sync jobs will continue.

Mirroring activation of a consistency group
Activation and deactivation of a consistency group affects all consistency group volumes.

Role updates
Role updates concerning a consistency group affects all consistency group volumes.

Dependency of the volume on its consistency group

- It is not possible to directly activate, deactivate, or update role of a given volume within a consistency group from the UI.
- It is not possible to directly change the interval of a given volume within a consistency group.
- It is not possible to set independent mirroring of a given volume within a consistency group.

Protecting the mirrored consistency group
Consistency group-related commands, such as moving a consistency group, deleting a consistency group and so on, are not allowed as long as the consistency group is mirrored. You must remove mirroring before you can delete a consistency group, even if it is empty.

Setting a consistency group to be mirrored

Volumes added to a mirrored consistency group have to meet some prerequisites.

Volumes that are mirrored together as part of the same consistency group share the same attributes:

- Target
- Pool
- Sync type
- Mirror role
- Schedule
- Mirror state
- Last_replicated snapshot timestamp

In addition, their snapshots are all part of the same last_replicated snapshot group.

To obtain the consistency of these attributes, setting the consistency group to be mirrored is done by first creating a consistency group, then setting it to be mirrored and only then populating it with volumes. These settings mean that

adding a new volume to a mirrored consistency group requires having the volume set to be mirrored exactly as the other volumes within this consistency group, including the last_replicated snapshot timestamp (which entails an RPO_OK status for this volume).

Note: A non-mirrored volume cannot be added to a mirrored consistency group. It is possible, however, to add a mirrored volume to a non-mirrored consistency group, and have this volume retain its mirroring settings.

Creating a mirrored consistency group

The process of creating a mirrored consistency group comprises the following steps.

Step 1 Define a consistency group as mirrored (the consistency group must be empty).

Step 2 Activate the mirror.

Step 3 Add a corresponding mirrored volume into the mirrored consistency group. The mirrored consistency group and the mirrored volume must have the following identical parameters:

- Source and target
- Pools
- Mirroring type
- RPO
- Schedule names (both local and remote)
- Mirror state is RPO_OK
- Mirroring status is Activated

Note: It is possible to add a mirrored volume to a non-mirrored consistency group. In this case, the volume retains its mirroring settings.

Adding a mirrored volume to a mirrored consistency group

After the volume is mirrored and shares the same attributes as the consistency group, you can add the volume to the consistency group after certain conditions are met.

The following conditions must be met:

- The volume is on the same system as the consistency group
- The volume belongs to the same storage pool as the consistency group
- Both the volume and the consistency group do not have outstanding sync jobs, either scheduled or manual (ad hoc)
- The volume and consistency group have the same synchronization status (synchronized="best_effort" and async_interval="rpo_ok")
- The volume's and consistency group's special snapshots, most_recent and last_replicated, have identical timestamps (this is achieved by assigning the volume to the schedule that is utilized by the consistency group)
- In the case that the consistency group is assigned with schedule="never", the status of the consistency group is initialization as long as no sync job has run.

Removing a volume from a mirrored consistency group

Removal of a volume from a mirrored consistency group is easy and preserves volume mirroring.

When you remove a volume from a mirrored consistency group, the corresponding peer volume is removed from the peer consistency group; mirroring is retained with the same configuration as the consistency group from which it was removed. All ongoing consistency group's sync jobs keep running.

Chapter 10. Multi-site mirroring

Multi-site mirroring is an IBM XIV Storage System technology that allows customers to set High Availability and Disaster Recovery solutions over multiple sites, keeping 3 copies of their data.

Key features of multi-site mirroring are:

Concurrent multiple multi-site mirroring

- The IBM XIV approach to multi-site mirroring includes 3 peers with one synchronous and two asynchronous replications among them (one of them at standby).
- Multiple multi-site mirroring configurations run concurrently per system, each with separate mirror peers.
- The source runs 2 concurrent mirrors into 2 different destinations
- Any given system can be represented in several multi-site configurations, each referencing different systems.
- A system can host mirroring peers with different roles in different multi-site configurations.

Extensibility

- Any existing two-way mirroring relation (synchronous or asynchronous) can be extended to three-way mirroring, with no need to disrupt the existing mirror relation.

  Note: Three-way mirroring cannot be configured on a mirrored consistency group, but can be configured on a local consistency group.
- The multi-site mirroring relation is created based on an already existing target connectivity.

Maintainability

- If one mirror of the multi-site mirror fails, the other mirror continues.

Multi-site mirroring terminology

The multi-site mirroring technology introduces some new terms, in addition to those mentioned in the synchronous and asynchronous mirroring chapters.

Master (Source)
The volume that is mirrored.

Substitute master (Secondary source)
The volume that synchronously mirrors the source.

Slave (Destination)
The volume that asynchronously mirrors the source.

Terminology of synchronous and asynchronous mirroring

Some of the concepts discussed on this chapter were introduced in previous mirroring chapters. For a summary of the terminology these chapters use, see here:

- "Remote mirroring basic concepts" on page 57.
- "Asynchronous remote mirroring terminology" on page 77.

Multi-site mirroring technological overview

The IBM XIV enables replication of a volume to two peer volumes that reside on other systems.

Components hierarchy

The multi-site relation clearly defines each system and the role it plays during a disaster recovery scenario.

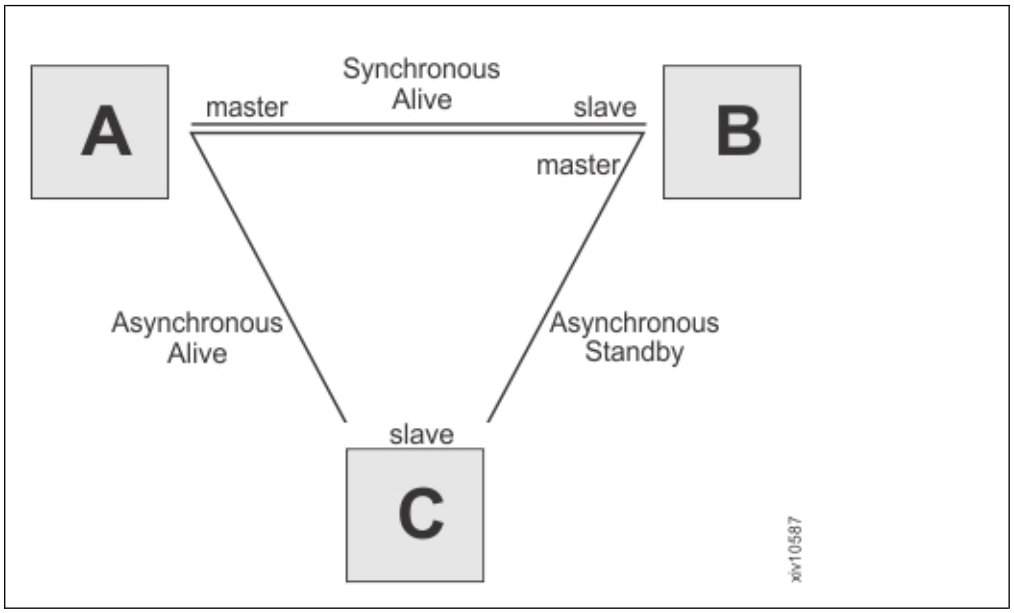


Figure 44. The hierarchy of multi-site mirroring components

The substitute master (System B) is synchronously mirrored with the master (System A), and takes on the role of the master to the slave system (System C) when the master (System A) becomes unavailable.

C is a replica of either A or B. If the A-C mirroring relation is active, then the B-C mirroring is inactive, and vice versa. That is, both A and B cannot write to C at the same time.

A-B mirror
The synchronous mirroring relation between the master and substitute master.

A-C mirror
The asynchronous mirroring relation between the master and slave.

B-C mirror
The asynchronous mirroring relation between the substitute master and the slave.

This mirroring relation is also known as the *Standby* mirror.

The mirroring relation that comprises the B-C mirroring relation can be either of the following types:

- Standby mirror - the third mirror of the multi-site mirroring definition, which is defined in advance

- Live mirror - an operational mirroring relation, which becomes operational only by request in case of disaster recovery

Defining the standby mirroring relation in advance requires that the target connectivity between B and C (or at least its definitions) needs to be in place between all systems when the multi-site mirroring relation is configured.

Table 8 and Figure 44 on page 110 display the roles of each of the systems that participate in the multi-site mirroring relations.

Table 8. The mirroring relations that comprise the multi-site mirroring

System	Role	A-B	A-C	B-C
A	Source	Synchronous mirroring relation. System A is the *master*. The mirror is *active*.	Asynchronous mirroring relation. System A is the *master*. The mirror is *active*.	
B	Secondary source	Synchronous mirroring relation. System B is the *slave*. The mirror is *active*.		Asynchronous mirroring relation. System B is the *master*. The mirror is *standby*.
C	Destination		Asynchronous mirroring relation. System C is the *slave*. The mirror is *active*.	Asynchronous mirroring relation. System C is the *slave*. The mirror is *standby*.

Note: Currently, multi-site mirroring does not support consistency group mirroring. However, it can be supported when volume mirroring is used and the volume is a part of a local consistency group.

Multi-site mirroring states

The IBM XIV multi-site mirroring technology has multiple states, or conditions, of operation. While each individual mirroring definition has its own state, the multi-site mirroring definition has a global state, too.

Global states

The following states are applicable to the global states of the mirrors:

Init All mirroring definitions are ready to start transferring data.

Operational
A steady state where both A-B and A-C are *Active*.

Degraded
If both A-B and A-C are active and synchronized, but A-C is *RPO lagging*, the mirroring state is *degraded*.

Inactive
When both A-B and A-C are *inactive*, the mirroring state is *inactive*.

Compromised

These are possible reason for a *compromised* state:

Disconnection
The link is *down* for either A-B or A-C.

Resync
Either A-B or A-C are in *resync* and the substitute master did not yet take ownership.

Following a partial change of role
There was a role change on either A-B or A-C.

Substitute master and slave states

The following states are applicable to substitute master and slave states:

Connected
The mirror with the master system is in a *connected* state.

Disconnected
The mirror with the master system is in a *disconnected* state.

Standby mirroring states

The following states are applicable to the standby mirror:

Up The standby mirror is defined and connected.

Down The standby mirror is defined and disconnected.

NA The standby mirror is not defined.

Chapter 11. IBM Hyper-Scale Mobility

IBM Hyper-Scale Mobility enables a non-disruptive migration of volumes from one storage system to another.

IBM Hyper-Scale Mobility helps achieve storage management objectives that are otherwise difficult to address. Consider the following scenarios:

- Migrating data out of an over-provisioned system.
- Migrating all the data from a system that will be decommissioned or re-purposed.
- Migrating data to another storage system to achieve adequate (lower or higher) performance, or to load-balance systems to ensure uniform performance.
- Migrating data to another storage system to load-balance capacity utilization.

The IBM Hyper-Scale Mobility process

This section walks you through the IBM Hyper-Scale Mobility process.

Hyper-Scale Mobility moves a volume from one system to another, while the host is using the volume. To accomplish this, I/O paths are manipulated by the storage, without involving host configuration, and the volume identity is cloned on the target system. In addition, direct paths from the host to the target system need to be established, and paths to the original host can finally be removed. Host I/Os are not interrupted throughout the migration process.

The key stages of the IBM Hyper-Scale Mobility and the respective states of volumes are depicted in Figure 45 on page 114 and explained in detail in Table 9 on page 114.

For an in-depth practical guide to using IBM Hyper-Scale Mobility, see the Redbooks publication IBM Hyper-Scale Mobility Overview and Usage.

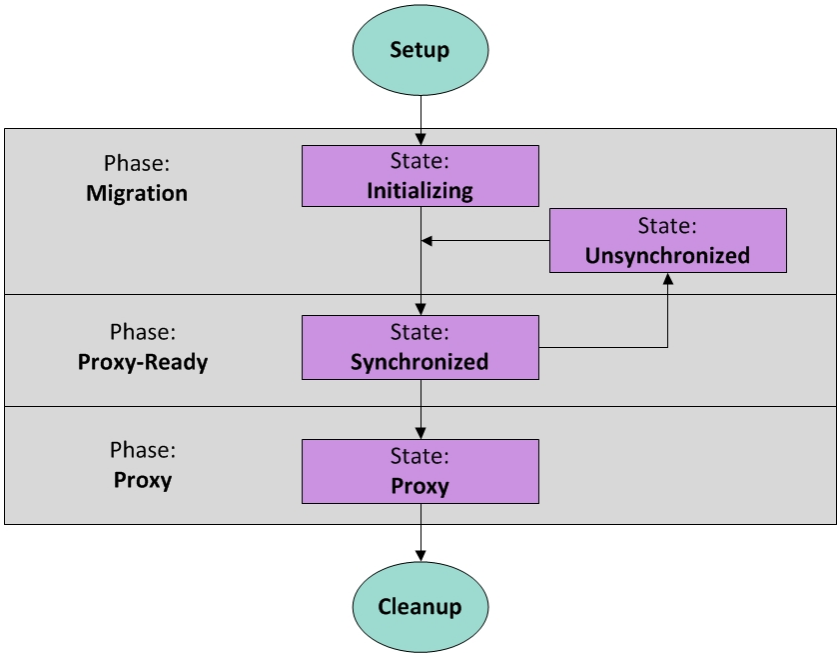


Figure 45. Flow of the IBM Hyper-Scale Mobility

Table 9. The IBM Hyper-Scale Mobility process

Stage	Description	Source and destination volume states
Setup	A volume is automatically created at the destination storage system with the same name as the source volume. The relation between the source and destination volumes is established.	The two volumes are not yet synchronized.
Migration	New data is written to the source and replicated to the destination.	**Initializing** - The content of the source volume is copied to the destination volume. The two volumes are not yet synchronized. This state is similar to the Initializing state of synchronous mirroring (see “Synchronous mirroring statuses” on page 61). As long as the source instance cannot confirm that all of the writes were acknowledged by the destination volume, the state remains Initializing.

Table 9. The IBM Hyper-Scale Mobility process (continued)

Stage	Description	Source and destination volume states
Proxy-Ready	The replication of the source volume data is complete when the destination is synchronized. The source serves host writes as a proxy between the host and the destination. The system administrator issues a command that moves the IBM Hyper-Scale Mobility relation to the proxy. Next, the system administrator maps the host to the destination. In this state, a single copy of the data exists on the destination and any I/O directed to the source is redirected to the destination.	**Synchronized** - The source was wholly copied to the destination. This state is similar to the Synchronized state of synchronous mirroring (see "Synchronous mirroring statuses" on page 61).
Proxy	New data in written to the source and is migrated to the destination. The proxy serves host requests as if it were the target, but it actually impersonates the target.	**Proxy** - The source acts as a proxy to the destination.
Cleanup	After validating that the host has connectivity to the destination volume through the new paths, the storage administrator unmaps the source volume on the source storage system from the host. Then the storage administrator ends the proxy and deletes the relationship.	

Chapter 12. Data-at-rest encryption

The IBM XIV Storage System utilizes full disk encryption for regulation compliance and security audit readiness.

Data-at-rest encryption protects against the potential exposure of XIV system sensitive data on discarded or stolen media. The encryption ensures that the data cannot be read, as long as its encryption key is secured. This feature complements physical security at the customer site, protecting the customer from unauthorized access to the data.

The encryption of the disk drives is transparent to hosts that are attached to the IBM XIV Storage System, and does not affect either their management or performance. The term data-in-flight refers to I/Os that are anywhere between the network interfaces, memory and Infiniband backbone. This type of data is not encrypted.

Common use cases that prompt the protection of data-at-rest are:

- Unauthorized access at Service providers SAN's with consolidated storage (e.g. disk theft)
- Component rotation:
  - Protect data following its physical removal from an IBM XIV Storage System at a customer site
  - Prevent discarded media from being compromised (e.g. failed disk)
- New component add - upgrade (MES) of SSD or module should maintain the encryption capabilities of IBM XIV Storage System.

HIPAA compatibility

IBM XIV Storage System complies with the following security requirements and standards.

The IBM XIV Storage System Data-at-Rest encryption complies with HIPAA Federal requirements as follows:

- User data is inaccessible without XIV system specific keying material.
- Physical separation of encryption keys from encrypted data, by using an external key server
- Cryptographic keys may be replaced at the user's initiative
- All keys stored must be wrapped and stored in ciphertext (not reside in plain text or hidden/obfuscated)
- AES 256 encryption is used to wrap keys and encrypt data, RSA 2048 encryption is used for public key cryptography
- Encryption configuration and settings must be auditable, thus the related information and notifications should be kept in events log

Chapter 13. Management and monitoring

The storage system can be monitored and fully controlled by using different management and automation tools.

The primary management tools for storage administrators are:

- **IBM XIV Management Tools**, which includes **IBM Hyper-Scale Manager** – Management server software connects to and controls one or more storage systems. Remote users can log into the server and use its advanced graphical user interface (GUI) for managing and monitoring multiple storage systems in real time.
- **IBM XCLI Utility** – Provides a terminal-based command-line interface for issuing storage system management, monitoring, and maintenance commands from a client computer upon which the utility is installed.

  The command-line interface is a comprehensive, text-based tool that is used to configure and monitor the system. Commands can be issued to configure, manage, or maintain the system, including commands to connect to hosts and applications.

Programmers can utilize the system's advanced application programming interfaces (APIs) for controlling and automating the system:

- Representational state transfer (REST) APIs
- CIM/SMI-S open APIs
- SNMP

Chapter 14. Event notification destinations

Event notifications can be sent to one or more destinations, meaning to a specific SMS cell number, e-mail address, or SNMP address, or to a destination group comprised of multiple destinations. Each of the following destinations must be defined as described:

SMS destination

An SMS destination is defined by specifying a phone number. When defining a destination, the prefix and phone number should be separated because some SMS gateways require special handling of the prefix.

By default, all SMS gateways can be used. A specific SMS destination can be limited to be sent through only a subset of the SMS gateways.

E-mail destination

An e-mail destination is defined by an e-mail address. By default, all SMTP gateways are used. A specific destination can be limited to be sent through only a subset of the SMTP gateways.

SNMP managers

An SNMP manager destination is specified by the IP address of the SNMP manager that is available to receive SNMP messages.

Destination groups

A destination group is simply a list of destinations to which event notifications can be sent. A destination group can be comprised of SMS cell numbers, e-mail addresses, SNMP addresses, or any combination of the three. A destination group is useful when the same list of notifications is used for multiple rules.

Event information

Events are created by various processes, including the following:

- Object creation or deletion, including volume, snapshot, map, host, and storage pool
- Physical component events
- Network events

Each event contains the following information:

- A system-wide unique numeric identifier
- A code that identifies the type of the event
- Creation timestamp
- Severity
- Related system objects and components, such as volumes, disks, and modules
- Textual description
- Alert flag, where an event is classified as alerting by the event notification rules.

- Cleared flag, where alerting events can be either uncleared or cleared. This is only relevant for alerting events.

Event information can be classified with one of the following severity levels:

Critical
Requires immediate attention

Major Requires attention soon

Minor Requires attention within the normal business working hours

Warning
Nonurgent attention is required to verify that there is no problem

Informational
Normal working procedure event

The IBM XIV Storage System provides the following variety of criteria for displaying a list of events:

- Before timestamp
- After timestamp
- Code
- Severity from a certain value and up
- Alerting events, meaning events that are sent repeatedly according to a snooze timer
- Uncleared alerts

The number of displayed filtered events can be restricted.

Event notification rules

The IBM XIV Storage System monitors the health, configuration changes, and activity of your storage systems and sends notifications of system events as they occur. Event notifications are sent according to the following rules:

Which events
The severity, event code, or both, of the events for which notification is sent.

Where The destinations or destination groups to which notification is sent, such as cellular phone numbers (SMS), e-mail addresses, and SNMP addresses.

Notifications are sent according to the following rules:

Destination
The destinations or destination groups to which a notification of an event is sent.

Filter A filter that specifies which events will trigger the sending of an event notification. Notification can be filtered by event code, minimum severity (from a certain severity and up), or both.

Alerting
To ensure that an event was indeed received, an event notification can be sent repeatedly until it is cleared by an XCLI command or the IBM XIV Storage Management GUI. Such events are called alerting events. Alerting events are events for which a snooze time period is defined in minutes. This means that an alerting event is resent repeatedly each snooze time

interval until it is cleared. An alerting event is uncleared when it is first triggered, and can be cleared by the user. The cleared state does not imply that the problem has been solved. It only implies that the event has been noted by the relevant person who takes the responsibility for fixing the problem. There are two schemes for repeating the notifications until the event is clear: snooze and escalation.

Snooze
Events that match this rule send repeated notifications to the same destinations at intervals specified by the snooze timer until they are cleared.

Escalation
You can define an escalation rule and escalation timer, so that if events are not cleared by the time that the timer expires, notifications are sent to the predetermined destination. This enables the automatic sending of notifications to a wider distribution list if the event has not been cleared.

Event information

Events are created by various processes, including the following:

- Object creation or deletion, including volume, snapshot, map, host, and storage pool
- Physical component events
- Network events

Each event contains the following information:

- A system-wide unique numeric identifier
- A code that identifies the type of the event
- Creation timestamp
- Severity
- Related system objects and components, such as volumes, disks, and modules
- Textual description
- Alert flag, where an event is classified as alerting by the event notification rules.
- Cleared flag, where alerting events can be either uncleared or cleared. This is only relevant for alerting events.

Event information can be classified with one of the following severity levels:

Critical
Requires immediate attention

Major Requires attention soon

Minor Requires attention within the normal business working hours

Warning
Nonurgent attention is required to verify that there is no problem

Informational
Normal working procedure event

The IBM XIV Storage System provides the following variety of criteria for displaying a list of events:

- Before timestamp
- After timestamp

- Code
- Severity from a certain value and up
- Alerting events, meaning events that are sent repeatedly according to a snooze timer
- Uncleared alerts

The number of displayed filtered events can be restricted.

Event notification gateways

Event notifications can be sent by SMS, e-mail, or SNMP manager. This step defines the gateways that will be used to send e-mail or SMS.

E-mail (SMTP) gateways

Several e-mail gateways can be defined to enable notification of events by e-mail. By default, the IBM XIV Storage System attempts to send each e-mail notification through the first available gateway according to the order that you specify. Subsequent gateways are only attempted if the first attempted gateway returns an error. A specific e-mail destination can also be defined to use only specific gateways.

All event notifications sent by e-mail specify a sender whose address can be configured. This sender address must be a valid address for the following two reasons:

- Many SMTP gateways require a valid sender address or they will not forward the e-mail.
- The sender address is used as the destination for error messages generated by the SMTP gateways, such as an incorrect e-mail address or full e-mail mailbox.

E-mail-to-SMS gateways

SMS messages can be sent to cell phones through one of a list of e-mail-to-SMS gateways. One or more gateways can be defined for each SMS destination.

Each such e-mail-to-SMS gateway can have its own SMTP server, use the global SMTP server list, or both.

When an event notification is sent, one of the SMS gateways is used according to the defined order. The first gateway is used, and subsequent gateways are only tried if the first attempted gateway returns an error.

Each SMS gateway has its own definitions of how to encode the SMS message in the e-mail message.

Chapter 15. User roles and permissions

User roles allow specifying which roles are applied and the various applicable limits.

Note: None of these system-defined users have access to data.

Table 10. Available user roles

User role	Permissions and limits	Typical usage
Read only	Read only users can only list and view system information.	The system operator, typically, but not exclusively, is responsible for monitoring system status and reporting and logging all messages.
Application administrator	Only application administrators carry out the following tasks: • Creating snapshots of assigned volumes • Mapping their own snapshot to an assigned host • Deleting their own snapshot	Application administrators typically manage applications that run on a particular server. Application managers can be defined as limited to specific volumes on the server. Typical application administrator functions: • Managing backup environments: – Creating a snapshot for backups – Mapping a snapshot to back up server – Deleting a snapshot after backup is complete – Updating a snapshot for new content within a volume • Managing software testing environment: – Creating an application instance – Testing the new application instance
Storage administrator	The storage administrator has permission to all functions, except: • Maintenance of physical components or changing the status of physical components • Only the predefined administrator, named *admin*, can change the passwords of other users	Storage administrators are responsible for all administration functions.

Table 10. Available user roles (continued)

User role	Permissions and limits	Typical usage
Technician	The technician is limited to the following tasks: • Physical system maintenance • Phasing components in or out of service	Technicians maintain the physical components of the system. Only one predefined technician is specified per system. Technicians are IBM XIV Storage System technical support team members.

Notes:

1. All users can view the status of physical components; however, only technicians can modify the status of components.
2. User names are case-sensitive.
3. Passwords are case-sensitive.

User groups

A user group is a group of application administrators who share the same set of snapshot creation permissions. This enables a simple update of the permissions of all the users in the user group by a single command. The permissions are enforced by associating the user groups with hosts or clusters. User groups have the following characteristics:

- Only users who are defined as application administrators can be assigned to a group.
- A user can belong to only a single user group.
- A user group can contain up to eight users.
- If a user group is defined with access_all="yes", application administrators who are members of that group can manage all volumes on the system.

Storage administrators create the user groups and control the various permissions of the application administrators.

Predefined users

There are several predefined users configured on the IBM XIV Storage System. These users cannot be deleted.

Storage administrator
This user id provides the highest level of customer access to the system.

Predefined user name: admin

Default password: adminadmin

Technician
This user id is used only by IBM XIV Storage System service personnel.

Predefined user name: technician

Default password: Password is predefined and is used only by the IBM XIV Storage System technicians.

Note: Predefined users are always authenticated by the IBM XIV Storage System, even if LDAP authentication has been activated for them.

User information

Configuring users requires defining the following options:

Role Specifies the role category that each user has when operating the system. The role category is mandatory. for explanations of each role.

Name Specifies the name of each user allowed to access the system.

Password
All user-definable passwords are case sensitive.
Passwords are mandatory, can be 6 to 12 characters long, use uppercase or lowercase letters as well as the following characters: ~!@#$%^&*()_+-={}|:;<>?,./\[] .

E-mail E-mail is used to notify specific users about events through e-mail messages. E-mail addresses must follow standard addressing procedures. E-mail is optional. Range: Any legal e-mail address.

Phone and area code
Phone numbers are used to send SMS messages to notify specific users about events. Phone numbers and area codes can be a maximum of 63 digits, hyphens (-) and periods (.) Range: Any legal telephone number; The default is N/A

Chapter 16. User authentication and access control

IBM XIV Storage System features role-based authentication either natively or by using LDAP-based authentication.

The system provides:

Role-based access control

Built-in roles for access flexibility and a high level of security according to predefined roles and associated tasks.

Two methods of access authentication

The following methods of user authentication are supported:

Native authentication

This is the default mode for authentication of users and groups that are defined on the storage system. In this mode, users and groups are authenticated against a database on the system.

LDAP

When enabled, the system authenticates the users against an LDAP repository.

Note: The administrator and technician roles are always authenticated by the IBM XIV Storage System, regardless of the authentication mode.

Native authentication

Native authentication if the default mode for authenticating users and user groups.

In this mode, users and groups are authenticated against a database on the system, based on the submitted username and password, which are compared to user credentials defined and stored on the storage system.

The authenticated user must be associated with a user role that specifies the system access rights.

LDAP authentication

Lightweight Directory Access Protocol (LDAP) support enables the IBM XIV Storage System to authenticate users through an LDAP repository.

When LDAP authentication is enabled, the username and password of a user accessing the IBM XIV Storage System (through CLI or GUI) are used by the IBM XIV system to login into a specified LDAP directory. Upon a successful login, the IBM XIV Storage System retrieves the user's IBM XIV group membership data stored in the LDAP directory, and uses that information to associate the user with an IBM XIV administrative role.

The IBM XIV group membership data is stored in a customer defined, pre-configured attribute on the LDAP directory. This attribute contains string values which are associated with IBM XIV administrative roles. These values might be LDAP Group Names, but this is not required by the IBM XIV Storage System.

The values the attribute contains, and their association with IBM XIV administrative roles, are also defined by the customer.

Supported domains

The IBM XIV Storage System supports LDAP authentication of the following directories:

- Microsoft Active Directory
- SUN directory
- Open LDAP

LDAP multiple-domain implementation

In order to support multiple LDAP servers that span over different domains, and in order to use the **memberOf** property, the IBM XIV Storage System allows for more than one role for the Storage Administrator and the Read⌂Only roles.

The predefined XIV administrative IDs “admin” and “technician” are always authenticated by the IBM XIV storage system, whether or not LDAP authentication is enabled.

Responsibilities division between the LDAP directory and the storage system

Following are responsibilities and data maintained by the IBM XIV system and the LDAP directory:

LDAP directory

- Responsibilities - user authentication for IBM XIV users, and assignment of IBM XIV related group in LDAP.
- Maintains - Users, username, password, designated IBM XIV related LDAP groups associated with IBM XIV Storage System.

IBM XIV Storage System

- Responsibilities - Determination of appropriate user role by mapping LDAP group to an IBM XIV role, and enforcement of IBM XIV user system access.
- Maintains - Mapping of LDAP group to IBM XIV role.

LDAP authentication logic

The LDAP authentication process consists of several key steps.

1. The LDAP server and system parameters must be defined.
2. A storage system user must be defined on that LDAP server. The storage system uses this user when searching for authenticated users. This user is later on referred to as system's configured service account.
3. The LDAP user requires an attribute in which the values of the storage system user roles are stored.
4. Mapping between LDAP user attributes and storage system user roles must be defined.
5. LDAP authentication must be enabled on the storage system.

Once LDAP is configured and enabled, the predefined user is granted with login credentials authenticated by the LDAP server, rather than the storage system itself.

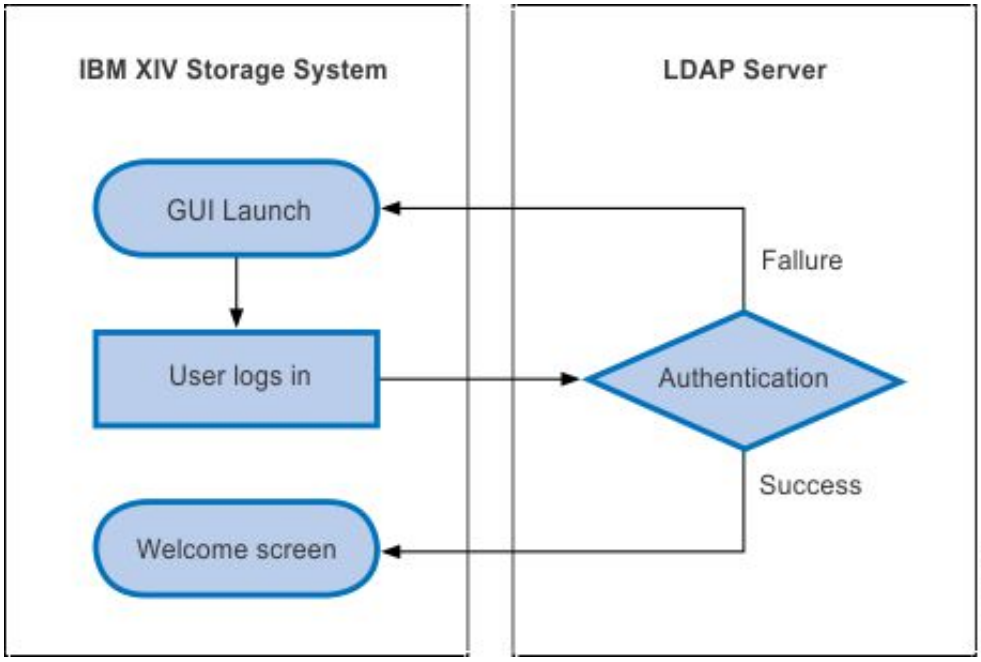


Figure 46. Login to a specified LDAP directory

User validation

During the login, the system validates the user as follows:

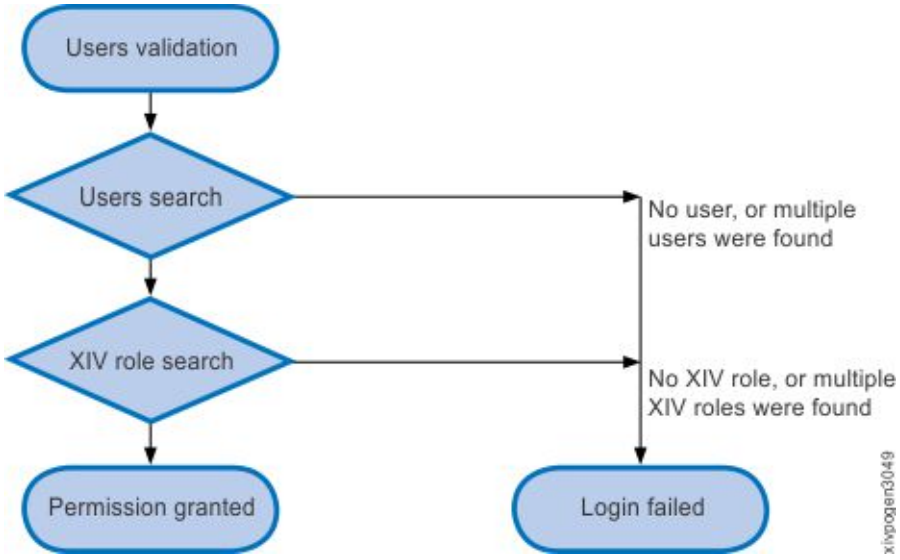


Figure 47. The way the system validates users through issuing LDAP searches

Issuing a user search

The system issues an LDAP search for the user's entered username. The request is submitted on behalf of the system's configured service account and the search is conducted for the LDAP server, base DN and reference attribute as specified in the storage system LDAP configuration.

The base DN specified in the storage system LDAP configuration serves as a reference starting point for the search – instructing LDAP to locate the value submitted (the username) in the attribute specified.

If a single user is found - issuing a storage system role search
The system issues a second search request, this time submitted on behalf of the user (with the user's credentials), and will search for storage system roles associated with the user, based on the storage system LDAP configuration settings.

If a single storage system role is found - permission is granted
The system inspects the rights associated with that role and grant login to the user. The user's permissions are in correspondence with the role associated by the storage system, base on the storage system LDAP configuration.

If no storage system role is found for the user, or more than one role was found
If the response by LDAP indicates that the user is either not associated with a storage system role (no user role name is found in the referenced LDAP attribute for the user), or is actually associated with more than a single role (multiple roles names are found) – login will fail and a corresponding message will be returned to the user.

If no such user was found, or more than one user were found
If LDAP returns no records (indicating no user with the username was found) or more than a single record (indicating that the username submitted is not unique), the login request fails and a corresponding message is returned to the user.

Chapter 17. Multi-Tenancy

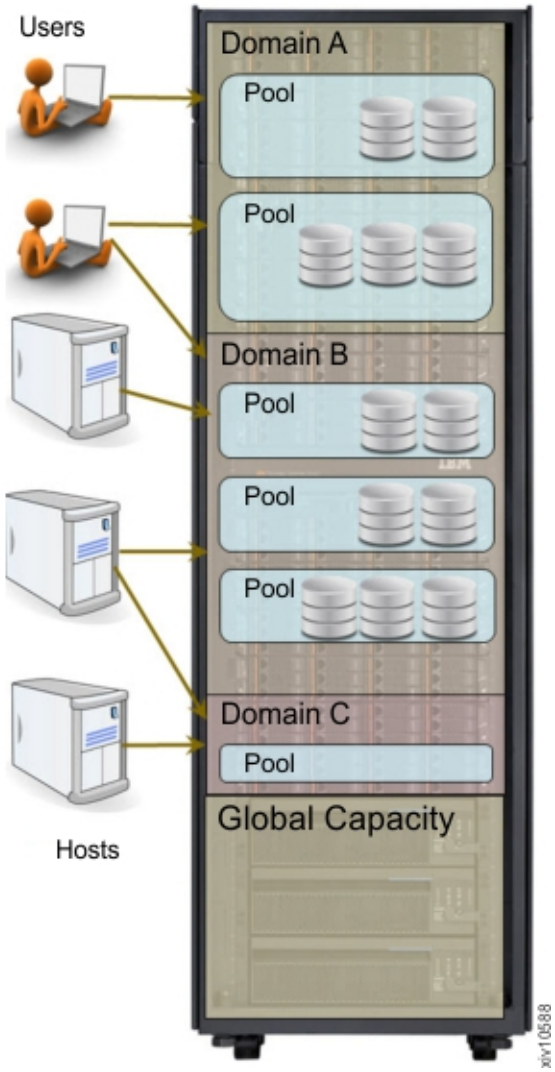
The storage system allows allocating storage resources to several independent administrators, assuring that one administrator cannot access resources associated with another administrator.

Multi-tenancy extends the storage system approach to role-based access control. In addition to associating the user with predefined sets of operations and scope (the applications on which an operation is allowed), the storage system enables the user to freely determine what operations are allowed, and where they are allowed.

The main idea of multi-tenancy is to allow the allocation of storage resources to several independent administrators with the assurance that one administrator cannot access resources associated with another administrator.

This resource allocation is best described as a partitioning of the system's resources to separate administrative *domains*. A domain is a subset, or partition, of the system's resources. It is a named object to which users, pools, hosts/clusters, targets, etc. may be associated. The domain restricts the resources a user can manage to those associated with the domain.

A domain maintains the user relationships that exist at the storage system level, as shown in the following figure.



A *domain administrator* is a user who is associated with a domain. The domain administrator is restricted to performing operations on objects associated with a specific domain.

The following access rights and restrictions apply to domain administrators:

- A user is created and assigned a role (for example: storage administrator, application administrator, read-only).
- When assigned to a domain, the user retains his given role, limited to the scope of the domain.
- Access to objects in a domain is restricted up to the point where the defined user role intersects the specified domain access.
- By default, domain administrators cannot access objects that are not associated with their domains.

Multi-tenancy offers the following benefits:

Partitioning

The system resources are partitioned to separate domains. The domains are assigned to different tenants and each tenant administrator gets

permissions for a specific, or several domains, to perform operations only within the boundaries of the associated domain(s).

Self-sufficiency
The domain administrator has a full set of permissions needed for managing all of the domain resources.

Isolation
There is no visibility between tenants. The domain administrator is not informed of resources outside the domain. These resources are not displayed on lists, nor are their relevant events or alerts displayed.

User-domain association
A user can have a domain administrator role on more than one domain.

Users other than the domain administrator
Storage, security, and application administrators, as well as read-only users, retain their right to perform the same operations that they have in a non-domain-based environment. They can access the same objects under the same restrictions.

Global administrator
The global administrator is not associated with any specific domain, and determines the operations that can be performed by the domain administrator in a domain.

This is the only user that can create, edit, and delete domains, and associate resources to a domain.

An *open* or *closed* policy can be defined so that a global administrator may, or may not, be able to see *inside* a domain.

Intervention of a global domain administrator, that has permissions for the global resources of the system, is only needed for:

- Initial creation of the domain and assigning a domain administrator
- Resolving hardware issues

User that is not associated with any domain
A user that is not associated with any domain has access rights to all of the entities that are not uniquely associated with a domain.

Working with multi-tenancy

This section provides a general description about working with multi-tenancy and its attributes.

The domain administrator
The domain administrator has the following attributes:

- Prior to its association with a domain, the future domain administrator (now a system administrator) has access to all non-domain entities, and no access to domain-specific entities.
- When the storage administrator becomes a domain administrator all access rights to non-domain entities are lost.
- The domain administrator can map volumes to hosts as long as both the volume and the host belong to the domain.
- The domain administrator can copy and move volumes across pools as long as the pools belong to domains administered by the domain administrator.

- Domain administrators can manage snapshots for all volumes in their domains.
- Domain administrators can manage consistency and snapshot groups for all pools in their domains. Moving consistency groups across pools is allowed as long as both source and destination pools are in the admin's domains.
- Domain administrators can create and manage pools under the storage constraint associated with their domain.
- Although not configurable by the domain administrator, hardware list, and events are available for view-only to the domain administrator within the scope of the domain.
- Commands that operate on objects not associated with a domain are not accessible by the domain administrator.

Domain

The domain has the following attributes:

- *Capacity* - the domain is allocated with a capacity that is further allocated among its pools. The domain provides an additional container in the hierarchy of what was once *system-pool-volume*, and is now *system-domain-pool-volume*:
  - The unallocated capacity of the domain is reserved to the domain's pools
  - The sum of the hard capacity of the system's domains cannot exceed the XIV system's total hard capacity
  - The sum of the soft capacity of the system's domains cannot exceed the XIV system's total soft capacity
- *Maximum number of volumes per domain* - the maximum number of volumes per system is divided among the domains in a way that one domain cannot consume all of the system resources at the expense of the other domains.
- *Maximum number of pools per domain* - the maximum number of pools per system is divided among the domains in a way that one domain cannot consume all of the system resources at the expense of the other domains.
- *Maximum number of mirrors per domain* - the maximum number of mirrors per system is divided among the domains.
- *Maximum number of consistency groups per domain* - the maximum number of consistency groups per system is divided among the domains.
- *Performance class* - the maximum aggregated bandwidth and IOPS is calculated for all volumes of the domain, rather than on a system level.
- The domain has a string that identifies it for LDAP authentication.

Mirroring in a multi-tenancy environment

- The target, target connectivity and interval schedule are defined, edited and deleted by the storage administrator.
- The domain administrator can create, activate and change properties to a mirroring relation based on the previously defined target and target connectivity that are associated with the domain.
- The remote target does not have to belong to a domain.
- Whenever the remote target belongs to a domain, it checks that the remote target, pool and volume (if specified upon the mirror creation) all belong to the same domain.

Chapter 18. Integration with ISV environments

The storage system can be fully integrated with different independent software vendor (ISV) platforms, APIs, and cloud environments, such as Microsoft Hyper-V, VMware vSphere, OpenStack, and more.

This integration can be implemented natively or by using IBM cloud software solutions, which can facilitate and enhance this integration.

For more information about the available cloud storage solutions, see the '**Platform and application integration**' section on IBM Knowledge Center.

VMware Virtual Volumes

XIV is now ready for *VMware Virtual Volumes (VVols)*. VMware Virtual Volumes (VVols) is a feature of VMware vSphere, based on a new storage architecture, that associates a single VM with multiple LUNs.

With VVols, the VMware vCenter (Web Client) administrator can offload VM-granular snapshots and cloning to IBM Storage, automate IBM storage provisioning by workload-aware policy, and apply VM-granular backup and *in-place* restore based on IBM Storage snapshots. IBM Storage administrators can define and publish workload-specific storage services to vCenter, and scale down management efforts, enjoying fully automatic volume life cycle management. Lastly, VVols is "elastic", meaning that Storage administrators do not need to pre-allocate large capacity for datastores. Instead, storage is instantly and automatically allocated (and reclaimed) on demand, at exactly the right amount.

VMware VVols automation is based on *VMware vSphere APIs for Storage Awareness (VASA)*. The *IBM Storage Provider for VMware VASA* is a feature of the *IBM Storage Integration Server*, and will support the orchestration of all VVols operations with XIV. For more information on the IBM Storage Provider for VMware VASA and the IBM Storage Integration Server, refer to the *IBM Storage Provider for VMware VASA* (http://www-01.ibm.com/support/knowledgecenter/STJTAG/hsg/hsg_vasa_kcwelcome.html) and *IBM Storage Integration Server* (http://www-01.ibm.com/support/knowledgecenter/STJTAG/hsg/hsg_isis_kcwelcome.html) documentation.

For a preview of *VMware Virtual Volumes (VVols)*, see http://blogs.vmware.com/vsphere/2012/10/virtual-volumes-vvols-tech-preview-with-video.html.

Prerequisites for working with VVols

Upon availability, a VVols deployment will require VVols-capable storage arrays and a VASA Provider.

- Make sure the following software and server versions are installed:
  - XIV version 11.5.1, and later
  - IBM Storage Integration Server version 2.0, and up (VASA 2.0-compliant)
  - VMware vCenter and VMware ESX servers
  - VMware vSphere Client
- Deployment of an IBM Storage provider for VASA (incorporated in the IBM Storage Integration Server)

- Define a Storage Integration Administrator (SIA) user role.

Integration with Microsoft Azure Site Recovery

Microsoft Azure Site Recovery (ASR) solution helps you protect important applications by coordinating the replication and recovery of private clouds across sites.

IBM XIV Storage System v11.6.0 supports Microsoft Azure Site Recovery, enabling customers using Microsoft System Center Virtual Machine Manager (SCVMM) to seamlessly orchestrate and manage XIV replication and disaster recovery.Support for Microsoft Azure Site Recovery is based on XIV support for SMI-S v1.6 (http://www.snia.org/ctp/conformingproviders/ibm.html#sftw4).

The SCVMM ASR integrates with storage solutions, such as IBM XIV CIM Agent, to provide site-to-site disaster recovery for Hyper-V environments by leveraging the SAN replication capabilities that are natively offered by IBM XIV storage systems. It orchestrates replication and failover for virtual machines managed by SCVMM.

SCVMM ASR uses the IBM XIV Remote Mirroring feature through SMI-S to create and manage the replication groups. IBM XIV Remote Mirroring is a host-independent, array-based data mirroring solution that enables affordable data distribution and disaster recovery for applications. With this feature, the users can copy virtual volumes from one IBM XIV storage system to another.

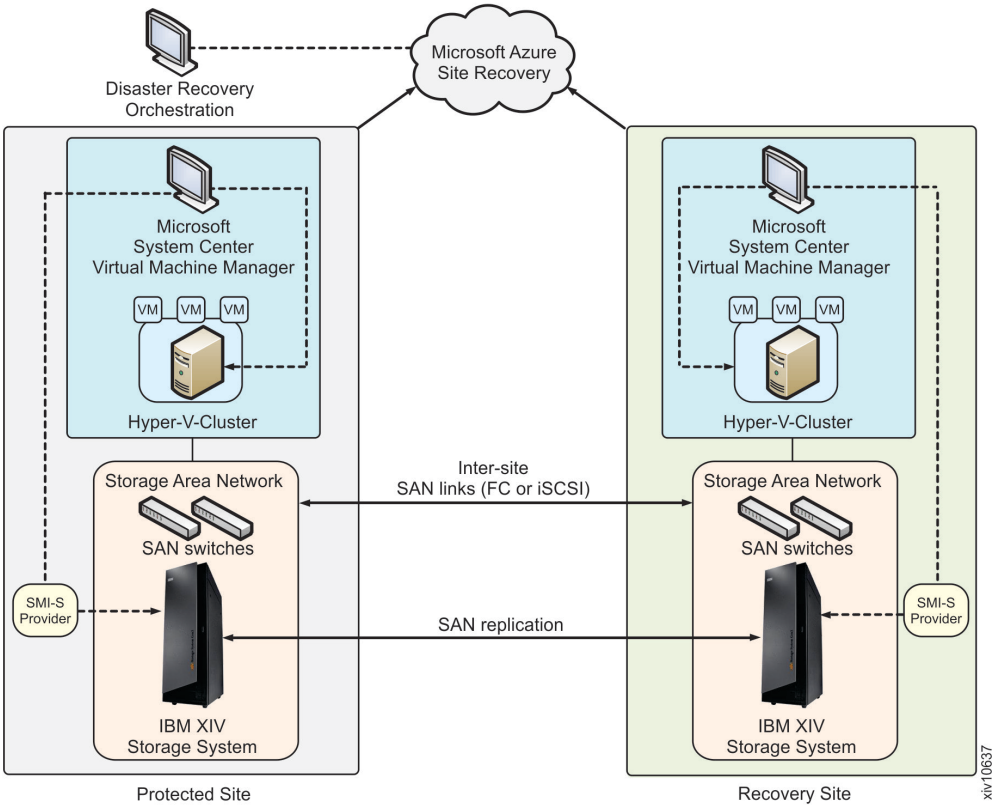


Figure 48. Overview of Microsoft Azure Site Recovery support

Chapter 19. Software upgrade

Non disruptive code load (hot upgrade) enables the IBM XIV Storage System to upgrade its software from a current version to a newer version without disrupting application service.

The upgrade process is run on all modules in parallel and is designed to be quick enough so that the applications' service on the hosts will not be damaged. The upgrade requires that neither data migration nor rebuild processes are run, and that all internal network paths are active.

During the non disruptive code load process there is a point in time dubbed the 'upgrade-point-of-no-return', before which the process can still be aborted (either automatically by the system - or manually through a dedicated CLI). Once that point is crossed - the upgrade process is not reversible.

Following are notable characteristics of the Non-disruptive code load:

Duration of the upgrade process

The overall process of downloading new code to storage system and moving to the new code is done online to the application/host.

The duration of the upgrade process is affected by the following factors:

- The upgrade process requires that you reduce all I/Os. If there are a lot of I/Os in the system, or there are slow disks, the system might not be able to stop the I/Os fast enough, so it will restart them and try again after a short while, taking into consideration some retries.
- The upgrade process installs a valid version of the software and then retains its local configuration. This process might take a considerable amount of time, depending on the future changes in the structure of the configuration.

Prerequisites and constraints

- The process cannot run if a data migration process or a rebuild process is active. An attempt to start the upgrade process when either a data migration or a rebuild process is active will fail.
- Generally, everything that happens after the point-of-no-return is treated as if it happened after the upgrade is over.
- As long as the overall hot upgrade is in progress (up to several minutes) no management operations are allowed (save for status querying), and no events are processed.
- Prior to the point-of-no-return, a manual abort of the upgrade is available.

Effect on mirroring

Mirrors are automatically deactivated before the upgrade, and reactivated after it is over.

Effect on management operations

During the upgrade process it is possible to query the system about the upgrade status, and the process can also be aborted manually before the 'point-of-no-return'. If a failure occurs before this point - the process will be aborted automatically.

Handling module or disk failure during the upgrade
If the failure occurs before the point-of-no-return, it will abort the upgrade. If it happens after that point, the failure is treated as if it happened after the upgrade is over.

Handling power failure during the upgrade
As for power failure before the point-of-no-return - power is being monitored during the time the system prepares for the upgrade (before the point-of-no-return). If a power failure is detected, the upgrade will be aborted and the power failure will be taken care of by the old version.

Preparing for upgrade

The IBM XIV Storage System upgrades the system code without disconnecting active hosts or stopping I/O operations.

Important: The upgrade must be performed only by an authorized IBM service technician.

Preparing for the upgrade

Before the code load (upgrade), fulfill the following prerequisites by verifying that:

1. The multipathing feature (provided by the operating system) is working on the host.
2. There are paths from the host to at least two different interface modules on IBM XIV.
3. There is no more than a single initiator in each zone (SAN Volume Controller attached to IBM XIV is an exception).
4. The host was attached to IBM XIV using the xiv_attach utility.
   - This is mandatory.
   - This applies to both installable HAK and portable HAK.
   - Exceptions to this prerequisite are supported platforms, for which no HAK is available (for example, VMware or Linux on Power systems).
5. The minimal version of the "IBM XIV Host Attachment Kit for Windows" is 1.5.3. This version prevents Windows hosts from a potential loss of access.
6. In a case of IBM XIV uses FC connectivity for remote mirroring, the two systems should be connected to a SAN switch. Direct connection is not supported and is known to be problematic.
7. Hosts should be attached to the FC ports through an FC switch, and to the iSCSI ports through a Gigabit Ethernet switch. Direct attachment between hosts and to the IBM XIV Storage System is not supported.

Be aware of the following:

1. Co-existence with other multipathing software is not supported as GA (RPQ approval is required).
2. Connectivity to other storage servers from the same host is not supported as GA (RPQ approval is required).
3. The mirroring is automatically suspended and resumed for a short while during the upgrade.
4. Mirroring from 10.2.4.x to 10.2.1.x or older versions is not supported.
5. There are special considerations where MS Geo Cluster is involved. Contact IBM support for more details.

Recommended practices:

1. It is highly recommended to have the latest XIV Host Attachment Kit installed on the host. Each release of a HAK fixes known issues on older versions.
   - Being on an older level means being exposed to problems already found and fixed.
2. It is recommended to follow the zoning best practices of IBM XIV as described in the Redbook and in the XIV Host Attachment Guide.
3. Best practice is to follow the OS Provider recommendations regarding service packs and storage-related Hot Fixes. These fixes are released from time to time by the OS provider and are outside of IBM XIV control. Some fixes are listed on the release notes of the latest available HAK and have to be applied before the upgrade.
4. It is recommended to keep your host system with up-to-date BIOS and HBA drivers.
5. It is recommended to perform the upgrade on time during which the workload is relatively low.

Availability of TA support:

1. If entitled with a TA service for your IBM XIV Storage System, contact your assigned Technical Advisor when planning for a code upgrade.

Chapter 20. Remote support and proactive support

To allow IBM to provide support for the storage system, the proactive support and remote support options are available.

Note: For various preventive and diagnostics support actions relating to the storage system's continuous operation, IBM Support requires customer approval. Without customer approval, these support actions cannot be preformed.

- **Proactive support** ("Call Home") – Allows proactive notifications regarding the storage system health and components to be sent to IBM Support at predefined intervals. Heartbeats and events are sent from the system to the IBM service center. The service center analyzes the information within the heartbeats and the events, correlates it with its vast database and can then trigger a component replacement prior to its potential failure.

  Upon detection of any hardware or software error code, both IBM Support and your predefined contact person are notified via e-mail, through a specified SMTP gateway. If IBM Support determines that the detected event requires service or further investigation, a new PMR is created and sent to appropriate IBM Support team. Proactive support minimizes the number of interaction cycles with IBM Support.
- **Remote support** – Allows IBM Support to remotely and securely access your storage system when needed during a support call. This option requires IP communication between the storage system and the IBM Remote Support Center. If a storage system does not have direct access to the Internet (for example, due to a firewall), use the XIV Remote Support Proxy utility to enable the connection. Remote support minimizes the time it takes to diagnose and remedy storage system operational issues.

Note: No data can be accessed by IBM Support when the remote support option is used.

Notices

These legal notices pertain to the information in this IBM Storage product documentation.

This information was developed for products and services offered in the US. This material may be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
USA

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

The performance data discussed herein is presented as derived under specific operating conditions. Actual results may vary.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at Copyright and trademark information website (www.ibm.com/legal/copytrade.shtml).

Microsoft is a trademark of Microsoft Corporation in the United States, other countries, or both.

Printed in USA

GC27-3912-10