FUJITSU

Storage Manager (StorMan) V6.0

Provisioning and managing virtualized storage resources

Edition April 2015

Comments… Suggestions… Corrections…

The User Documentation Department would like to know your opinion on this manual. Your feedback helps us to optimize our documentation to suit your individual needs.

Feel free to send us your comments by e-mail to:
manuals@ts.fujitsu.com

Certified documentation
according to DIN EN ISO 9001:2008

To ensure a consistently high quality standard and user-friendliness, this documentation was created to meet the regulations of a quality management system which complies with the requirements of the standard DIN EN ISO 9001:2008.

cognitas. Gesellschaft für Technik-Dokumentation mbH
www.cognitas.de

Copyright and Trademarks

This manual is printed on paper treated with chlorine-free bleach.

Contents

1 Introduction

1.1 Functional overview

The main scope of Storage Manager (further on referenced as StorMan) is the virtualization and dynamic management of storage resources and their dynamic assignment to servers in the following cases:

- Storage provisioning tasks

  Example: An application requires additional storage capacity according to a specific policy that should be assigned quickly and automatically.

- Repurposing scenarios

  Example: An application is migrated to a new server requiring access to the same storage resources as before.

- Dynamic Infrastructure scenarios

  Example: An application expands dynamically to an additional server requiring access to the same storage resources as before.

- Disaster recovery scenarios

  Example: An application is subject to dynamic failover after failure of a server requiring access to the same storage resources.

- Storage consolidation scenarios

  Example: The storage resources of an application are consolidated.

- Storage information and monitoring scenarios

  Example: With the scope of uniform integration of storage arrays into server management software (e.g. ServerView Operations Manager).

A uniform management service for automatic storage provisioning, information and monitoring is provided by StorMan.

StorMan provides a common virtualization layer offering:

- Common and stable interfaces for
  - Integration in BS2000 (SHC-OSD)
  - standalone usage
- Integration with server management software such as ServerView Operations Manager, openSM2, etcManagement options for
  - SE Manager on BS2000 SE Servers
  - SQ Manager on BS2000 SQ Servers
  - SQ HA (High Availability) concepts with HAPST
  - Automation with prescheduled or event-triggered scripts based on StorMan's CLIs

- Encapsulation of the backend complexity of

  – Storage-vendor-specific interfaces (Fujitsu, EMC, NetApp) and

  – Storage connectivity types on the backend

  The following figure provides an overview of the architectural layers:
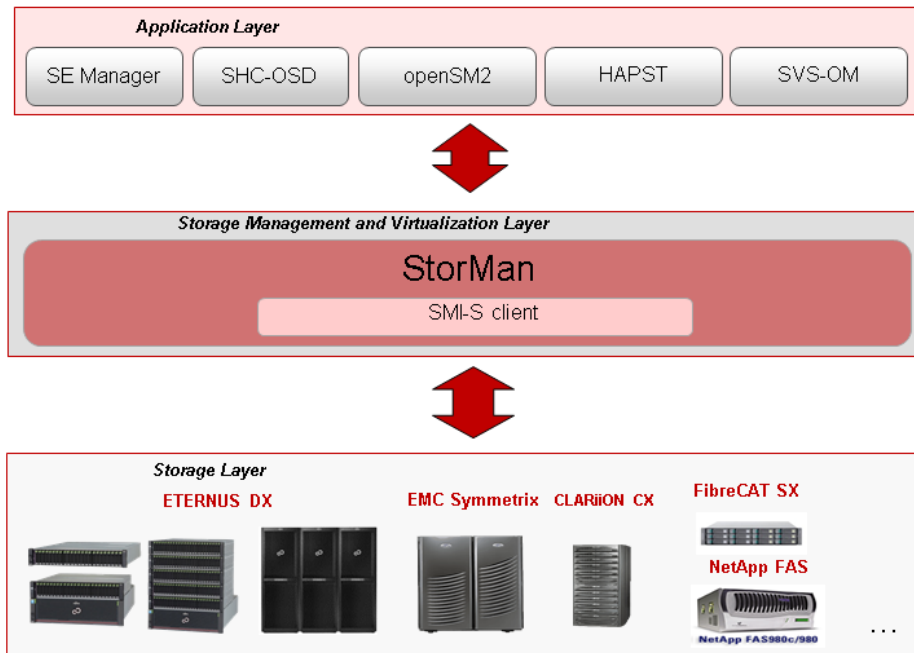
Figure 1: Overview

The base technology of StorMan is the SMI-S (Storage Management Initiative Specification), standard defined and supported by the SNIA (Storage Networking Industry Association). SMI-S is the standard for storage management in the storage world. However, additional to SMI-S vendor-specific APIs are used for integration.

The functionality of StorMan covers the main functions needed for storage provisioning, namely:

- Managing and administrating storage resources (storage volumes and pools)
- LUN creation and deletion
- LUN masking (managing the hosts access to volumes)
- Replication service (local-mirroring replication based on snap and clone technology) and remote replication

Additionally, StorMan provides the functionality required for information and for monitoring storage arrays, namely:

- Discovery
- Information about logical and physical components
- Status monitoring
- Statistical and performance metrics

1.2 Documentation

With exception of the WebUI functions you will find a complete description of StorMan (e.g. concept, architecture, installation, CLI functions) in the StorMan manual.

For users of the WebUI the manual only contains a chapter describing the WebUI startup and basic information for using the WebUI.

The complete reference of WebUI functions you will find online in the StorMan help system which you can call context sensitive from the StorMan WebUI (refer to section "Using Help").

Additional product information

Current information, version and hardware dependencies and instructions for installing and using a product version are contained in the associated Release Notice. These Release Notices are available at *http://manuals.ts.fujitsu.com*.

See the Release Notices for special information on a new release.

1.3 Documentation guide for different scenarios

For general information about StorMan refer to chapter "Functions and Architecture".

For installation of StorMan refer to chapter "Installation". In addition to system and software requirements use the installation section concerning your system platform.

Which other chapters are relevant for the user depends on the usage scenario.

1.3.1 "Standalone usage"

Description of the StorMan tasks:

- For configuration and administration of storage resources to be managed by StorMan refer to chapter "Setting up StorMan configurations". The concerning functions are available on the CLI. Alternatively offered by the WebUI.

- To provide host with resources managed by StorMan refer to chapter "Administration and Provisioning tasks". The concerning function are available on the CLI. Alternatively offered by the WebUI.

- To replicate storage devices refer to chapter "Replication tasks". The concerning functions are available on the CLI. Alternatively offered by the WebUI.

- To get statistic and performance information from storage systems and their storage devices refer to chapter "Statistics and performance tasks". The concerning functions are available on the CLI.

Description of the StorMan interfaces:

- All CLI commands can be used. For the syntax description refer to chapter "StorMan for programmers – CLI description".

- Alternatively to the use of CLI commands the StorMan WebUI is provided. For basic information about startup and using the WebUI refer to chapter "The graphical user interface – StorMan WebUI". Please notice that the complete WebUI function reference is only contained in the online help system.

Description of special issues depending on the used storage system:

- Depending on the used storage system some functions to configure storage resources may not be available with StorMan (e.g. create a logical device). Refer to chapter "Special properties of storage systems".

1.3.2 Integration with server management software

For integration with server management software such as ServerView Operations Manager and performance monitoring software such as openSM2 only a subset of StorMan functions is available. This subset is delivered with the name of **StorManMonitor** and consists of the functions for information and monitoring.

StorManMonitor does not include active management functions and does not include the WebUI. As the interfaces of StorManMonitor functions are identical to the StorMan interfaces they are described as StorMan interfaces further on.

Description of the StorMan tasks:

* For configuration and discovery of storage resources to be managed by StorMan refer to chapter "Setting up StorMan configurations".

Description of the StorMan interfaces:

* The following subset of CLI commands is available:

  – The CLI commands `storcfg host` and `storcfg cimom` can be used for configuration.

  – The CLI command `storcfg system -show` can be used for gathering detailed configuration and status information of the storage systems.

  – The CLI command `storstat` that provides statistical information and metrics can be used for monitoring the performance of the storage systems.

  – Additionally the CLI commands `stormandb`, `storcheck`, `storparam`, `storinfo` und `storemc` can be used.

  For the syntax description refer to chapter "StorMan for programmers – CLI description".

1.3.3 Integration in BS2000

For integration in BS2000 with the BS2000 product SHC-OSD the complete integration of StorMan is done by SHC-OSD internally. Refer to "SHC-OSD" manual [1] for further details.

To manage ETERNUS DX storage systems by SHC-OSD from BS2000, StorMan is running on the Management Unit (M2000) of an SE Server or remote on a Windows or Linux server. The configuration of storage resources has to be set up by StorMan on the specific server. Please refer to chapter "Setting up StorMan configurations".

Description of the StorMan tasks:

* For configuration of storage resources to be managed by StorMan refer to chapter "Setting up StorMan configurations".

* To provide host with resources managed by StorMan refer to chapter "Administration and Provisioning tasks".

* To replicate storage devices on ETERNUS DX storage systems refer to "Replication tasks".

Description of the StorMan interfaces:

* All CLI commands can be used. For the syntax description refer to chapter "StorMan for programmers – CLI description".

Description of special issues depending on the used storage system:

* Depending on the used storage system some functions to configure storage resources may not be available with StorMan (e.g. creation of logical volumes). Refer to chapter "Special properties of storage systems".

1.3.4 Storage Management for SE Servers - Integration in SE Manager

For integration with SE Servers, StorMan is integrated by default as an add-on pack in the SEM of the M2000. For further details, please refer to "Installing StorMan Server on M2000".

The StorMan WebUI functionality is completely integrated into SEM. The complete functionality of StorMan is also available on CLI level for the user role service on M2000.

The main focus for StorMan on M2000 is the management of the storage resources of the SE Server. For SE-Servers StorMan provides additional functions on its WebUI to provide information about further Storage Ressources (tape storage) and to link additional storage management software (e.g. ETERNUS SF).

Description of the StorMan tasks:

●   For Information and Monitoring of Storage Ressource of the SE Server please refer to the StorMan online help.

●   For configuration and administration of storage resources to be managed by StorMan refer to chapter "Setting up StorMan configurations".

●   To provide hosts with resources managed by StorMan refer to chapter "Administration and Provisioning tasks".

●   To replicate storage devices refer to "Replication tasks".

Description of the StorMan interfaces:

●   All CLI commands are reserved for the service only. For the syntax description refer to chapter "StorMan for programmers – CLI description".

●   The StorMan WebUI is completely integrated into the SEM on M2000 and is launched from the SEM. For basic information about startup and using the WebUI refer to chapter "The graphical user interface – StorMan WebUI". Please note that the complete StorMan WebUI function reference is only contained in the online help system.

For the specific installation and deinstallation tasks on M2000 please refer to chapter "Installation by the SE Manager" and "Deinstallation of StorMan on M2000".

1.3.5 Integration in SQ Manager

For integration with SQ Series Business Servers, StorMan is integrated by default as an add-on package in the SQ Manager of the Management and Remote Service Console (MARS). For further details, please refer to "Operation and Administration" manual [3]".

The StorMan WebUI functionality is completely linked into and launched from the SQ-Manager. The complete functionality of StorMan is available on CLI level for the user role administrator on MARS.

The main focus for StorMan on MARS is the management of the storage resources of the SQ-Server especially to support HA scenarios.

Description of the StorMan tasks:

•   For configuration and administration of storage resources to be managed by StorMan refer to chapter "Setting up StorMan configurations".

•   To provide hosts with resources managed by StorMan refer to chapter "Administration and Provisioning tasks".

•   To replicate storage devices on ETERNUS DX storage systems refer to "Replication tasks".

Description of the StorMan interfaces:

•   All CLI commands can be used in this scenario (however, not all may be applicable). For the syntax description refer to chapter "StorMan for programmers – CLI description".

•   The StorMan WebUI is completely integrated into the SQ Manager on MARS and can be launched from the SQ Manager. For basic information about startup and using the WebUI refer to chapter "The graphical user interface – StorMan WebUI". Please note that the complete WebUI function reference is only contained in the online help system.

•

For the specific installation and deinstallation tasks on MARS please refer to chapter 5.7 and 6.5.

1.4 Notational conventions

The following fonts and symbols are used in this manual to indicate different types of information:

Font/Symbol	Indicates	Example
AaBbCc123	System outputs are written in this typewriter font.	http port 5988
AaBbCc123	Commands entered by the user are written in typewriter font bold	
Italic text	Program interface functions and menus are written like this.	Select the menu *Add new host*
" "	Manual, chapter and section titles are enclosed by double inverted commas.	See chapter 3, "Installation"
►	The enter symbol indicates an action that the operator must perform (e.g. enter something on the keyboard).	► Click the *Attach* button.
⚠	Indicates warnings.	
i	Indicates important information about product operation.	

The following fonts and symbols are used in the CLI syntax description:

Font/Symbol	Indicates	Example
AaBbCc123	Syntax elements of function call are written in this typewriter font.	storcfg
Italic text	Variables are written in this typewriter font	*function* or *hostname*
{ }	Braces enclose alternatives; one of the possible values shown within the braces must be entered.	{-name hostname | -ip ip}
|	A bar serves to separate alternative parameters or values.	-name *hostname* | -ip *ip*
[ ]	Square brackets enclose optional parameters, i.e parameters which may be omitted.	[-interface *type*]
[,...]	This entry after a value signifies that a list of values can be specified.	-addip *ip*[,...]

2 Functions and Architecture

2.1 Functional structure

From the structural point of view, StorMan consists of multiple functional components:

- Storage provisioning

- Storage information and monitoring

- Statistics and performance monitoring

- Storage system based replication functions

These components can be used separated or combined depending on the needs of the configuration.

2.1.1 Storage provisioning

The storage provisioning functionality of StorMan covers the high-priority functions required for storage virtualization and dynamic storage provisioning, namely:

- Managing and administrating storage resources (storage volumes and pools) in a virtualization layer

- LUN creation and deletion

- LUN masking (managing the hosts' access to volumes)

These features are provided by StorMan's WebUI
(see the chapter "The graphical user interface – StorMan GUI")

and as CLI for automation support in scripts
(see the chapter "StorMan for programmers – CLI description").

StorMan supports the Storage Provisioning features for following storage systems:

Storage system	Vendor	Supported connectivity type
ETERNUS DX410/440/8700 S2	Fujitsu	FC
ETERNUS DX500/600 S3	Fujitsu	FC
Symmetrix VMAX	EMC Corporation	FC

2.1.2 Storage information and monitoring

The storage information and monitoring features supported are those required for information and for monitoring storage arrays mainly for integration purposes, namely:

• Discovery

• Information about logical and physical components

• Monitoring of status

These features are implemented by the CLI.

StorMan supports the Information and Monitoring features for following storage systems:

Storage system	Vendor	Supported connectivity type
ETERNUS DX60/80/90	Fujitsu	FC / iSCSI
ETERNUS DX60/80/90 S2	Fujitsu	FC / iSCSI
ETERNUS DX100/200/500/600 S3	Fujitsu	FC / iSCSI
ETERNUS DX410/440/8700 S2	Fujitsu	FC / iSCSI
FibreCAT CX	Fujitsu	FC / iSCSI
CLARiiON CX	EMC Corporation	FC / iSCSI

2.1.3 Statistics and Performance

The statistic and performance features supported are required for monitoring storage arrays mainly as used in by openSM2 and suitable for Managed Storage Services implementations.

The statistic and performance monitoring features are provided on volume (LUN) level, namely the following:

• reads per second

• writes per second

• KB read per second

• KB write per second

• Average IO times for reads and writes

These features are implemented by the CLI.

StorMan supports the Statistics and Performance features for following storage systems:

Storage system	Vendor	Supported connectivity type
ETERNUS DX410/440/8700 S2	Fujitsu	FC / iSCSI
ETERNUS DX100/200/500/600 S3	Fujitsu	FC / iSCSI
Symmetrix VMAX	EMC Corporation	FC

2.1.4 Replication

The Replication Service features cover the information and management of local and remote mirroring functions implemented in the storage systems based on volume level replication.

These features are provided by StorMan's WebUI
(see the chapter "The graphical user interface – StorMan GUI")

and as CLI for automation support in scripts
(see the chapter "StorMan for programmers – CLI description").

Local Replication

The Local Replication Service covers the information and management of local mirroring functions used e.g. by SHC-OSD to integrate ETERNUS DX storage systems in BS2000.

The Local Replication Service features support the management of the following functions:

- Local mirroring based on full volume copies (clones)
- Local mirroring based on pointer based technologies (snaps)

StorMan supports the following storage systems:

Storage system	Vendor	Local replication type
ETERNUS DX410/440/8700 S2	Fujitsu	Clones and Snaps
ETERNUS DX500/600 S3	Fujitsu	Clones and Snaps

Remote Replication

The Remote Replication Service covers the information and management of remote mirroring functions used e.g. by SHC-OSD to integrate ETERNUS DX storage systems in BS2000.

The Remote Replication Service features support the management of the following function:

- Remote mirroring based on full volume copies

StorMan supports the following storage systems:

Storage system	Vendor	Remote replication mode
ETERNUS DX410/440/8700 S2	Fujitsu	Synchronous
ETERNUS DX500/600 S3	Fujitsu	Synchronous/ Asynchronous
Symmetrix VMAX	EMC Corporation	Synchronous

2.2 Logical structure for storage management

The main requirement for storage management is met by an abstract interface with a set of functions that can be invoked on behalf of the StorMan Client. The StorMan Client has to specify an object of storage management, e.g. the storage resource (identified by its storID) and in case of storage provisioning a server that needs to be provisioned with the storage resource.

2.2.1 Virtualization layer

The virtualization layer is a core part of StorMan. It administers the storage resource descriptions (configuration data). These descriptions are specific for each storage product. Therefore virtualization means providing a single interface for different kinds of descriptions.

Repository

A repository is required for consistent administration of the mapped resource descriptions. StorMan uses its own internal repository implemented as an internal database. While the database software is embedded into StorMan, its data can optionally be kept internally or externally in SAN or NAS environment depending on configuration.

Configuration data

Administration is divided into 2 layers:

* Configuration layer

   This layer, referred to as the storage pool layer below, handles storage configuration. The StorMan administrator defines the StorMan managed storage pools as the storage resources available for provisioning. Further on these storage pools administrated by StorMan are just called **pools** to differentiate them from the notion of storage pools used in the context of storage systems.

   In its simplest form, each pool identified by its unique pool ID describes one storage system of the entire storage configuration known by StorMan (e.g. an ETERNUS DX storage system identified by its serial number and a subset of volumes configured in this storage system). Each pool description contains general pool attributes and storage system- and volume-related attributes. The StorMan Server administrates the pool layer.

* Dynamic layer

   The existing storage assignments represented by a unique storage identifier (storID) can be managed using this layer. The current attribute and status of the provisioning functions are held in the dynamic layer for each storage identifier. Attributes of the storage identifier include the LUN of the storage volume, its current size, currently connected servers, its harbouring pool, etc. The storage identifier is created the first time storage is assigned and optionally deleted after de-assignment. An option of configuring a storage identifier statically is also provided for provisioning cases where a predefined storage (LUN) is required and was already configured.

   This data is kept in the StorMan repository with the same options for administration as described for the configuration layer.

In provisioning tasks that contain new assignments of storage such as creating a volume, the caller has to specify a storage resource and/or attributes of the storage that should be provided for the request. This description of the requested storage attributes is called storage policy below. Storage policy descriptions have a specific, extendable format containing information such as a specific pool.

An administration interface (WebUI) to configure and manage the Virtualization Layer's configuration data offers basic management functions (e.g. adding, removing, modifying and displaying pools and storage identifiers). Storage provisioning functions are also available on the WebUI.

Initialization of the configuration data when setting up a StorMan configuration in an existing storage environment is supported. StorMan managed pools can be configured based on the results of a discovery function. Existing assignments between server and storage are represented by internally generated storIDs in the configuration data.

Beside these mapping and administration tasks, the virtualization layer provides the necessary parameters for the storage layer.

2.2.2 Storage layer

The storage layer incorporates the implementation of the SMI-S client for the provisioning functions.

According to the SMI-S approach, the provisioning functions are transformed into the corresponding SMI-S calls. Distribution to the specific products is performed on the object manager level. According to the registered object managers of the vendor-specific SMI-S Providers, StorMan client will call the corresponding vendor-specific SMI-S Providers.

Proprietary interfaces of the vendors may be incorporated for special functions or if support is not yet provided by SMI-S.

Standardization: SMI-S

SMI-S is SNIA's standard in the domain of storage management.

The main functions for information and monitoring, storage provisioning (e.g. device creation, device masking) used by StorMan are defined in SMI-S as of V1.2 and implemented by storage vendors.

The implementation of replication functions is based on SMI-S as of V1.4.

Storage–vendor-specific interfaces (APIs)

StorMan additionally incorporates proprietary interfaces of storage vendors in its storage layer. StorMan uses them e.g.to monitor Tape Storage on SE Servers or to support NetApp storage systems by using ManageONTAP API.

2.3 Architecture

2.3.1 General

The following chapter covers the main lines of StorMan's architecture. The core part, i.e. the StorMan Server incorporating the virtualization layer and the storage specific layer is implemented as a portable server.
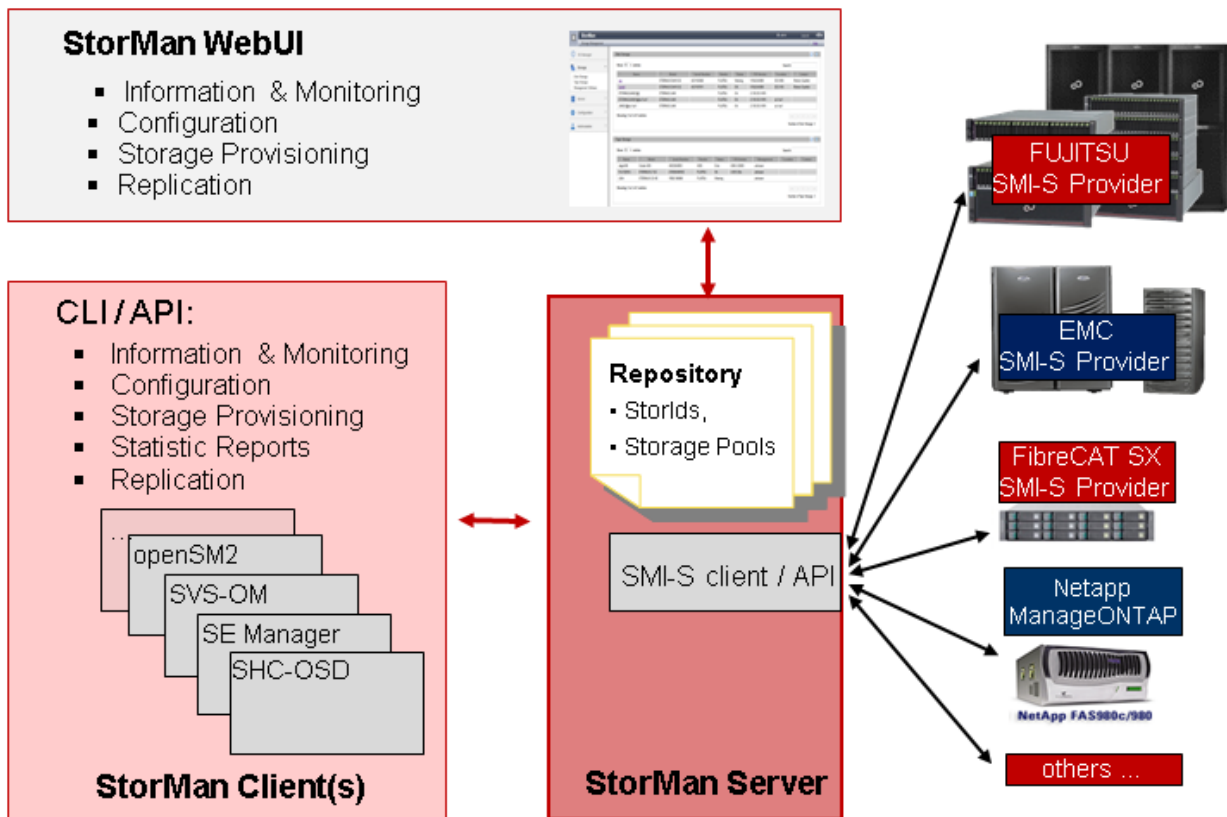
Figure 2: Architecture

The provisioning functions supported on the StorMan Client side are provided as CLIs (Command language interfaces). An API that provides the information and monitoring functions is available for internal purpose.

The StorMan Server is installed and runs as a service. It may be placed on any server of the configuration depending on the caller's needs and storage application requirements.

The StorMan Client part resides on the same server as the calling application and provides its caller with the management functions implemented by the StorMan Server. The calling application is not aware of the communication between the StorMan Client and StorMan Server. Therefore the StorMan Client and StorMan Server can be installed on the same server or on different servers connected by LAN.

2.3.2 Authorization

The StorMan Server runs on its hosting server as a service with administrator rights. Protection for this user identification can be set depending on the server and the platform.

The administration WebUI connects to the server that hosts the StorMan Server by specifying the configurable port number (default value 4178).

2.3.2.1 StorMan User

Access to the StorMan Server can be controlled by assigning a userid and password managed by StorMan using the command `storauth`. If a userid and password protection is set, for any call from CLI the combination of userid and password must be specified. To access the WebUI, user and password must be specified in the login screen.

On SE Servers the StorMan WebUI access is done directly from SE Manager for all authorized users. All users with SE Manager roles Administrator and Storage Administrator are allowed to access StorMan. CLI calls are allowed on the MU only for users of group service.

Additionally, the internal user administration of StorMan has to be maintained in parallel to manage the StorMan Roles.

2.3.2.2 Role Concept

StorMan usage requires the possibility to control the access and management of the storage configuration, as in typical configurations only a part of the connected storage is dedicated to a specific server or application.

The StorMan Role Concept provides the following functions:

● Define different users to manage specific storage resources for different applications. At least one administrator is empowered for all functions and all resources.

● The StorMan user and role management is reserved for the administrator.

● The administrator assigns roles to users authorizing them for administration of specific storage resources (storage pools) or information only.

Three predefined StorMan Roles are supported.

The storage administrator dynamically assigns the StorMan Roles to StorMan users to empower them for specific StorMan pools.

The Storage Pool concept provides the possibility to virtualize and partition the available storage resources.

2.3.2.3 StorMan Roles

The StorMan Role enables a StorMan User for a specific task. The following predefined StorMan Roles can be assigned to a StorMan User.

● Storage Administrator (StorAdmin) - This is the StorMan Role for the storage. The owner of this role is enabled for all StorMan functions for all StorMan Pools including the management of StorMan Users and their Roles.

● Pool Administrator (PoolAdmin) – This is the StorMan Role for administrators of specific storage resources (StorMan pools).The functions are applicable for (one or multiple) StorMan pools as defined by the Storage Administrator.

● Information (StorInfo) – This is the StorMan Role to show and monitor activities on the storage configuration in a datacenter. The owner is authorized for the complete information functionality of StorMan including performance and statistical data. But he is not authorized for any changes or modifications.

The StorMan Roles consist of a defined and fixed set of StorMan Authorizations.

The following overview shows the assignment of StorMan Authorizations to StorMan Roles.

StorMan Authorization Name	StorMan Role		
	StorAdmin	PoolAdmin	Info
StorMan Settings and parameters	X		
StorMan User Management	X		
Storage overall configuration	X		
Pool specific configruation	X	X	
Pool specific provisioning	X	X	
Pool specific replication	X	X	
Statistical and performance data	X	X	X
Information and Monitoring	X	X	X

2.3.2.4 Role Concept of MU and StorMan Roles

The Role Concept of MU defines an overall role for 'Storage Administrator'. Only users with this role are authorized for StorMan calls. By default all allowed SE Manager users get the StorMan role StorAdmin. The StorMan specific assignment of roles and pools is done by StorMan and may restrict the default role StorAdmin. The final authorization of the caller is done by StorMan internally, according to the StorMan Role and Pool of the StorMan User.

2.3.2.5 Access to the SMI-S Providers

Access to the SMI-S Providers and the managed storage systems is normally restricted to authorized personnel and protected by user and password combinations.

2.3.3 Communication and security

The core communication is based on Socket connections in sending the functions invoked on the StorMan Client to the StorMan Server and in receiving the respective results and return codes. Using the CLI functions the communication handling is encapsulated in the CLI call, i.e. a session is established for each call.

The WebUI provides administration and configuration functionalities, such as managing of pools (pool IDs) and StorIDs on the StorMan Server and management functions for storage provisioning. The WebUI applications can be called from any workstation within the LAN.

The communication between CLI and StorMan Server as well as between StorMan WebUI and StorMan Server is secured by SSL encryption.

2.3.4 Repository

Configuration data is contained in an internally consistent repository. The GUI or the CLI can be used for the administration function of pools and storage identifiers in the repository.

The repository is located in the following directory:

- `%PROGRAMFILES%\Fujitsu\StorMan\repository` (Microsoft Windows)

- `/var/opt/SMAWstor/StorMan/repository` (Linux, M2000, MARS)

There is no access for the user to view or modify the repository, but it is possible to change its location, e.g. when a high availability solution is configured (see section "High availability").

2.3.5 Error handling

Management tasks are critical for the calling application. A stable reaction and detailed return information are guaranteed for the caller of StorMan for functions terminated with errors or for which the response from called storage system is missing.

An internal logging function that logs all activities of the StorMan Server in a specific log file is available. The log file is primarily used for diagnostic purposes for StorMan.

2.3.6 High availability

High availability has to be provided by all components of a storage configuration:
StorMan Server, StorMan Client and SMI-S Providers.

This is of special interest in HA configurations for SQ Servers supporting Storage Clusters by the product HAPST. For further details, please refer to "Cluster Solutions for **SQ Servers**" manual [5]".

StorMan Server

The StorMan Server is designed as a single server and can therefore be considered as a single point of failure for Storage Management.

To provide high availability, a second redundant StorMan Server can be set up to take over requests from the StorMan Clients. These redundant StorMan Servers are optionally integrated with clustering software.

To support integration into clustering software, StorMan provides functions to start and stop the StorMan Server on Linux platforms and a check function to check the status of the StorMan Server for all supported platforms.

The most critical part in a takeover scenario from a StorMan Server to a standby StorMan Server is the consistency of data in the internal repository. This can be achieved by:

1. Keeping the repository data of the internal database and the configuration data of both StorMan Servers externally in the SAN or NAS accessible from both servers. This can be defined by configuring the `DataDirectory` parameter in the `smserver.ini` file.

2. Ensuring that the StorMan Servers have read and write access to the database files on both hosts, e.g. by using a user group that is identically on both hosts.

3. Assuring that the 2 StorMan Servers are not running concurrently.

4. Supporting a "rolling upgrade" of the 2 StorMan Servers from one release n to release n+1 in this configuration.

StorMan Client

The StorMan Client is not critical from the HA perspective as it can easily access an alternative StorMan Server in the LAN.

SMI-S Provider

SMI-S Providers are implemented as proxy or embedded providers.

For SMI-S Providers running in proxy mode, high availability can be achieved by configuring a redundant provider and integration with clustering software.

In high-availability configurations where the SMI-S Provider resides on the same host(s) as the StorMan Server, it should be part of the same cluster.

Multiple SMI-S Providers can be configured in StorMan to manage the same storage system. The StorMan Server will automatically select an available SMI-S Provider to control the storage system, thus assuring a consistent view of the configuration when changing the SMI-S Provider usage.

Embedded SMI-S Providers (e.g. ETERNUS DX storage arrays) run directly on the storage array and are therefore available as long as the storage array.

Proprietary interfaces of storage vendors

Supported proprietary interfaces of storage vendors (e.g. ManageONTAP of NetApp in StorMan) are bundled with the StorMan Server. High availability is therefore assured by HA of the StorMan Server.

3 System requirements

3.1 Hardware

3.1.1 Server

StorMan supports following servers:

- All current PRIMERGY servers as StorMan Clients and StorMan Servers

> **i** For Storage Arrays managed by a proxy SMI-S Provider (as there are EMC Symmetrix DMX and CLARiiON CX), a server with IP connectivity to the managed storage arrays is required to host the CIM Server (SMI-S Provider).
>
> For EMC Symmetrix DMX Storage arrays the server requires FC connectivity to the managed storage arrays.

- StorMan can manage any server connected to external storage via FC, as the server and its HBAs are part of the StorMan configuration if the storage array is supported.

3.1.2 Storage

ETERNUS DX Storage

StorMan V6.0 supports the information and monitoring functions for:

- ETERNUS DX models DX60, DX80 and DX90 based on the embedded SMI-S Provider available as of microcode version V10L30.
- ETERNUS DX models DX60 S2 based on the embedded SMI-S Provider available as of microcode version V10L12.
- ETERNUS DX models DX60 S3 based on the embedded SMI-S Provider available as of microcode version V10L25.
- ETERNUS DX models DX80 S2 and DX90 S2 based on the embedded SMI-S Provider available as of microcode version V10L12.
- ETERNUS DX410/DX440 S2 and DX8700 S2 based on the embedded SMI-S Provider available as of microcode version V10L56.
- ETERNUS DX100/DX200/DX500/DX600 S3 based on the embedded SMI-S Provider available as of microcode version V10L20.

StorMan V6.0 supports the provisioning functions for

- ETERNUS DX410/DX440 S2 and DX8700 S2 based on the embedded SMI-S Provider available as of microcode version V10L56.
- ETERNUS DX500/DX600 S3 based on the embedded SMI-S Provider available as of microcode version V10L20.

StorMan V6.0 supports the replication service functions for

- ETERNUS DX410/DX440 S2 and DX8700 S2 based on the embedded SMI-S Provider available as of microcode version V10L56.

- ETERNUS DX500/DX600 S3 based on the embedded SMI-S Provider available as of microcode version V10L20.

EMC Storage

StorMan V6.0 supports the provisioning functions for SAN connected storage of EMC Corporation, namely the Symmetrix DMX, Symmetrix VMAX products as supported by SMI-S Provider V4.6 and Solutions Enabler V7.6.

> **i** For further details of supported Symmetrix models, please refer to the EMC Support Matrix (ESM) available at on the manufacturer's website.

StorMan V6.0 supports the information and monitoring functions required primarily for integration with ServerView for the current CLARiiON CX models as supported by SMI-S Provider V4.6 and Solutions Enabler V7.6.

StorMan V6.0 supports the replication service functions required primarily for integration in BS2000 with SHC-OSD for the Symmetrix VMAX models as supported by SMI-S Provider V4.6 and Solutions Enabler V7.6 as of Enginuity Version 5876.

StorMan V6.0 does **not** support the information and monitoring functions required e.g. for integration with ServerView for Symmetrix VMAX and Symmetrix DMX storage of EMC corporation.

NetApp Storage

StorMan supports the provisioning functions for the full range of SAN connected storage of NetApp® FAS (FAS2xx, FAS9xx, FAS3xxx) as supported by ManageONTAP 7.2.5 of NetApp (7-mode only).

StorMan V6.0 does **not** support the information and monitoring functions required e.g. for integration with ServerView for storage of NetApp.

3.2 Software

3.2.1 General requirements

The platforms supported are:

- Microsoft Windows Server 2008
- Microsoft Windows Server 2012
- SuSe Linux Enterprise Server (SLES) 10 SPx
- SuSe Linux Enterprise Server (SLES) 11 SPx
- Red Hat Enterprise Linux (RHEL) ES 5 and 5.x
- Red Hat Enterprise Linux (RHEL) ES 6.x
- Red Hat Enterprise Linux (RHEL) ES 7.x
- MARS V2.0
- M2000 V6.0

On the specific platform the StorMan WebUI additionally requires (pls. see "Installing the StorMan "):

- Web Server (e.g. Apache or IIS) must be configured

3.2.2 Additional software requirements

Further on the following software packages are necessary:

3.2.2.1 SMI-S Provider in 'embedded' Mode

SMI-S Providers running in embedded mode are installed and available directly on the Storage Processor. No additional software has to be installed. The embedded SMI-S Providers have to be enabled on the Storage System.

Currently embedded SMI-S Providers are available for the following storage arrays:

- ETERNUS DX60, DX80 and DX90
- ETERNUS DX 60 S2, DX80 S2, DX90 S2, DX410/DX440 S2, DX 8700 S2
- DX60 S3, DX100 S3, DX200 S3, DX500 S3, DX600 S3
- Symmetrix VMAX

ETERNUS DX60/DX80/DX90 Storage

As the SMI-S Provider for ETERNUS DX Storage is 'embedded', it is delivered by default with the storage array as part of the microcode and must be enabled.

Namespace

The interop name space for the provider is the default name space.

```
interop
```

Access to CIM-OM

Access to the CIM-OM is restricted to authorized users as described in the Fujitsu ETERNUS SMI-S Provider Release Notes.

ETERNUS DX S2 and ETERNUS DX S3 Storage

As the SMI-S Provider for ETERNUS DX Storage is 'embedded', it is delivered by default with the storage array as part of the microcode and must be enabled.

Namespace

The interop name space for the provider is the default name space.

```
interop
```

Access to CIM-OM

Access to the CIM-OM is restricted to authorized users as described in the Fujitsu ETERNUS SMI-S Provider Release Notes.

Statistical and Perfomance data

PMCC must be installed and the installation path has to be set in `smserver.ini` using parameter `PMCC_InstallationFolder`.
On Linux systems the permissions for execution of PMCC CLI "`stxpmonitor`" must be allowed for user "`storman`" (e.g. by `chmod a+x /opt/FJSVstxp/bin/stxpmonitor`). This is not required on M2000.

The access data from PMCC to the storage system must be defined in StorMan using:

```
storcfg system –mod –system <system name> –statuser <user> –statpwd <pwd>
```

3.2.2.2 SMI-S Provider in Proxy Mode

EMC Storage

- SMI-S Provider V4.6 for SMI-S V1.2 and higher from EMC Corporation.

  SMI-S Provider V4.6 is based on EMC Solutions Enabler V7.6.

For further details of supported Symmetrix and CLARiiON CX models, please refer to the EMC Support Matrix (ESM) and to EMC SMI-S Provider Version 4.6. Release Notes available on the manufacturer's website.
The following operating systems are supported (as of EMC Support Matrix):

– Microsoft Windows Server 2008 R2 and 2012 Windows Enterprise Edition

– Red Hat Enterprise Linux (RHEL) ES 5.x, ES 6.x

– SuSe Linux Enterprise Server (SLES) 10 and 11

- The EMC SMI-S Provider is **not** part of the StorMan package:

  – The EMC SMI-S Provider is available as a download on the manufacturer's website for CLARiiON CX systems.

  – The EMC SMI-S Provider is embedded on the Service Processor for Symmetrix VMAX.

- An FC connection for the CIM Server (SMI-S Provider) is required to support EMC Symmetrix DMX storage arrays.

NetApp Storage

As for StorMan V6.0 the support of NetApp® FAS storage is based on ManageONTAP API; **no** SMI-S Provider is required.
Only ONTAP API in 7-mode is supported!

3.2.3 Installation and start of EMC SMI-S Provider

3.2.3.1 Download

Windows and Linux platforms

Installation instructions for Windows and Linux are contained in the EMC Release Notes.

3.2.3.2 Special issues for CLARiiON CX systems

If you are working with CLARiiON CX systems, check or carry out the following on the server on which the EMC SMI-S Provider is installed:

- The SMI-S Provider can be prevented from discovering single CLARiiON CX systems with the `claravoid` file. This file is contained in the directory

  – `%PROGRAMFILES%\EMC\SYMAPI\config` (Microsoft Windows)

  – `/var/symapi/config` (Unix/Linux)

  and includes one line per CLARiiON CX with the serial number of the CLARiiON CX.

3.2.3.3 Special issues for Symmetrix DMX systems

- The SMI-S Provider Host has to be connected to the Symmetrix DMX by FibreChannel.

- The feature Device (LUN) Masking delivered by "EMC VolumeLogix" has to be enabled on the Symmetrix DMX and on every FC director port on which the application hosts are connected (VCM-enabled).

- A VCM device has to be configured in the Symmetrix system and the VCM database has to be initialized. Please refer to the "EMC Solutions Enabler Symmetrix Device Masking CLI Product Guide", available on the manufacturer's website, for information on how to do this.

- The VCM device should be mapped to the SMI-S Provider Host to permit the SMI-S Provider to use it as a gatekeeper device before performing any device masking actions.

- After the VCM device is created and mapped, execute `symcfg discover` and `sympd list` on the SMI-S Provider host to make sure that the VCM device is visible to the host

   To avoid conflicts with gatekeeper devices, you are recommended to set the following parameter in the `.../symapi/config/options` file:

   ```
   SYMAPI_WAIT_ON_LOCKED_GK          = ENABLE
   ```

- If the VCM device is visible to the host, StorMan via SMI-S Provider can attach additional gatekeeper devices itself. You are strongly recommended to attach at least 3 additional gatekeeper devices, because a `storapi` demon and a `storevnt` demon are also running on the SMI-S Provider host (see also the Release Notes for EMC SMI-S Provider). If only the VCM device is visible for the host (no additional gatekeepers), the SMI-S Provider may receive IO Errors because this device (used as a gatekeeper) is in use.

- *Note:* If additional gatekeeper devices are removed for the host later on, the SMI-S Provider has to perform a Symmetrix discovery again before it can continue working with the VCM device only (by calling `storcfg cimom -discover`).

- The SMI-S Provider can also be prevented from discovering single Symmetrix DMX systems with the `symavoid` file. This file includes one line per Symmetrix with the serial number of the Symmetrix. The file is contained in the directory:

   ```
   %PROGRAMFILES%\EMC\SYMAPI\config (Microsoft Windows)
   /var/symapi/config (Unix/Linux)
   ```

3.2.3.4 Special issues for Symmetrix VMAX systems

If you are working with Symmetrix VMAX systems the embedded SMI-S Provider running on the Service Processor is recommended.

- No FibreChannel connected '*gatekeeper*' device is required to manage the Symmetrix VMAX system.

- Access to the SMI-S Provider is restricted to its https port (default 5989).

3.2.3.5 CIM-OM Settings

The following chapters describe the setting of the SMI-S Provider recommended for usage in StorMan configurations. They are defined in the parameter file `\emc\ECIM\ECOM\Providers\OSLSProvider.conf` in the installation directory.

The SMI-S Provider has to be restarted to activate modified settings.

Please note that these settings should be saved as they are removed at deinstallation time.

Please refer to EMC Release Notes for further information.

Access to CIM-OM

Access to the CIM-OM is restricted to authorized users as described in the EMC Release Notes. Please specify the cimuser and cimpassword when adding the specific CIM-OM to the StorMan configuration (command `storcfg cimom –add … -cimuser … -cimpwd`).

Synchronous processing of replication function

Synchronous processing of replication functions is required by StorMan. Therefore the following parameters have to be set in:

`\emc\ECIM\ECOM\Providers\OSLSProvider.conf:`

`OSLSProvider/com.emc.cmp.osls.se.array.ReplicationService.replica.creation.action.synchronous = true`

`OSLSProvider/com.emc.cmp.osls.se.array.ReplicationService.replica.modification.action.synchronous = true`

Enable remote replication

The remote replication function is disabled by default. It can be enabled in the SMI-S Provider if remote replication is configured. This is required to support HA configurations for Symmetrix VMAX:

`OSLSProvider/com.emc.cmp.osls.se.array.ReplicationService.remote.enable = true`

LUN masking on Symmetrix VMAX

When LUN masking is to be performed on Symmetrix VMAX systems, the following parameters have to be set:

`OSLSProvider/com.emc.cmp.osls.se.array.LunMask.map.operation.synchronous = true`

`OSLSProvider/com.emc.cmp.osls.se.array.LunMask.unmap.operation.synchronous = true`

On Symmetrix VMAX systems, volumes are mapped implicitly by the SMI-S Provider during an attach operation, if they are not yet mapped to the target ports, but per default they are not unmapped during a detach operation. If it's preferred to unmap volumes during detach operations the provider can be configured to do that by:

`OSLSProvider/com.emc.cmp.osls.se.array.LunMask.unmap.operation.enable`

Start the Provider

After these items have been clarified, restart the SMI-S Provider.

− On Linux Unix systems, perform the following steps:

    − Stop the CIM Server:

        `–kill SIGTERM <PID of ECOM process>`

    − Start the CIM Server:

        `/opt/emc/ECIM/ECOM/bin/ECOM –d`

    − Check the CIM Server / SMI-S Provider (optional):

        `/opt/emc/ECIM/ECOM/bin/ECOM –v`

    and check the process named 'ECOM –d'.

        `ps –ef|grep 'ECOM –d'`

− On Windows systems, this can be performed via the 'Services' .

    − stopping the CIM Server:

        `\...\ EMC\ECIM\ECOM\bin\sm_service stop ecom.exe`

    − starting the CIM Server:

        `\...\EMC\ECIM\ECOM\bin\sm_service start ecom.exe`

> **i** Fujitsu provides no warranty for these tools. Refer to EMC Corporation latest Release Notes for details.

4 Functions for SE Servers

4.1 Storage Support for SE Server

4.1.1 Overview

4.1.1.1 Storage Type and functional levels

StorMan V6.0 extends Storage Management integration for SE Servers supporting the following categories:

- Disk Storage
- Tape Storage
- Linkage to Storage Management Software

The functional level of the support is different depending on the storage type and the available interfaces.

- Active management support for Disk Storage as available via SMI-S (already available since StorMan V5.0)
- Information and Monitoring for Tape Storage and Disk Storage based on proprietary, vendor specific interfaces
- Listing of unmanaged storage systems by StorMan registration for any storage without administration interface or not supported by StorMan

4.1.1.2 Supported Storage Systems

The following table shows the Storage System support in StorMan V6.0:

Storage Model / Series	Vendor	Type	SE Location	StorMan Interface	Comment
ETERNUS DX60 / DX80 / DX90 S2	FUJITSU	Disk	intern / extern	SMI-S	Information and Monitoring
ETERNUS DX4x0 S2	FUJITSU	Disk	intern / extern	SMI-S	Active management
ETERNUS DX 8700 S2	FUJITSU	Disk	extern	SMI-S	Active management
ETERNUS DX100 / DX200 S3	FUJITSU	Disk	intern / extern	SMI-S	Information and Monitoring
ETERNUS DX500 / DX600 S3	FUJITSU	Disk	intern / extern	SMI-S	Active management
ETERNUS JX40 S2	FUJITSU	Disk	Intern	CLI	Information and Monitoring if connected to SU X86
ETERNUS LT40/60 S2	FUJITSU	Tape	intern / extern	CLI	Information and Monitoring; SNMP support

ETERNUS CS HE / CS8000 / CS50 / CS800	FUJITSU	Tape	Intern / extern	CLI	Information and Monitoring; proprietary CLI as of V5.1
Symmetrix VMAX Series	EMC	Disk	extern	SMI-S	Active management
Symmetrix DMX Series	EMC	Disk	extern	SMI-S (Proxy)	Information and Monitoring; external SMI-S server
Netapp FAS	NetApp	Disk	any	netappapi	Information and Monitoring: proprietary API
CLARiiON CX / FibreCAT CX	EMC	Disk	Extern	SMI-S (Proxy)	Information and Monitoring; external SMI-S server
Scalar i2k/i6k	Quantum	Tape	extern	CLI	Information and Monitoring; SNMP support
Scalar i500	Quantum	Tape	extern	CLI	Information and Monitoring; SNMP support
Scalar i10k	Quantum	Tape	extern	CLI	Information and Monitoring; SNMP support

4.1.2 Active Management

Active Management for Storage Systems is supported as documented in StorMan V5.0 Manual. Extensions for new HW models and FW versions are provided.

The storage systems must be registered first by storcfg cimom –add or by the StorMan WebUI specifying SMI-S as Management Instance (-interface smis).

4.1.3 Information and Monitoring

4.1.3.1 Common Aspects

StorMan V6.0 provides flexible and generic support of further Storage Systems not supported by SMI-S, based on vendor specific storage interfaces. These functions are provided for SE Servers on M2000 only.

The storage systems to be monitored have to be registered in StorMan first.

This can be done by WebUI specifying the model of the Storage system together with its access data.

4.1.3.2 Specific Aspects for Storage Models

The following chapters describe the storage specific support.

● Support for ETERNUS CS

StorMan on M2000 provides information and monitoring functions for ETERNUS CS HE, ETERNUS CS8000, ETERNUS CS800 and ETERNUS CS50 storage systems as of CS V5.1.

StorMan supports them based on the CLI of ETERNUS CS.

The registration in StorMan is done for each monitored ETERNUS CS system via WebUI.

To connect to the system IP address, user and password is required.

● Support for ETERNUS LT40/60 S2

StorMan on M2000 provides information and monitoring functions for ETERNUS LT40/60 S2 storage systems.

StorMan supports them based on SNMP V1 on ETERNUS LT by community name (default: public).

The registration in StorMan is done for each ETERNUS LT system via WebUI.

To connect to the system IP address and SNMP community is required.

● Support for Quantum Scalar Tape Libraries

StorMan supports them based on SNMP V1 on ETERNUS LT by community name (default: public).

StorMan in SEM environment provides information and monitoring functions for Quantum Scalar Tape Libraries.

StorMan supports them based on SNMP. They currently support SNMP V1, but are encouraging SNMP V3 usage for new models. The following models are supported:

• Quantum Scalar i2k / i6k: with SNMP V1 by community name (default: publicCmtyStr)

• Quantum Scalar i500: with SNMP V3 (default user: admin) or V1 (must be enabled on the library, default: publicCmtyStr)

• Quantum Scalar i10k: with SNMP V1 by community name (default: public)

The registration in StorMan is done for each system via WebUI.

To connect to the system IP address and SNMP community is required.

● Support Quantum Scalar I500

with SNMP V3: by user/pwd

with SNMP V1: by community (default: publicCmtyStr)

● Support Quantum Scalar I6000

with SNMP V3: by user/pwd

with SNMP V1: by community (default: publicCmtyStr)

● Support Quantum Scalar I10K

with SNMP V1: by community (default: public)

● Support for ETERNUS JX40

ETERNUS JX40 disk storage on SUx86 and AU only connects directly via SAS.

No LAN access is available for ETERNUS JX management. Information about the ETERNUS JX40 is only available directly from SUx86 via M2000.

ETERNUS JX40 connected to AU's are not supported.

No StorMan registration is required for ETERNUS JX system. StorMan discovers them by default.

4.1.4 Listing of unmanaged Storage Systems

StorMan V6.0 provides the option to add/remove/modify/show Storage Systems to its repository, without interfaces for management and or monitoring. These systems are listed in the Storage System Overview. Therefor static information is maintained in the StorMan repository, just to display a complete view of the storage environment.

Unmanaged Storage Systems registers by WebUI. or CLI (on M2000 for group service).

4.1.5 Linkage to Storage Management Software

For SE-Server StorMan V6.0 maintains links to overall Storage Management SW in StorMan repository. Storage Management SW can be registered and unregistered in StorMan for this purpose. This function provides access to external Storage Management SW out of the StorMan WebUI.
The functions are provided by WebUI.

4.2 StorMan Add-on Pack on M2000

4.2.1 Handling of StorMan as M2000 Add-on Pack

4.2.1.1 Definition

For SE-Servers StorMan is defined as Add-on Pack of M2000 named 'STORMAN' to assure independent installation and de-installation.

StorMan is part of the SW delivery of SE-Servers. The StorMan delivery is an independent media and part of M2000. However, the version is decoupled from M2000 versions. This is mandatory to assure an independent StorMan Support for new Storage HW/FW versions.

The installation of StorMan is mandatory on M2000. However, StorMan may be temporary not installed (e.g. during upgrade installation) or not available (stopped – due to any error condition).

The SEM functions for Add-On Packs handle Installation / De-installation of StorMan via SEM. StorMan does not require specific activation; it is automatically active after successful installation.

SEM has to check for installation and availability of StorMan to activate/deactivate calls for the Storage Management (URLs in SEM). The name of the Add-On Pack in M2000 is STORMAN.

No third party SMI-S Providers are supported on M2000. To Support EMC Symmetrix VMAX, StorMan accesses the embedded provider on the VMAX Service Processor. Support of other EMC storage systems requires an external SMI-S Proxy Server.

4.2.1.2 StorMan Authorization on M2000

As of V6.0 StorMan has its own Authorization and Role Concept. When running on M2000, StorMan adapts to the Authorization Concept of M2000.

StorMan authorization on M2000 depends on the access:

● CLI access is only supported for users of group service on MU with full access to StorMan CLI without additional authorization check in StorMan

● WebUI access is controlled on 2 levels.

  ● SEM users with role 'Administrator', 'Operator' and 'Service' are accepted by StorMan.

  ● The range of supported functions and manageable storage resources for these authorized users are controlled by the StorMan internal user and role concept.
  SEM users must be defined as StorMan Users with an assigned StorMan role before they can access StorMan.

A StorMan Role enables a StorMan User for a specific task. The following predefined StorMan Roles can be assigned to a StorMan User.

● **Storage Administrator (StorAdmin)** - This is the StorMan Role for the storage administrator). The owner of this role is enabled for all features required to set up and maintain StorMan including the management of StorMan Users and their Roles.

  The owner of this role is also enabled for the complete storage management and to create StorMan pools for dedicated storage systems, create/delete/assign volumes.

- **Pool Administrator (PoolAdmin)** – This is the StorMan Role for the administrator of specific storage resources (StorMan pools).The functions are applicable on explicitly assigned (one or multiple) StorMan pools.

- **Information (StorInfo)** – This is the StorMan Role to show and monitor activities on the storage configuration in a datacenter. The owner of this role disposes of the complete information functionality of StorMan including performance and statistical data, but no authorization for any changes or modification.

4.2.1.3 PMCC support as part of Add-on Pack StorMan

StorMan needs PMCC of FUJITSU to provide statistical and performance data for ETERNUS DX. On M2000 the suitable PMCC version is packed and delivered together with StorMan as part of the StorMan Add-on Pack.

StorMan V6.0 supports PMCC V2.4 (including ETERNUS DX S3 support).

Preconditions

The collection of statistical data is started/stopped either by StorMan (caller openSM2 or StorMan CLI) or via the ETERNUS DX WebUI. No other instance should use PMCC in parallel except of AST management via ETERNUS SF.

To provide the PMCC output data, StorMan creates a data subdirectory for every single ETERNUS DX System in path: `/var/opt/SMAWstor/StorMan/pmcc/<storage name>`.

Network Conditions:

Open the firewall between StorMan Server (on MU) and the ETERNUS DX for the following port numbers. On MU this is set up in M2000 up by default.

No.	Target device	Port number/ protocol	Connection direction at start
1	ETERNUS DX	1999/tcp	From the performance information acquisition server to ETERNUS Disk storage system
2	ETERNUS DX	1372/tcp	The Maintenance-Secure connection from the performance information acquisition server to ETERNUS
3	ETERNUS DX	22/tcp	The Maintenance-Secure connection from the performance information acquisition server to ETERNUS

Installation / De-installation

PMCC rpm (FJSVstxp-2.4.0-0) installs automatically during StorMan Add-on Pack installation (before StorMan installation).

PMCC rpm (FJSVstxp-2.4.0-0) de-installs automatically during StorMan Add-on Pack de-installation (after StorMan de-installation).

The installation requires 142 MB disk capacity for installation and additionally up to 25 MB for each monitored ETERNUS DX System (PMCC V2.4).

The following directories are used:

- Program directory   `/opt/FJSVstxp`
- Environment setup directory   `/etc/opt/FJSVstxp`
- Data directory   `/var/opt/FJSVstxp`

Normally the PMCC installation directory `PMCC_InstallationFolder` must be set in StorMan configuration file `smserver.ini`. On M2000 this is not necessary as the PMCC CLI 'stxpmonitor' is linked to the command path at installation time (`PMCC_InstallationFolder` remains an empty string).

4.2.2 Interaction of StorMan with M2000

SE-Server Information in StorMan

StorMan on M2000 automatically configures the SU's and MU's with their FC HBA information in its repository for SE-Servers. Based on this information StorMan is able to inform about the Server to Storage connection of the SE Server-Units and to provide as Storage oriented and a Server oriented view on the Storage resources

The SE-Server information is provided automatically as part of StorMan Add-on Pack installation time

4.2.3 Logging

StorMan as Add-on Pack supports the logging concept on M2000.

StorMan in 'stand alone' mode offers the StorMan logging for diagnostic purpose only.

4.2.3.1 System Logging

StorMan currently does not use the system logging in `/var/system/messages`.

4.2.3.2 StorMan Logging

StorMan uses its proprietary StorMan Logging facility in `/var/opt/SMAWstor/StorMan/log/` for diagnostic purpose. The logging level is set by CLI and/or StorManWebUI.

StorMan V6.0 introduces a maximal retention time for logging files. It is defined in the inititalization file `smserver.ini`.

The parameter `TraceFileMaxAge` defines the number of days that the log files are kept. The Default value is 0 (means that the files are never removed). E.g.:

`TraceFileMaxAge = 7` means 7 days

The effect of the parameter:

● StorMan log-files are zipped when reaching the size of 100 MB or latest when changing the date.

● All log-files and zip-files with content older than the number of days specified by `TraceFileMaxAge` are removed (i.e. all files starting with stormanTrace*).

4.2.4 Network considerations

4.2.4.1 StorMan client to StorMan server connections

StorMan client to server connections support socket connections with IPv6 and IPv4.

The communication on M2000 is secure (openssl encryption) by default.

WebUI connections to M2000 are secured via https.

4.2.4.2 Connections from StorMan server to Internal and External Storage

Common

For the CIMOM configuration in StorMan the IP-Address together with authentication data is specified. StorMan supports IPv4 and IPv6. The network connection itself between M2000 and Storage System is not relevant for StorMan. However, the firewall settings must allow the access.

Internal Storage

Internal Disk Storage

- ETERNUS DX S2 and S3 series
  have only one full LAN management port (MNT). Therefor connection to private AND public network is not possible (Additional service port e.g. for AIS connect is available).
  There are 2 options (recommended option 2):

  - Option 1: Connect to internal Control LAN:

    No management access from public Admin LAN possible → complete management must be done by SE Administrator only

  - Option 2: Connect to public Admin LAN:

    Management access via public Admin LAN only and by SE Administrator (StorMan)

- ETERNUS JX 40: no connection / no LAN management interface

Internal Tape Storage

- ETERNUS LT40
  has one LAN management port. There are 2 options:

  - Connect to private Management Network:

    No management access from public Admin LAN → complete management must be done by SE Administrator only

  - Connect to public Management Network:

    Management access via public Admin LAN only and by SE Administrator (StorMan)

External Storage

All external Storage (Disk and Tape) must be accessible via public Admin LAN.

5 Installation

5.1 Installing the StorMan Server on Linux

You have to be logged in as `root`. Enter the following command to install StorMan:

```
rpm -i StorMan-<version>.<platform>.rpm
```

i
- See the readme file for the correct version.
- Use the 64 bit version <platform> = x86_64 if suppported by your platform.
- 32 bit support is provided by <platform> = i586
- Use -iv or -ivv for extended information.
- A StorMan userid is created during installation. This userid is normally called `storman`. The password is locked. Only if a userid `storman` already exists a user `storman1` is created. If necessary, `storman2`, `storman3` , etc. are created accordingly.

If an earlier version of StorMan is already installed, it must be deinstalled first. Please refer to section "Deinstallation" for information on deinstallation.

5.2 Installing the StorMan Server on Windows

i
An installed previous version of StorMan must be deinstalled first. Please refer to section "Deinstallation" for information on deinstallation.

Perform the following steps to install StorMan:

► Log in with "administrator" rights.

► Execute

```
StorMan_Setup-<version>-<platform>.exe
```

- Use the 64 bit version <platform> = x64 if suppported by your platform.
- 32 bit support is provided by <platform> = x86.

► Select the installation language and click *OK*:
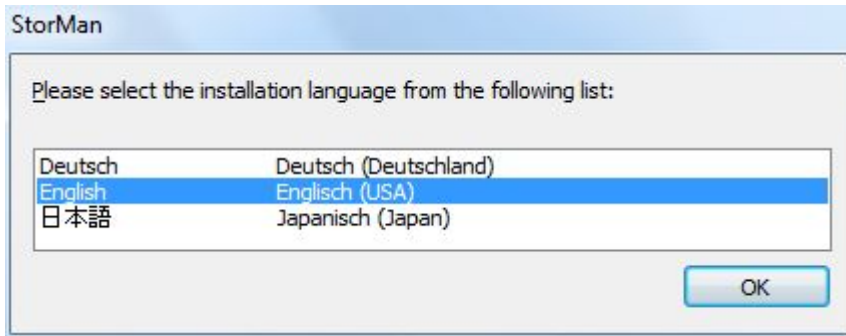
► Click *Next* in the Installation Welcome window:

► Select *I agree to these terms and conditions* in the License Agreement window and click *Next*:
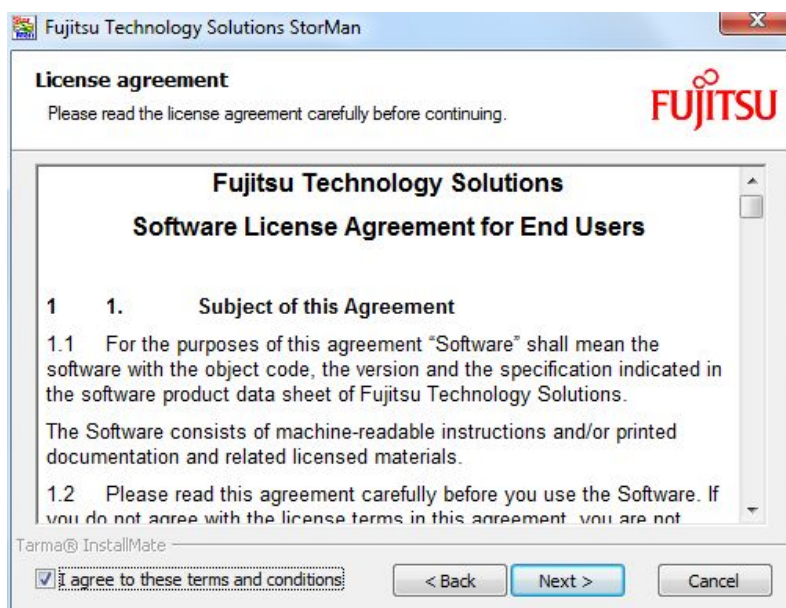
► Select your installation options. You have the choice of installing any combination of the Server, the Client or the WebUI. Then click *Next*.
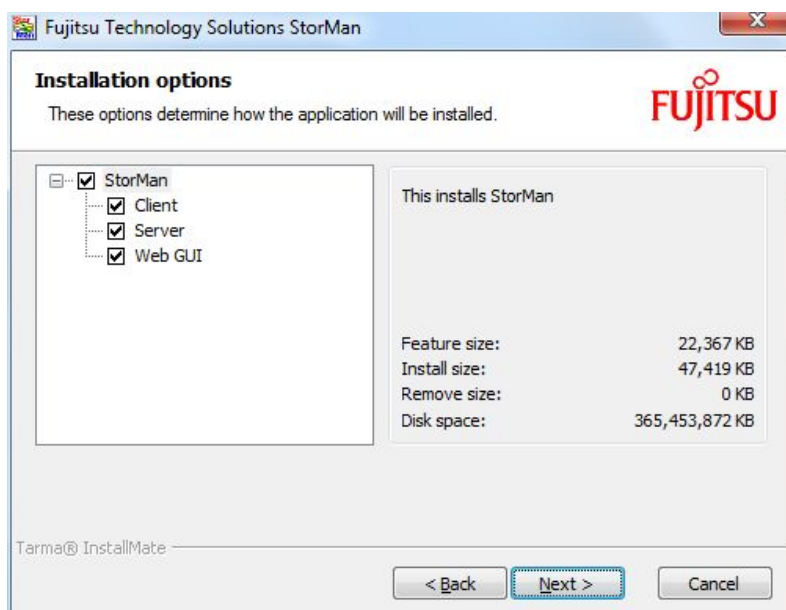
► Specify the installation folder and the StorMan Server port for StorMan communication if you need to change the default values. Afterwards click *Install*.
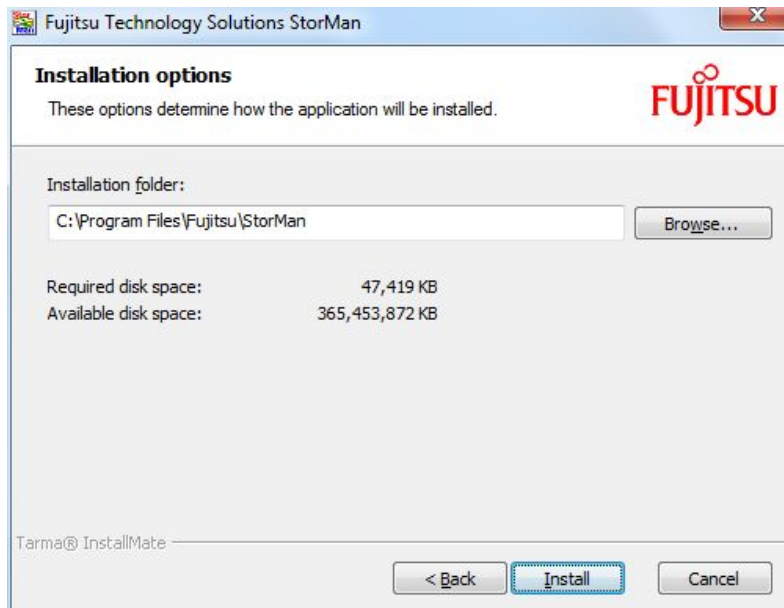
− Port number 4178 (default value) is a registered port at IANA which is reserved for StorMan. However, a different port number may be used, but it must be specified by StorMan Clients for each connection.

− If only the StorMan Client and / or the WebUI have been selected for installation, only the installation folder is offered for optional specification.

The following screen shows that installation has been successful completed:
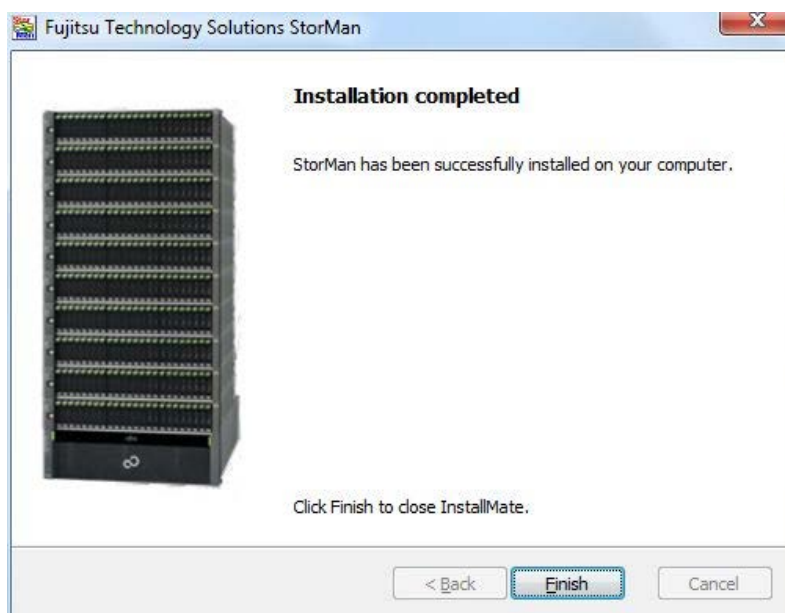
5.3 Configuration file

Configuration settings for the StorMan Server are contained in the `smserver.ini` file. It is the basic configuration file.

Most importantly, it defines the port number of the StorMan Server (`StormanServerPort` parameter). Note that StorMan uses one port with the number 4178 as the default. This port was assigned to StorMan by the Internet Assigned Numbers Authority (IANA). If necessary, you can edit the file to change the port number. All StorMan clients must specify the port number (4178 is used as default) to communicate with StorMan server.

On Linux systems `smserver.ini` is stored in the following directory:

    /var/opt/SMAWstor/StorMan/config

On Windows systems `smserver.ini` is located in the StorMan installation directory:

    %PROGRAMFILES%\Fujitsu\StorMan\config (as the default)

The file is the same on Linux and Microsoft Windows:

```
 [StorMan]
#
#=====================================================================
#        StorMan server initialisation file: smserver.ini
#=====================================================================
#
# ----- StorMan  parameters -------------------------------------------
#
# -- set variable path name for the data directory
# for subdirectories backup, log and repository:
# default values:
# - Windows: the installation home directory
# - others:  the installation home directory
#            or the environment variable STORMAN_HOME_DATA if set.
# DataDirectory = /var/storman
#
#
# -- port number for the StorMan server:
# default value: 4178
StormanServerPort = >>serverport<<
#
# -- usage of openssl:
# values:
#  1  suppress non-ssl requests
# default: 0
#OpenSslUsage = 0
#
# -- Timeout value for connect to CIMOMs:
# value range: <1 - 6000>
# default value: 10
#CIMOMConnectTimeOut = 10
#
# -- Timeout value for StorMan client calls in seconds:
# value range: >= 0, 0 indicates unlimited
# default value: 600
#ClientCallTimeOut = 600
#
#
# -- support of authorization:
# values:
#  0  authorization disabled
#  1  authorization necessary, show allowed
#  2  authorization necessary, for show too
# default value: 1
#AuthorizationLevel = 1
#
#
# ----- Parmeters of the CIMOM (SMI-S Provider) ---------------
#
# -- Timeout value for CIMOM calls:
# value range: <0 - 6000>
# default value: 600
#CimTimeOut = 600
#
# ----- Parameters for function storcreate
#
# -- Timeout value for CIMOM calls of function storcreate:
# should be less than CimTimeOut and ClientCallTimeout
# value range: <0 - 6000>
# default value: 480
# 0 = unlimited
# CreateTimeout = 0
#
# ----- Eternus statistic -------------------------------------------
```

```
#
#
#
# -- PMCC installation folder
# necessary for retrieving performance data
#PMCC_InstallationFolder = C:\Program Files\ETERNUS-PMCC
#
#
# -- PMCC interval time
# interval time for pmcc to store data
# default value: 180
#PmccIntervalTime = 180
#
# -- Refreshtime for Eternus statistics:
# value range: <30 - 6000>
# default value: 240
#StatisticRefreshTime = 240
#
# -- Number of refreshes without statistic retrieval before stop:
# value range: <0 - 6000>
# 0 means no stop
# default value: 4
#NumRefresh4Stop = 4
#
# ----- SEM ----------------------------------------
#
#
#
# -- sem session command
# the command to use for session read
#SemSessionCmd = semSession
#
# ----- Trace Levels -------------------------------------------
#
# -- maximum level to trace
# values:
#  <0 trace all
#  0  trace severe errors only
#  1  trace errors too
#  2  trace even warnings
#  3  trace information
#  4  trace debug
#  5  trace extended debug
# default value: 4
#MaxTraceLevel = 4
#
# -- trace communication:
# only for enabled debugging (MaxTraceLevel >= 4)
# values:
#  0  trace off
#  1  trace tcp/ip
#  2  trace messages
#  4  trace cim tcp/ip
#  8  trace cim messages
#  add the values to enable the traces
# default value: 0
#DebugCom = 0
#
# -- trace communication and debugging info of worker threads
# values:
#  0  trace off
#  1  trace worker
#  2  trace standard updater
#  4  trace mirror updater
#  8  trace CIMOM status updater
```

```
# 16  trace statistics collector
# 32  trace main thread
# 64  trace indication handler
# 128 trace script updater
# 256 disable client trace
#  add the values to enable/disable the traces
# default value: 0
#TraceWorker = 0
#
# maximum age in days for trace files
# 0 means no limit
# TraceFileMaxAge = 7[StorMan]
#
```

Some other parameters can be modified by editing `smserver.ini` or by using the `storparam` command as described in section "storparam":

`CimTimeOut`	Represents the timeout value in seconds for requests to the CIM-OM. A request to the CIM Server is cancelled if there is no response. The default setting is 600 seconds.
`CreateTimeout`	Timeout in seconds for dynamic volume creation (if a new volume is created during a `storcreate` request). The default setting is `unlimited`. The value is ignored if the `timeout` parameter is specified with the `storcreate` command, see section "storcreate".
`DataDirectory`	Supports a variable path name setting for the data directory containing the subdirectories backup, log and repository. The default values are depending of the platform: –Windows: the installation home directory –others: the installation home directory or the environment variable `STORMAN_HOME_DATA` if set.

i All entries in the configuration file are optimized for StorMan and care should be taken if they are changed. In particular, entries not mentioned here should not be changed.

Files signed with `*.orig` are default backup files.

5.4 Starting and stopping the StorMan service

Upon startup, storman reads its configuration file. The server is configured to listen on the default port 4178 (or any other user-defined port).

Once the server has started and performed some initialization activities, it will do the work of listening for and answering requests from clients.

Linux

The StorMan service is configured by installation being started automatically by the `rc` scripts. It can also be started manually as `root`.

To start or stop StorMan on Linux systems, enter:

```
/etc/init.d/storman start | stop
```

To check if StorMan is running, enter:

```
storcheck
```

This function is useful for high-availability configurations where applications are monitored by HA tools. For details about this command, please refer to the CLI description in section "storcheck".

For high-availability configurations, starting and stopping the StorMan feature can be integrated into `rc` script processing

Windows

The StorMan service is configured to start automatically as a Windows service.

To start or stop StorMan on Windows systems use the service manager.

To check if StorMan is running, check Windows services for "StorMan Service" or enter

```
storcheck
```

in a command shell. This function is useful for high-availability configurations where applications are monitored by HA tools. For details about this command, please refer to the CLI description in section "storcheck".

5.5 Installing the StorMan WebUI (stand-alone mode only)

A Web Server must already be installed on the server.

Your document root will vary depending on how you have configured your web server.

- In case of Apache, the default document root is
  (example for Linux SLES11, Apache version 2.0+):

  `/srv/www/htdocs`

- The path for IIS documents is `c:\...\wwwroot` (Windows only).

5.5.1 Copying web documents (Apache Server)

The StorMan WebUI is delivered as a zip-file. After installation it is contained in the installation directory
.../StorMan/WebUI/WebUI-.zip (e.g. for Linux: /opt/SMAW/SMAWstor/StorMan/WebUI/WebUI.zip)
Perform the following steps to extract StorMan web documents into the document directory of your web
server (document root) (example for Linux):

► Create a StorMan directory:

```
mkdir –p /srv/www/htdocs/StorMan
```

► Extract all files and subdirectories from `/opt/SMAW/SMAWstor/StorMan/WebUI/WebUI.zip` into
the StorMan directory under your web server document root:

```
cd /srv/www/htdocs/StorMan
unzip /opt/SMAW/SMAWstor/StorMan/WebUI/WebUI.zip
```

5.6 Installing StorMan Server on M2000

StorMan is part of the official delivery for SE Servers. Therefore it will be ready installed at delivery of the
system. StorMan is available as Add-on Pack of the SE Server and can be managed idependently.

The following steps have to be performed in case of an update installation only.

On M2000 the installation of StorMan Server and StorMan WebUI runs automatically as StorMan is
integrated as an Add-on pack on M2000. Installation can be performed using the SE Manager.

Refer to "SE Server: Operation and Administration" manual [6].

5.6.1 Installation by the SE Manager

The SE manager provides the management functions for Add-on Packs in the "Update" tab of the
"Hardware > Server > Service" menu:

As first step start "Upload add-on package" to upload the StorMan software from a DVD, a local or a
network drive to MARS. The name of the StorMan iso image delivered for MARS is:

```
MV.STORMAN-<version>.iso
```

After the upload is completed the add-on pack STORMAN-<version> is displayed and the installation of
StorMan can be started by clicking the Install icon.

After successful installation the SE Manager displays the add-on pack with the status "installed".

If an earlier version of StorMan is already installed, it must be deinstalled first. Please refer to section
"Deinstallation".

5.6.2 Postinstallation steps

After successful installation the complete features of StorMan are ready available. If required the StorMan roles for the userids have to be defined or adjusted.

5.6.3 Configuration file

`smserver.ini` is the basic configuration file.

The settings in the StorMan configuration file can be modified by using the WebUI or command storparam. No direct changes in the configuration file are supported.

5.6.4 Starting and stopping the StorMan service

The StorMan service is configured by installation being started automatically by the rc scripts. It can also be started manually under an administrator or service account.

Upon startup, storman reads its configuration file. The server is configured to listen on port 4178.

To start or stop StorMan on M2000 enter:

```
storman -start | -stop
```

To check if StorMan is running, enter:

```
storcheck
```

5.7 Installing StorMan Server on MARS

StorMan is part of the official delivery for SQ Servers. Therefore it will be ready installed at delivery of the system. The following steps have to be performed in case of an update installation only.

On MARS the installation of StorMan Server and StorMan WebUI runs automatically as StorMan is integrated as an Add-on package on MARS. Installation can be performed using the SQ manager on the MARS server.

Refer to "SQ Server: Operation and Administration" manual [3].

5.7.1 Installation by the SQ Manager

The SQ manager provides the management functions for add-on packages in the "Update" tab of the "MARS management > Administration" menu:

As first step start "Upload add-on package" to upload the StorMan software from a DVD, a local or a network drive to MARS. The name of the StorMan iso image delivered for MARS is:

```
MV.STORMAN-<version>.mars.iso
```

After the upload is completed the add-on package `STORMAN-<version>` is displayed and the installation of StorMan can be started by clicking the Install icon.

After successful installation the SQ Manager displays the add-on package with the status "installed".

| i | If an earlier version of StorMan is already installed, it must be deinstalled first. Please refer to section "Deinstallation". |

5.7.2 Postinstallation steps

After successful installation the complete features of StorMan are available.

There are no extra installation steps required for the WebUI Installation. A link to the StorMan WebUI is automatically integrated in the Storage tab in the Dashboard of the SQ Manager.

5.7.3 Configuration file

`smserver.ini` is the basic configuration file.

The settings in the StorMan configuration file can be modified by using command `storparam`. No direct changes in the configuration file are supported.

5.7.4 Starting and stopping the StorMan service

The StorMan service is configured by installation being started automatically by the `rc` scripts. It can also be started manually under an administrator or service account.

Upon startup, storman reads its configuration file. The server is configured to listen on port 4178.

To start or stop StorMan on MARS enter:

```
rcmng start | stop storman
```

To check if StorMan is running, enter:

```
storcheck
```

6 Deinstallation

6.1 Deinstallation of StorMan on a Linux Server

To uninstall StorMan enter:

```
rpm -e StorMan
```

There are backup files and configuration files which are not removed as part of the uninstall process:

- In the directory `/var/opt/SMAWstor/StorMan/config/` the configuration file `smserver.ini` is not removed.

- In the directory `/var/opt/SMAWstor/StorMan/backup/` all backup files are not removed to make sure that your data is not lost with the deinstallation.

- In the directory `/var/opt/SMAWstor/StorMan/repository/` the StorMan repository containing the complete configuration managed by StorMan is not removed.

- In the directory `/var/opt/SMAWstor/StorMan/log/` the StorMan logging files are not removed.

If you are sure that you do not need these files any longer, delete the directory `/var/opt/SMAWstor/StorMan/` manually.

6.2 Deinstallation of StorMan on a Windows Server

To uninstall StorMan, use the Windows software deinstallation utility.

There are backup files and configuration files which are not removed as part of the uninstall process (default paths below):

- In the directory `%PROGRAMFILES%\Fujitsu\StorMan\config\` the configuration file `smserver.ini` is not removed.

- In the directory `%PROGRAMFILES%\Fujitsu\StorMan\backup\` all backup files are not removed to make sure that your data is not lost with the deinstallation.

- In the directory `%PROGRAMFILES%\Fujitsu\StorMan\repository\` the StorMan repository containing the complete configuration managed by StorMan is not removed.

- In the directory `%PROGRAMFILES%\Fujitsu\StorMan\log\` the StorMan logging files are not removed.

If you are sure that you do not need these files any longer, delete the directory `%PROGRAMFILES%\Fujitsu\StorMan\` manually.

6.3 Deinstalling the WebUI

Remove the files or the symbolic link described in section "Copying web documents (Apache Server)" from the Web Server document root.

6.4 Deinstallation of StorMan on M2000

The SE Manager can be used to deinstall StorMan on M2000 using the features to manage Add-on packages. (refer to SE Manager help and the "SE Server: Operation and Administration" manual [6]).

After successful deinstallation the complete CLI and WebUI features of StorMan are removed.

6.5 Deinstallation of StorMan on MARS

The SQ Manager can be used to deinstall StorMan on MARS using the features to manage Add-on packages. This can also be used for the deinstallation of the SMI-S Provider add-on package on MARS (refer to SQ Manager help and the "SQ Server: Operation and Administration" manual [3]).

After successful deinstallation the complete CLI and WebUI features of StorMan are removed.

6.6 Upgrade Installations

During the start of StorMan V6.0 installation it is automatically checked if a StorMan version is already installed and the upgrade can be done automatically.

However, if a data in place upgrade is planned that should save existing StorMan managed configuration the migration steps should be done as described below.

Migration from previous versions to StorMan V6.0 is supported as described below.

Data in place upgrades for StorMan configurations from previous StorMan Versions to StorMan V5.0 is supported "on the fly": During the upgrade process the StorMan repository is upgraded automatically.

| i | Please note that downgrading from StorMan V6.0 to a lower StorMan version is not supported. For this reason it is recommended to create a backup of the repository using `stormandb -dump` before starting the upgrade. |

If your managed storage configuration requires the EMC SMI-S Provider, this requires an additional upgrade step.

The following migration scenario is supported to upgrade from StorMan V5.0 to StorMan V6.0. The steps have to be performed according to the specific platform:

1. Create a backup of your StorMan repository:

   ```
   stormandb -dump
   ```

2. Make sure your application is not using StorMan during the upgrade.

3. Stop the StorMan Server:

   ```
   storman -stop
   ```

4. Only if you are using EMC SMI-S Provider:
   Upgrade EMC SMI-S Provider to version V4.6 (according to the EMC Release Notes)

5. Start StorMan upgrade installation for StorMan V6.0

6. Check if StorMan V6.0 was successfully updated and is running:

   ```
   storcheck
   ```

7. Check the correct settings in the `smserver.ini` file after successful upgrade.

If you are using EMC SMI-S Provider an additional step is necessary:

8. Only if you are using EMC SMI-S Provider:
   Check if EMC SMI-S Provider V4.6 was successfully updated and is running:

   ```
   storcfg cimom -show
   ```

7 The graphical user interface – StorMan WebUI

This chapter describes how to start and work with graphical user interface (WebUI).

7.1 Starting the WebUI

StorMan WebUI in StorMan V6.0 replaces the Java based StorMan GUI of V5.0. All existing functions of StorMan V5.0 GUI are ported to V6.0. Additional functions are provided for Storage Management of SE and SEM integration.

StorMan WebUI can be used in

● 'stand-alone' mode for StorMan in Windows / Linux environments

● For SE-Server on M2000 integrated in SEM.
  On M2000 StorMan WebUI supports no 'stand-alone' mode. StorMan does not check this.

● In SQ210 environments with MARS integration the StorMan WebUI is linked in SQ Manager instead of the Java StorMan GUI

7.1.1 Stand-alone Mode

The WebUI can be started as follows:

• Start your browser.

• Enter the following URL:

  `http://<hostname>/StorMan/index.php`

• The StorMan Login window opens next asking to enter or confirm the server name or IP address and the port number of the StorMan Server. The default port number is 4178.

  The Server Name specifies a host system running StorMan Service. The port name and server are saved in the file `smConf.ini` in the USERPROFILE directory.

  A userid and a password can be specified if the access to the StorMan Server is restricted to specific StorMan users. The userid is a StorMan specific user identification defined by WebUI or CLI (refer to section "StorMan user management tasks").

• Click *OK* to confirm the settings. Then the WebUI main window appears next (see section "Elements of StorMan WebUI").

7.1.2 In SE Manager

On SE Server StorMan WebUI is integrated in the SE Manager as an Add-on Pack.

Preconditions:

● Add-on Pack name is STORMAN

● Supported Browsers:

  – Internet Explorer as of V10 (default)

  – Firefox as of V17

The Session Management is controlled by the calling SE Manager. This means:

- No extra authorization (Login) when calling StorMan.

- Main navigation is done by SE Manager → StorMan WebUI is called by link:

  To change from SE Manager to the StorMan WebUI click the **Storage Manager** tab in the menu **Hardware -> Storage**.

  – From every single screen, there is a 'return to SE Manager' function/button to the calling SE Manager screen.

  – Session validation is done at each klick, the session timer is reset.

  – In case of timeout detection during session validation a redirection to the SE Manager Log in screen is done.

  – The Log out function in the Header returns to the SE Manager Log in screen.

  – The 'Change Language' function is NOT provided in SE Manager environment, as in SE Manager environment the language is only set by SE Manager.
  StorMan uses the language setting defined for the session language specific output.
  The 'Change Language' function is provided in 'stand-alone' mode.

  – The Navigation is able to expand/collapse.

  – The StorMan Help System is automatically installed in SE Manager.

Example:

7.1.3 In SQ Manager

On SQ Server the Start button for the StorMan WebUI is integrated in the SQ Manager.

To call the StorMan WebUI click the **Start Storage Manager** button in the menu **MARS -> Storage**.

The StorMan Connection window opens next asking for authorization as described in section "Stand-alone Mode".

7.2 Elements of StorMan WebUI

In stand-alone mode the main window of the WebUI appears in a new window after you have started the WebUI and entered the server name and port. The window's title bar informs about the application and the current StorMan server.

The main window contains a header with product information and the **Help** button. After successful login the header additionally displays the current user name and the **Logout** button.

i	If no StorMan user is defined on the StorMan server the login is only possible without **Account** and **Password** (the fields must not contain a value).

Following a successful login, the StorMan WebUI's main window opens. In the event of an input error, you can correct the input data and repeat the login.
Possible errors:

● An invalid account or an invalid password cause a login error.

● An invalid **Server** or **Port** causes a communication error.

If the WebUI is called from SE Manager there is no login necessary. The WebUI appears in the SE Manager window.

 If the WebUI is called from SQ Manager refer to section "Stand-alone Mode".

7.2.1 Main window

The main window of StorMan opens in a new window as soon as you have logged in.

The window's title bar informs you of the application and the current StorMan server.

The header area contains general product information, language options, help, and the logout.

The following elements are contained below the header area:

● The **tree structure** on the left enables an object to be selected which is to be displayed in the work area.
● The **work area** on the right-hand side displays the data on the selected object and permits actions to be performed.

7.2.2 Navigation

The navigation in the Storage Manager consists of the following main menus:

- ⊖ **SE Manager** (exists only if the call took place via the SE Manager)

- **Storage**

- **Server**

- **Configuration**

- **Authorization**

The main menus **Storage**, **Server**, and **Configuration** can be expanded.

When you click a main menu, the tree structure beneath it expands. Below this you see objects and functions as links. Navigation using the main menus is also referred to as **primary navigation**.

When you click a link, a tab opens in the working area which enables you to manage or operate the object or function Some functions are distributed over more than one tab, and these are displayed at the top of the working area. These tabs are also referred to as **secondary navigation**.

A main menu expands in the following cases:

- When you click the main menu again.

- When you click a link in another main menu.

Hiding or displaying the tree structure

You have the option of hiding or displaying the tree structure:

The toggle icon ▤ in the information bar hides or displays the tree structure.

7.2.3 Work area

The work area shows object tables in accordance with the object which is selected in the main menu's tree structure.

The main menu **Authorization** does not contain a tree structure to select objects.

7.2.3.1 Tab

When a pool or volume is selected in the main menu Storage or when a server or an HBA port is selected, the work area is split into multiple tabs to display different object tables:

- The **Overview** tab displays an information overview.

- The **Connections** tab displays connections of volumes.

- The **Replications** tab displays the volumes for which volume replication is defined.

7.2.3.2 Object table

The topmost object table displays the properties of the selected object.

Optionally one or more tables can display additional information on the selected object and the objects subordinate to it.

7.2.4 Task areas

StorMan offers functions for the following main tasks:
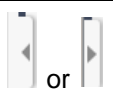
- Setting up and configuring StorMan in the main menu **Configuration**

- Managing storage resources in the main menu **Storage**

- Managing server and storage resource resources in the main menu **Server**

- Managing assignment of storage resources in the **Connections** tab from the storage view in the main menu **Storage** or from the server view in the main menu **Server**

- Managing volume replication in the **Replication** tab from the storage view in the main menu **Storage** or from the server view in the main menu **Server**

- Managing StorMan users and authorizations in the main menu **Authorization**

7.2.5 Using Help

The StorMan help starts in a separate browser window when clicking the **Help** button or **F1** key in the WebUI window. If the help is already started only the existing help window is activated.

The help window displays on the left-hand side the navigation area and on the right-hand side the help topic depending on the selected object in the WebUI at calling time.

Icons on the top

◀ or ▶	hides or shows the navigation area
🖨	prints the displayed topic
←	scrolls back one page in the topic history
→	scrolls forward one page in the topic history
≡	expands hidden text in the displayed topic
≡	collapses hidden text which was expanded in the displayed topic
✎ₓ	removes the highlighting of search terms

Help navigation

Navigation by TOC is selected by default. The table of contents displays in a tree view the hierarchy of all help topics. Selecting a TOC entry displays the concerning topic. Topics which contain subordinate topics will appear as book.
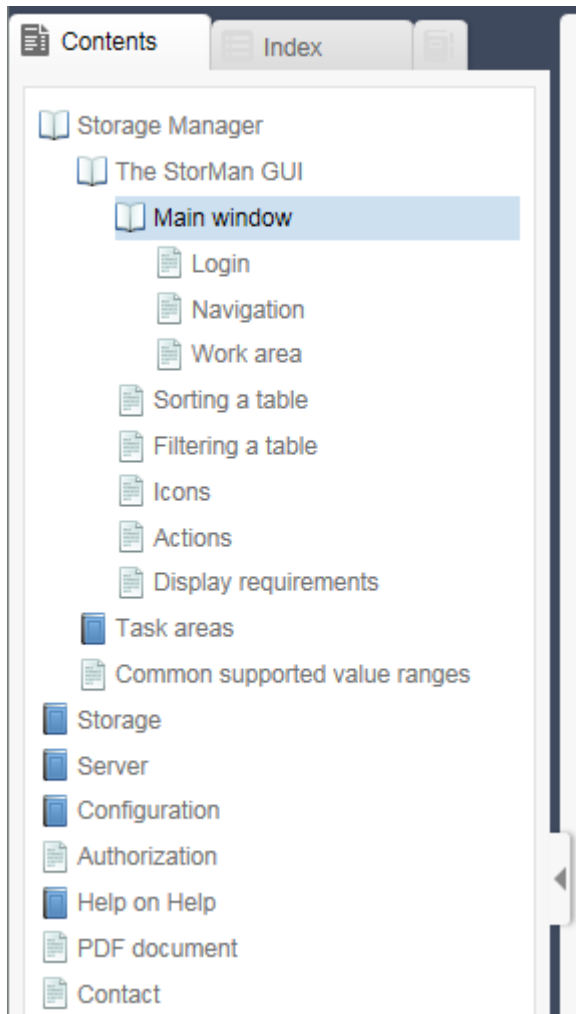
Clicking such a topic

- "opens" the book and shows the contained topics on the tree
- or "closes" the book and hides the contained topics.

To change to navigation by Index or Glossary select the concerning tab in the navigation area.

The top of each topic displays the topic hierarchy (path from main topic to displayed topic).

Clicking to an upper topic on this path displays the concerning topic.

For a detailed description of the WebUI functions please refer to the StorMan online help.

8 Setting up StorMan configurations

8.1 Overview

Configuration features are supported by the StorMan WebUI and CLI. The basic functions are available on both interfaces, but some functions are only available on the CLI. Functions for trouble-shooting are only available on the CLI (refer to section "Diagnostic aids").

8.2 Setting up StorMan configurations using the WebUI

Use the **Configure** tab of the WebUI to set up StorMan. See chapter "The graphical user interface – StorMan GUI" on basic information on the WebUI.

The **Configure** tab reflects the view of the storage configuration rooted by a Management Instance. The tree view linked to a Management Instance reflects all storage arrays that can be managed by the respective CIM Host.

Following functions are available:

- Add a new Management Instance

- Remove Management Instance

- Display/modify Management Instance properties (implicit function)

- Add a new pool

- Delete a pool

- Rename a pool

- Display/modify pool properties (implicit function)

- Moving devices into pools

- Display/modify device properties (implicit function)

- Remove a storage system

The StorMan help system contains the complete WebUI functions reference. Refer to section "Using Help".

8.3 Setting up StorMan configurations using the CLI

This section is a description based on the StorMan CLI.

For details about the CLI commands, refer to chapter "StorMan for programmers – CLI description".

8.3.1 Specify one or more CIM Servers / Management Instances

CIM Servers hosting SMI-S Providers in proxy mode and storage systems with embedded SMI-S Providers must be defined in StorMan's configuration to allow the communication between StorMan and SMI-S Provider.

Execute the command

```
storcfg cimom -add -name myCimHost [-cimuser myCimUser -cimpwd myCimPassword]
```

to specify the name of the host running the CIM Server. If you have specified a CIM User (see section "CIM-OM Settings"), you have to specify it as part of this command as described in section "storcfg". A discovery of the storage systems by the CIM Server is started automatically.

For Eternus DX storage arrays with embedded SMI-S Providers, the CIM Server name or IP address corresponds with the storage system.

For Symmetrix VMAX storage arrays, the CIM Server is embedded on the Service Processor of the storage system. Only https access is supported.

For storage arrays from NetApp, the name of the NetApp FAS system is specified as API Server, because the used ManageONTAP functions reside on the filer. Execute the command

```
storcfg cimom -add -name myNetApp -interface netappapi
```

to specify the name of the NetApp FAS filer and the interface type as `netappapi` instead of a CIM Server.

8.3.2 (Re)Discover storage systems

Execute the command

```
storcfg cimom -discover -name myCimHost
```

This starts a discovery of the storage systems by the CIM Server. A complete discovery of the storage systems and its configuration by the CIM Server is required to update the configuration data of the CIM Server, e.g. in case a new storage system has been connected. This command can take some minutes depending on how many storage systems are connected and how large they are (from 15 to 60 seconds per storage system).

Verify the results with

```
storcfg system -show
```

For CLARiiON CX systems without an FC connection use the `storemc` command as described in section "storemc" to make the CLARiiON CX visible for the SMI-S Provider before executing the `storcfg cimom -discover` command.

8.3.3 Create StorMan pools

If the storage volumes you want to provide for your servers are already created (bound) in the storage system, you have to ascertain the logical device numbers of the volumes to be used. If not, you can create new devices by using a StorMan pool which allows dynamic creation of new devices in the storage system. To ascertain appropriate devices that are already created, you can use the StorMan WebUI or the storage system's administration software or ask your storage system administrator. A pool is created using:

```
storcfg pool -add -poolid myPool -system system [-restype dynamic]
```

If you want to create new devices in the storage system for that pool, use the parameter `-restype dynamic`.

Pools can be shown using

```
storcfg pool -show
```

8.3.4 Using existing storage volumes

To add devices to your pool, ask the administrator of the storage system for unused logical devices (LUNs). Then use

```
storcfg volume -add -poolid myPool -system system
       -deviceid 0075[,0076,...] [-storid myStorID][,myStorID2]
```

(assuming that 0075 is the device number / device ID of one of your logical volumes).

Repeat this command for each volume or specify a list. If storIDs are specified, they are assigned to the volume and can be used for further `attach` / `detach` commands. If no storIDs are specified, you have to use `storcreate` to reserve these devices and assign storID to it before they can be attached by storattach.

If this device has LUN masking connections to one or more hosts when it is added to the pool, StorMan automatically assigns a storID to the volume because it is already `in-use`. In this case it will not be used for automatic storage reservation by `storcreate`.

8.3.5 Creating new storage volumes

If the pool has the property `-restype dynamic`, the `storcreate` function creates a new volume in the array if no free volume with the requested size and RAID type is already contained in the pool.

```
storcreate -storID myStorID -poolid myPool -size size-in-MB -raidType RAID1
```

It is also possible to specify no pool, but an array, or a model, or a RAID type, or a size. In these cases StorMan looks for an appropriate pool. See the CLI reference for details. Do not create new volumes without permission from the administrator of the array. RAID groups with enough space left have to be available when new devices are created.

8.3.6 Discover changes to a storage array

StorMan is able to detect configuration changes in the storage system automatically by means of the CIM-OM. However a discover process can be triggered by StorMan, e.g.:

- For one storage system use

```
storcfg system -discover -system system
```

- For all storage systems visible to a CIM Server use

```
storcfg cimom -discover -name hostname-of-the-cim-server -discover deep
```

8.3.7 Remove a storage system

A storage array that is no longer managed by StorMan should be removed from StorMan configuration.

- To remove one storage system use:

  ```
  storcfg system –rem –system system
  ```

- In case of an EMC storage system you should also remove it from the EMC SMI-S Provider's repository:

  ```
  storemc –rem ... –system system
  ```

- In case the removed storage system is the only visible to a specific CIM Server you should also remove the CIM server to avoid automatic re-discovery of the storage system:

  ```
  storcfg cimom –rem –name hostname-of-the-cim-server
  ```

8.4 Setting up StorMan configurations on M2000

StorMan configurations on M2000 can be delivered at least partly preconfigured. However all activities to setup or modify the StorMan configuration are available for the storage administrator via WebUI, that is completely integrated into SEM. On M2000 the complete functionality for StorMan is available as for any other platform.

To set up or modify StorMan configurations on M2000 using the StorMan WebUI please refer to chapter 8.2.

To setup or modify StorMan configurations on M2000 based on the StorMan CLI please refer to chapter 8.3. On M2000 the StorMan CLI is reserved for the service personnel only. For details about the CLI commands, refer to chapter "StorMan for programmers – CLI description".

On M2000 there are some additional functions available to maintain the StorMan configuration. The current chapter handles configuration functions available for M2000 only.

8.4.1 Configure Server Units in StorMan repository

The StorMan repository contains the server names and IP addresses of all Server Units of the SE-Server managed by SEM. The necessary information is obtained automatically and added to the StorMan repository by internal use of

```
storadmin configsrv
```

This is automatically processed during installation on M2000 and is available to the service to update after configuration changes of the SE Server Units.

8.4.2 Configure Tape Storage for information and monitoring

By WebUI You can register information in StorMan from Storage systems which StorMan cannot manage or monitor. This enables StorMan to display these storage.

8.5 Setting up StorMan configurations on MARS

StorMan configurations on MARS can be delivered at least partly preconfigured. However all activities to setup or modify the StorMan configuration are available for the storage administrator. On MARS the same functionality for StorMan is available as for any other platform.

To set up or modify StorMan configurations on MARS using the StorMan WebUI please refer to chapter 8.2.

To setup or modify StorMan configurations on MARS based on the StorMan CLI please refer to chapter 8.3. For details about the CLI commands, refer to chapter "StorMan for programmers – CLI description".

On MARS there are some additional functions available to maintain the StorMan configuration. The current chapter handles configuration functions available for MARS only.

8.5.1 Specific StorMan functions on MARS

The following functions are performed on MARS, automatically as part of the StorMan installation.

The specific StorMan functions available on MARS are provided by CLI `storadmin` and its specific functions. These functions are available for administrator and service accounts on MARS.

8.5.1.1 Configure StorMan repository

The StorMan repository must contain the server names and IP addresses of all SQ Server Units and Application Units managed by MARS. The necessary host information can be obtained and added to the StorMan repository by using `storadmin configsrv`:

```
storadmin configsrv
```

This is automatically processed during installation on MARS and may be helpful to configure changes of the SQ Server Units.

8.6 Diagnostic aids

8.6.1 StorMan

StorMan provides a number of files containing important information for troubleshooting and for diagnosing errors.

To ensure the best possible support when problems are encountered while StorMan is being used, you are recommended to recreate the problem which occurred if this is possible after you have set the highest debug level on the StorMan server for the duration of recreation process using the StorMan CLI:

```
storparam –debug 5
```

More detailed logging information you may get using the parameters `–debugcom` and `–traceworker` of the StorMan CLI `storparam`.

Please use the StorMan CLI to provide the current configuration data:

```
stormandb –dump
```

Afterwards, collect the following files and transmit them to your support organization:

1. The StorMan logging files with format `stormanTraceyyyy-mm-dd` in the `./log` directory for the day(s) concerned.

2. The configuration file of the StorMan server smserver.ini in the `./config` directory.

3. The StorMan database StorMan.db in the `./repository` directory. It can be linked to another directory or file (e.g. to a filer -> `/filer/StorMan/repository/StorMan.db`).

4. The saved StorMan database files `AddData.sql` and `AddTables.sql` (created by `stormandb –dump`) in the `./backup` directory.

These files are located in the `/var/opt/SMAWstor/StorMan` directory for Linux environments and in the `C:\Programme\Fujitsu\StorMan` directory for Windows environments.

Additional support for Linux (including M2000 and MARS)

On Linux platforms and the service account (for M2000 and MARS) you may also get all diagnostic information as a g-zipped tar archive by using the StorMan CLI

```
storadmin getlogs
```

The resulting file is located at:

- `storManLogs.tar.gz` for Linux

- `/tmp/storManLogs.tar.gz` for M2000 and MARS.

To remove all log files use

```
storadmin remlogs
```

8.6.2 SMI-S Provider

Please refer to the Release Notes of the specific vendor for general diagnostic information regarding the support of SMI-S Providers.

In some cases the SMI-S Provider might not be aware of configuration changes on the storage array initiated by other management tools or instances (e.g. if an unexpected error `STORMAN_FAULT` or `STORMAN_CIMOM_ERROR` is returned as result of a storattach or storcreate function).

In this situation you are recommended to initiate an update of the internal SMI-S Provider repository (only in the case of Proxy configurations) using

```
storcfg system -discover
```

or

```
storcfg cimom -discover
```

and retry the failed function.

9 StorMan user management tasks

The administration of StorMan users and authorizations is supported by CLI and WebUI. Currently all authorizations to access StorMan are controlled by StorMan internal user and password combinations.

> **i** If no StorMan user is defined the complete StorMan functions are available for all connecting StorMan servers. No authorization checks are done.

The following section is a description based on the StorMan CLI and shows the actions to administrate StorMan users.

The corresponding WebUI functions are described in the WebUI's help system (refer to section "Using Help").

9.1 Define a new user and his authorization

Define a new user, its password protection and StorMan role in StorMan Repository by using

```
storauth user –add –name user –passwd pwd –role role
```

Defining the first StorMan user activates the authorization checks for all connecting StorMan servers. All subsequent calls require the specification of `user` and `passwd`.

9.2 Modify the password of an existing StorMan user

Modify the password of a user already contained in the StorMan repository by using

```
storauth user –mod –name user –newpasswd newpwd –user user –pwd pwd
```

9.3 Remove a StorMan user

Remove a StorMan user from the StorMan administration by using

```
storauth user –rem –name user –user user –pwd pwd
```

> **i** After removing the last user the complete StorMan functions are available again for all connecting StorMan servers. No more authorization checks are done.

9.4 Display information related to StorMan users

Display information related to all StorMan users by using

```
storauth user –show [–user user –pwd pwd]
```

Displays information related to a particular StorMan user by using

```
storauth user –show –name user [–user user –pwd pwd]
```

9.5 Assign new StorMan Role to StorMan user

Assign a new StorMan role to an StorMan user using

```
storauth user –mod –name user –newrole role
```

9.6 Assign new list of pool ID's to StorMan user

Assign a new list of pool ID's to an StorMan user with the *PoolAdmin* role using

```
storauth user –mod –name user –newpoolids poolid[,...]
```

10 Administration and Provisioning tasks

10.1 Overview

Using the provisioning features of StorMan hosts can be provided with necessary storage resources managed by StorMan. All hosts which are allowed to request storage resources are managed in the StorMan database (repository). With the provisioning features a host can be added to the StorMan repository and storage resources be attached to or detached from this host. The provisioning of a host will be ended by removing it from the repository.

Provisioning features are supported by the StorMan WebUI and CLI. The basic functions are available on both interfaces, but some functions are only available on the CLI.

10.2 Administration and Provisioning tasks using the WebUI

The **Provisioning** tab in the main window provides the provisioning features of StorMan on the WebUI.

Hosts that should be provisioned by StorMan have to be added to the StorMan database using the Server tree.

Following functions are available:

- Add a new host
- Remove a host
- Add a new WWPN
- Remove a WWPN
- Display devices available for provisioning (implicit function)
- Attach a device or a selection of devices
- Detach a device or a selection of devices
- Display attached devices (implicit function)

The StorMan help system contains the complete WebUI functions reference. Refer to section "Using Help".

10.3 Administration and Provisioning tasks using the CLI

This section is a description based on the StorMan CLI. If you prefer to use the WebUI for configuration, see chapter "Administration and Provisioning tasks". For details about the CLI commands, refer to chapter "StorMan for programmers – CLI description".

10.3.1 Add a new host

Add the hosts that are to be provided with storage units (LUNs) via StorMan using the `storcfg host` command, e.g.:

```
storcfg host -add -name myApplicationHost -ip ip-adr1,ip-adr2,...
              -wwpn wwpn1,wwpn2,...
```

StorMan needs the Host WWPNs to attach / detach logical devices (LUNS) using the LUN masking functions.

WWPNs can be specified in the format `210000C09F9568F8` or `21:00:00:C0:9F:95:68:F8`.

The new entries can be checked using the `storcfg host -show` command.

A host can be removed from the StorMan repository using

```
storcfg host -rem -name myApplicationHost.
```

Hosts can also be specified just by a WWPN (see section "storcfg host" for details). However, if the hosts have more than one HBA, you are recommended to specify a name and / or IP address for each host to provide a better general overview.

> HBA WWPNs can be determined using the software for administering the FC switch or software
>
> supplied by the HBA vendor. They also should be printed on the HBA board.

10.3.2 Attach a logical device to a host

The devices contained in the pool(s) can be attached to any host in the StorMan administration (repository).

- If all HBAs of that host should have access to the device, use

  ```
  storattach -storid myStorID -hostname myApplicationHost
  ```

- If only one HBA should have access to the device, use

  ```
  storattach -storid myStorID -hostwwpn WWPN-of-one-HBA
  ```

10.3.3 View the attachments to a device

Use the command

```
storcfg volume -show -storid myStorID -showconnections
```

All FC pathes and iSCSI pathes between the host and the device are shown. Each path is a combination of InitiatorID, TargetID and SCSI host device number (LUN). The InitiatorID is the identifier of the connection on the host side and can be a WWPN (for FC connections) or an IQN (for iSCSI connections). The targetID is the identifier of the connection on the side of the storage system and can be the WWPN of the storage port (for FC connections) or the IQN of the storage-port for iSCSI connections).

⚠️ **Volume Mapping to storage ports:**
Different storage systems follow different strategies to provider LUN mapping and LUN masking Functions.

On EMC Symmetrix storage systems, volumes have to be mapped to the storage ports first before they are accessible to connected servers. The mapping is independent of the LUN masking settings for the storage port. LUN masking can be enabled on storage port level and LUN masking requires volumes already mapped to the specific storage port(s).

ETERNUS DX storage systems can be configured either for LUN masking or for LUN mapping on storage port level, see chapter: "Special Properties of Fujitsu ETERNUS DX storage systems" – "LUN mapping and LUN masking".

StorMan information functions are displaying host connections assigned by LUN masking only. This is reflected by the output storcfg volume –show by the additional property called "*HostAccess*" (see example in chapter "StorMan for programmers – CLI description" – "storcfg volume".

- The property 'HostAccess' has the value "Restricted" if LUN masking is active for the connected storage port and the volume is visible only via lun-masking.

- The property 'HostAccess' has the value "AllConnected", if the volume is mapped to a storage port without LUN masking enabled. In result every host connected to this storage port can access the volume.

Refer also to chapter "Special Properties of Fujitsu ETERNUS DX storage systems" – "LUN mapping and LUN masking"

10.3.4 Detach a logical device from a host

Use the command

```
stordetach -storid myStorID -hostname myApplicationHost
```

to discontinue the host access to a specific logical device.

10.3.5 Release a storage volume

Use the command

```
stordelete -storID myStorID
```

The device is released and can be used by the `storcreate` function for new reservations. If the pool has the property `STORMAN_RESOURCE_TYPE_DYNAMIC` and the device was dynamically created by StorMan, it will also be removed (unbound) in the storage system also.

11 Replication tasks

The Replication Service features support the management of mirroring functions:

● local mirroring based on full volume copies (clones)

● local mirroring based on pointer based technologies (snaps)

● remote mirroring based on full volume copies

Information and active management of the Replication Service are supported by CLI and WebUI.

The following section is a description based on the StorMan CLI and shows some typical actions to administrate volume level replications from the original to a snap or clone replication.

> **i** Please note that not all functions are supported for all storage system according to their availability by the storage vendor or model specific implementation.

The corresponding WebUI functions are described in the WebUI's help system (refer to section "Using Help").

11.1 Working with full local mirrors (clones)

11.1.1 Creating clone pairs

To start a clone session for local mirroring or for migration purpose a clone pair is created. After starting the session, the synchronization process from source volume to target volume is started. The clone volume is no longer accessible to host.

```
stormirror -create -mirrortype full -source storid -target storid -waitforsync
```

11.1.2 Suspending clone volumes

To use a clone volume by a second host or application independently, the clone unit has to be suspended.

```
stormirror -suspend -source storid -target storid
```

The consistency of data at the point in time of activation must be assured by the application:

11.1.3 Restarting clone pairs

After finishing independent processing of the target volume the clone pair can be restarted. By restating all modifications done on the target volume are discarded and the changes on the source volume are copied to the target volume. The target volume is no longer accessible by host.

```
stormirror -restart -source storid -target storid
```

11.1.4 Swapping the source volume – clone volume attributes

The attributes of original and clone volumes can be exchanged. This can be done by using the -swap option. As result the former original volume becomes the new clone volume and the former clone volume becomes the original volume. The mirroring relationship of the pair is kept but the mirroring direction is inverted.

```
stormirror -swap -source storid -target storid
```

11.1.5 Terminating a clone pair

A clone pair can be terminated to stop the mirroring relationship between source volume and target volume. Both volumes can be used independently further on.

```
stormirror -terminate -source storid -target storid
```

11.1.6 Information about clone pairs

The command

```
stormirror -show [ -storid storid ]
```

shows the information about a specific volume specified by its storid and all its mirroring related information.

11.2 Working with snapshots

11.2.1 Creating snapshot pairs

To start a snap session a snap pair is created by assigning a snap volume to a source volume. For ETERNUS DX systems the snapshot is directly activated and made accessible to the host at creation time. Afterwards a host or application may use the snap volume independent to the source volume.

```
stormirror -create -mirrortype snap -source storid -target snapstorid
```

11.2.2 Restoring the snap volume from the clone

In case that all changes on the source volume should be discarded, the snap volume can be restored to the source volume. The snap volume remains accessible by host.

```
stormirror -restore  -source storid -target storid
```

11.2.3 Terminating a snap pair

A snap pair can be terminated to stop the snap session between source volume and snapshot. Both volumes can be used independently further on.

```
stormirror -terminate  -source storid -target storid
```

11.2.4 Information about snap pairs

The command

```
stormirror -show [ -storid storid ]
```

shows the information about a specific volume specified by its storid and all its mirroring related information.

11.3 Working with full remote mirrors (synchronous and asynchronous)

11.3.1 Creating remote mirror pairs

To start a remote mirror session between two connected storage systems of the same model a remote mirror pair is created. After starting the session the synchronization from source volume to target volume is started. The remote target volume is no longer accessible to host.

```
stormirror -create -mirrortype full -replica remote -source storid
      -target storid -waitforsync
```

11.3.2 Suspending remote target volumes

To use a remote target volume by a second host or application independent to the source volume the remote mirror has to be suspended.

```
stormirror -suspend -source storid -target storid
```

The consistency of data at the point in time of activation must be assured by the application:

11.3.3 Restarting remote mirror pairs

After finishing independent processing of the remote target volume the remote mirror pair can be restarted. By restarting all modifications done on the target volume are discarded and the changes on the source volume are resynchronized to the target volume. The target volume is no longer accessible by host.

```
stormirror -restart -source storid -target storid
```

11.3.4 Swapping the source volume – target volume attributes

In case that the attributes of source and target volumes should be changed, this can be done with the –swap option. As result the former source volume becomes the new target volume and the former target volume becomes the source volume. The remote mirroring relationship of the pair is kept but the mirroring direction is inverted.

```
stormirror -swap -source storid -target storid
```

11.3.5 Terminating a remote mirror pair

A remote mirror pair can be terminated. As a result the mirroring relationship between source volume and target volume is stopped. Both volumes can be used independently further on.

```
stormirror -terminate -source storid -target storid
```

11.3.6 Information about remote mirror pairs

The command

```
stormirror -show [ -storid storid ]
```

shows the information about a specific volume specified by its storid and all its mirroring related information.

12 Statistics and performance tasks

The statistic and performance features supported are suitable for monitoring of storage arrays.

The statistic and performance monitoring features are provided on volume (LUN) level, namely the following:

- reads per second

- writes per second

- MB read per second

- MB write per second

- Response times for read and write IOs

- Cach Hit Rates for read and write IOs

- Average IO times for reads and writes

These features are available only for internal use (in particular openSM2).

12.1 Evaluation

As StorMan's `storstat` interface is providing counters and sampled metrics in conjunction with time stamp representing the absolute collection time for the statistics. The calling application has to calculate and normalize the values to obtain the metrics per second.
This can be achieved by periodical calls in time intervals (t_n, t_m) calculating the rates for the delta = t_m - t_n.

1. To calculate the IO and MB values per second please use the following method:

$$\texttt{ReadIOs per second} = \frac{\texttt{delta(ReadIOs)}}{\texttt{delta(statisticTime)[s]}}$$

2. To calculate the average read (or write) times for ETERNUS DX and Symmetrix please use the following method:

$$AverageReadTime\,[\mu s] \;=\; \frac{delta(SampledReadsTime)}{delta(SampledReadsCounter)}$$

$$AverageWriteTime\,[\mu s] \;=\; \frac{delta(SampledWriteTime)}{delta(SampledWriteCounter)}$$

| i | An evaluation tool has to assure to get statistical data at least once during one sample interval (default values: 3 min for ETERNUS DX, 5 min for Symmetrix). |

12.2 Getting statistics

The `storstat` command delivers statistical data (refer to "storstat").

Statistical data for volumes

► Execute the command

```
storstat volume -get -system *308
```

to get the statistic data for all volumes of the storage system with serial number *308 (wildcard specification).

13 Provisioning tasks: Special properties of storage systems

13.1 Special Properties of Fujitsu ETERNUS DX storage systems

13.1.1 LUN mapping and LUN masking

StorMan V6.0 supports LUN masking for ETERNUS DX410/DX440 S2 and DX8700 S2 and DX500 S3 and DX600 S3 models.

On ETERNUS DX systems the access from a Server HBA to a specific volume via a FC target port is controlled by LUN mapping and LUN masking settings. The LUN masking is realized based on the so called "*AffinityMode*". The *AffinityMode* can be enabled or disabled for every single FC target port.

Access to the Volumes is controlled by LUN mapping when *AffinityMode* is disabled. Storage volumes are mapped to an amount of target ports and every initiator port that is connected to this target port has access to these volumes.

Access to the Volumes is controlled by LUN masking when *AffinityMode* is enabled. Affinity groups can be created. An Affinity group contains an amount of storage volumes and can be connected (allocated) to each Initiator port (HBA) via one or more target ports on the ETERNUS DX system.

The LUN masking feature of ETERNUS DX systems is using Affinity Groups (AG). An AG is an amount of logical volumes that have a SCSI host device number. By connecting the AG with one or more Host initiators (Port WWPNs) via one or more target ports, the volumes become accessible for the host initiators. See figure bellows:
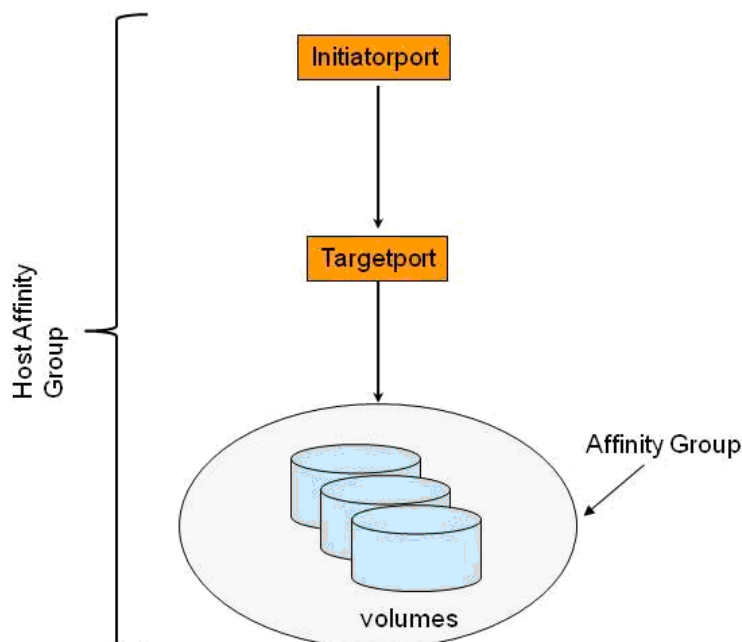
Figure 3: Elements of LUN masking in ETERNUS DX systems

StorMan information and management functions support the AffinityMode only, i.e. StorMan shows host device connections consisting of "`Initiatorid - targetid - hostlun`" only for volumes contained in Affinity groups. Please refer to chapter "View the attachments to a device".

The "*attach*" (storattach) and "*detach*" (stordetach) functions are supported for target ports with enabled AffinityMode.

Before using StorMan for LUN masking with ETERNUS DX, consider the following:

- The FC connections and zoning should be set up already.

- The specific settings for the frontend ports and the Host Response settings for every Initiator (HBA) should be checked and modified by using the WebUI or CLI of the storage system. See the system specific documentations on *http://manuals.ts.fujitsu.com* and *http://storage-system.fujitsu.com*.

However, the LUN masking can be performed even before these settings have been done, based on the planned FC connections between the host side iniator ports (wwpns) and the target ports (target wwpns) on the ETERNUS DX.

13.1.1.1 LUN masking handling by StorMan

For `storattach` calls, StorMan first checks the connections for each initiator port:

- if an initiator port is already connected to an Affinity Group, the specified volumes are added to this AG assigning the Host LUN numbers specified by caller. If no Host LUN is specified the next free Host LUN numbers are assigned automatically.

- If no Affinity Group exists for an initiator port, StorMan creates a new Affinity Group for each initiator-target-pair.

StorMan is not aware of the FC connection between initiator port and target port. Therefore for the very first attach of volumes to a server both, initiator port and target port wwpns have to be specified using the parameters "-*hostwwpn*" and "-*targetwwpn*". The parameters "-*hostwwpn*" and "-*targetwwpn*" are supported for ETERNUS DX systems only.

If a list of targetwwpns is specified, the corresponding list of hostwwpns has to be specified in the same sequence for the storattach call. Each element of the hostwwpn list will be assigned to the corresponding element of the targetwwpn list. This is the way to define multiple FC-connections between servers and ETERNUS DX storage systems.

`stordetach` calls are processed as follows:

- for each initiator port defined by the specified parameter –hostname or the list of -hostwwpn StorMan determines the existing Affinity Groups and removes the specified volumes from the Affinity Group.

- a list of targetwwpns can be specified only if the initiator ports are defined by a list of hostwwpns. StorMan determines the AffinityGroups defined by the sequence of the 2 lists (initiator target pairs) and removes the volumes from the respective Affinity Groups. This proceeding is especially useful when e.g. HBA's (initiator port) have been physically removed from a host and other connections remain unchanged.

If an Affinity Group becomes empty, because the last volume was removed by a stordetach call, the empty AG itself is removed automatically. Therefore, the parameter "-destroyconnection" is not required and not supported for ETERNUS DX systems.

13.1.1.2 Maximum number of volumes/ Host-LUNs

StorMan supports up to 256 Volumes/ Host-ILuns in one storattach/ stordetach command.

ETERNUS DX supports up to 4096 Host-LUNS in one Affinity Group depending on the settings in the Host Response (e.g. preset Host Response 'BS2000').

As of ETERNUS DX S3 this feature is also supported by StorMan storattach/ stordetach command, i.e. up to 4096 volumes can be added to one Affinity Group (by storattach).

13.1.1.3 LUN masking with ETERNUS DX S2 / S3 WebUI

For ETERNUS DX S2 systems the LUN masking for the same server should not be performed by using both tools, the ETERNUS DX S2 WebUI and StorMan. The ETERNUS DX S2 WebUI uses Hostgroups and Portgroups which are not compatible with the LUN masking as performed by SMI-S.

> ⚠️ WARNING!
>
> Especially when a LUN group for more than one Server has been setup with the ETERNUS WebUI, you should not use the StorMan `storattach` or `stordetach` commands for these server because they will always affect all of these servers!
> If you plan to use the StorMan LUN Masking functionality for these Servers, the LUN groups should be deleted.

13.1.2 Dynamic creation of storage volumes / logical devices

If required by the `storcreate` function, StorMan checks existing storage pools (RAID groups) with the desired RAID type in the storage system. If one with the requested available size is found, a new volume is created in that RAID group. If no RAID Group with the requested size and RAID Type is found, the command returns with `STORMAN_NO_STORAGE_FOUND`. Storman does not create a new RAID group.

13.2 Special properties of EMC Symmetrix storage systems

In order to use LUN masking functions, the masking software has to be installed and activated in the storage system. For Symmetrix systems this is "VolumeLogix".

Details about the storage system-specific LUN masking functionality can be found in the manuals for these products that are available on the manufacturer's website.

On Symmetrix DMX systems a volume configuration management (VCM) database device has to be configured and at least 3 gatekeeper devices must be visible to the host on which the SMI-S Provider is running. The VCM database contains the device (LUN) masking records. These records describe the amount of devices each FC initiator can access via one or more storage ports (in other words the "view" for any initiator to the storage system). Furthermore, LUN masking functionality has to be enabled for the FC target ports in the Symmetrix system to which the application hosts are connected.

For Symmetrix VMAX systems running the SMI-S Provider embedded on the Service Processor no FC connection and no configured gatekeeper devices have to be available to for the SMI-S Provider. Additionally the SMI-S Provider in proxy mode is still supported, and requires FC connectivity and configured gatekeepers on it's hosting server. as described for Symmetrix DMX. For details please refer to chapter "LUN masking in Symmetrix VMAX systems".

Further information about installation and preparation of the SMI-S Provider can be found in the release notes for the SMI-S Provider.

13.2.1 Dynamic creation of storage volumes / logical devices

If required by the `storcreate` function, StorMan checks existing storage pools (RAID groups) with the desired RAID type in the storage system. If one with the requested available size is found, a new volume is created in that RAID group. If no RAID group with the requested size and RAID Type is found, StorMan / SMI-S-Provider tries to create a new one. Please note that the required number of unallocated disks and available disks to create a new RAID group depends on the RAID type.

Creation of storage volumes in a storage system is an asynchronous process that can take a few minutes. Thus, if StorMan is used for automated storage provisioning, you are recommended to configure the amount of required volumes before they are required by any applications using StorMan or the storage system specific administration tools in order to keep the time for provisioning short.

It is recommendable to use the Symmetrix-specific tools to create volumes because Storman creates **one** volume per StorMan request and each request causes a time-consuming reconfiguration.

A Symmetrix volume created dynamically by StorMan is automatically mapped to all FC target ports (LUN mapping) of the Symmetrix. With future releases of StorMan it will be possible to specify the storage ports to which a newly created volume is mapped.

StorMan with SMI-S as of 1.2 supports the following RAID types for EMC storage systems:

RAID Level	Symmetrix	Comments
RAID 0	NO	
RAID 1	YES	
RAID 5	YES	RAID 5 feature has to be activated on Symmetrix systems
RAID 6	YES	RAID 6 feature has to be activated on Symmetrix systems
RAID 1-0	NO	
RAID 3	NO	
RAID-S	YES	
Unprotected (Disk)	YES	

13.2.2 LUN masking in Symmetrix DMX systems

In Symmetrix DMX systems, the Volume Logix software provides LUN masking features. Details about it can be found on the manufacturer's website. Records in the VCM database control the LUN masking. Each of these records defines a relationship between one HBA (represented by its WWPN), one Symmetrix FC Port and a number of logical devices (storage volumes / LUNs). One HBA may have access to a different number of LUNs via different FC target ports (anyhow, StorMan does not provide support for attaching / detaching a device via a specific target portThe entries in the VCM database are called "SCSIProtocolController" in SMI-S terms (abbreviated as "**SPC**"). On Symmetrix systems, entries in the "Login History Table" (LHT) are also represented by SPCs. So if an active HBA is connected to the Symmetrix, this is an SPC that defines a FC connection that is used by StorMan for "attach/detach" requests.

Setting of the host device number during an attach operation is not yet supported for Symmetrix DMX systems.
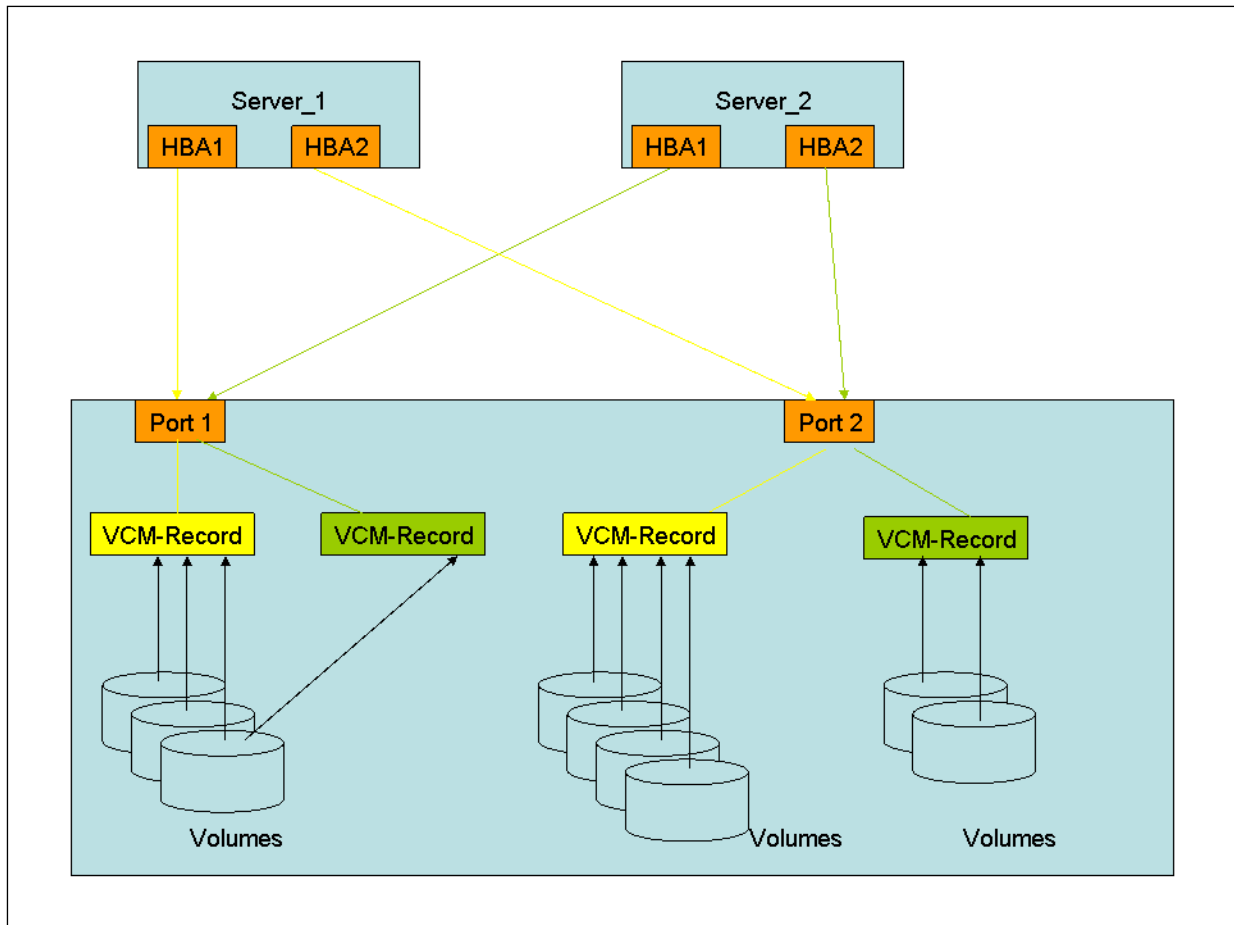
Figure 4: LUN masking in EMC Symmetrix

If a `storattach` function is received, StorMan checks if SPCs already exist for this HBA-WWPN(s). If one or more are found, the volumes are added to them, and become visible for that HBAs via the related Symmetrix target ports. The attach function is refused if the complete amount of volumes is not mapped to those ports! No new SPCs (VCM records) are created in this case.

You cannot specify target ports for an attach request with StorMan. If no VCM records (SPC) exist for HBAs which are to be attached, StorMan first checks the mapping of the specified volumes. If all volumes are mapped to the same amount of target ports, StorMan will create one VCM record for each initiator and each of those target ports. StorMan does not modify the Mapping in Symmetrix systems. Once VCM records exist for a HBA, StorMan will use these and only these records for further attaches, i.e. it is not possible to attach a group of volumes for a WWPN via different groups of Symmetrix ports. Check if the volumes are mapped to the same target ports the host is connected to.

⚠ Note, that on an FC target port without LUN masking enabled, each connected host can access all devices mapped to this FC port. StorMan modifies the LUN masking only and does not modify the device mapping. Therefore devices to be provided by StorMan must already be mapped to the required ports and LUN masking has to be enabled on these ports.

Volumes property hostaccess: If a volume is mapped to FC ports on which LUN masking is not enabled, its property `hostaccess` will have the value `allconnected`. In this case it is accessible by all connected hosts via these ports. Otherwise it has the value restricted and the hostaccess is controlled by lun masking.

Before using StorMan for LUN masking with Symmetrix, consider the following:

- If VCM entries for the hosts (to be provided by StorMan) already exist, the HBAs affected and the target ports have to be physically connected and connected by FC zoning.

- Volumes that are not dynamically created by StorMan must already be mapped to the storage target ports that are accessible from the hosts.

- FC zoning for the hosts and storage systems is already configured and active.

- •StorMan cannot add volumes to VCM entries on which the "dynamic lun addressing" (dla) feature is enabled. In this case the `storattach` function will terminate with the error code `STORMAN_FCT_NOT_POSSIBLE`.

Prerequisites for LUN masking in Symmetrix systems

The FC director port-specific settings on the Symmetrix systems have to be checked or modified by EMC service personnel. These settings can differ for different operating systems.

13.2.3 LUN masking in Symmetrix VMAX systems

13.2.3.1 Storage Provisioning based on 'Autoprovisioning Groups'

LUN masking

The LUN masking for EMC Symmetrix VMAX storage array is implemented based on "Autoprovisioning groups" and is processed in 2 steps:

- Step 1:

  Three types of resource groups are created:

  a. InitiatorGroup (IG): Groups of Server HBA's (Host WWPNs),

  b. TargetPortGroup (PG): Groups of Target Ports

  c. StorageGroup (SG): Groups of Storage Volumes

- Step 2:

  A so called "Masking View" is created. The Masking View consists of ONE group of each type. The Masking View defines the visibility of all elements of the contained groups, i.e. all HBA's of the Initiator Group can access all Volumes of the StorageGroup via the Target Ports of the TargetPortGroup

The usage of Masking Views allows a simple management of LUN masking as "attaching" and "detaching" storage IDs to a server results in adding / removing the LUNs to / from a StorageGroup that is part of the Masking View. Similar adding / removing multiple access paths implies adding / removing Target-Ports to the specific Target Port Group or adding / removing Server HBA's to the specific Initiator Group of the Masking View.

Preconditions:

- A StorageGroup that is part of a Masking View must not be empty. In result removing all Volumes from the StorageGroup is only possible if the related Masking View is also removed.

- Volumes can be part of multiple StorageGroups.

- Target-Ports can be part of multiple TargetPortGroups

- Server HBA's can be part of only one InitiatorGroup. The feature "*InitiatorGroups being part of an other InitiatorGroup*"' is not supported by StorMan

LUN-Mapping

For EMC Symmetrix VMAX storage arrays the mapping can be done implicitly as part of the LUN masking. However LUN masking controls are significantly faster if LUN-Mapping is already done in advance. If the mapping is to be performed as part of masking, make sure, that the SMIS-Provider is configured to do that synchronous, as described in chapter 3.2.3.5.

Assigning Host-LUNs

With EMC Symmetrix VMAX storage arrays assigning a Host-LUN (HLU / SCSI Device number) is no longer dependent on LUN Mappping.

Host-LUN's can be assigned at the time of adding a volume to a StorageGroup or at creation time of a Masking View.

If no Host-LUN is specified they are assigned automatically starting with '0' or the lowest free number. If the Volumes are not yet mapped they are mapped automatically.

13.2.3.2 LUN masking handling by StorMan

storattach

If the Server is already part of a Masking View, StorMan adds the storage IDs to the existing StorageGroup. A new Masking View is created for the specified HBAs /host if not yet existing.

`storattach` supports only ONE Masking View. `storattach` is rejected (STORMAN_BAD_CONFIG) if the specified HBAs / host are already in different InitiatorGroups or the InitiatorGroups are contained in multiple Masking Views. In this case the Masking View has to be removed first by using `stordetach`.

If a part of the specified storage IDs are already contained in the existing Masking View, only the additional storage ID's are added. The storattach call is terminated successful.

For each `storattach` call with one HBA / host a dedicated TargetPortGroup and StorageGroup is created, even if they are already part of a TargetPortGroup or StorageGroup with the same elements. The names generated by StorMan are based on the following scheme:

- "IG_<hostname>"
- "PG_<hostname>"
- "SG_<hostname>" and
- "VIEW_<hostname>".

The Host-LUNs are assigned at:

a) Creation time of the Masking View or

b) When adding volumes to the Storagegroup (SG)

stordetach

With `stordetach` the specified storage IDs (volumes) are removed from the Masking View of the specified host / HBA .

One Volume may be part of multiple Storage Groups and Masking Views for one host. If the last volume is removed from the Masking View and the Storage Group, the Masking View itself and all its groups are removed. This way existing multiple Masking Views for a specific host can be removed by StorMan using `stordetach` command (e.g. in cases when Masking Views have been created by other tools than StorMan).

Differences to Symmetrix DMX

`storattach` for Hosts / HBAs not yet FC connected to the VMAX is not supported. All specified hosts / HBAs have to be FC connected to the VMAX as StorMan's LUN masking is done based on this information.

Adding additional pathes between server and storage after performing a storattach via Target-Ports not yet contained in the TargetPortGroup: The new Target-Ports are not considered when attaching additional Volumes (storage IDs). To attach them there are to options:

- Separate `storattach` calls for the volumes and the new HBAs
- First call `stordetach` for all existing volumes to delete the existing Masking View and afterwards `storattach` to create a new Masking View containing all HBAs.

14 StorMan for programmers – CLI description

This chapter covers all the StorMan features related to StorMan administration and storage provisioning provided on CLI level.

The StorMan CLI is not required for the Information and Monitoring features delivered for server management integration except the function `storcfg cimom` to set up the SMI-S Provider configuration.

14.1 General

14.1.1 Common CLI aspects

Some aspects are valid for all commands of the StorMan CLI. These are specified here and not repeated for each command.

14.1.1.1 Parameter file

The CLI can read additional command parameters from a file. This can be useful especially for connection parameters that do not change, or if the command line is too short to specify all required parameters. This file must be specified as

`–file` *file*

The file will be interpreted as a list of specified parameters prior to all other parameters in the command line.

The `–file` option is not designed to contain multiple commands; only additional parameters for a single command can be specified.

Lines beginning with # in such a file are interpreted as comments and therefore ignored.

The `–file` option is not mentioned in the syntax diagrams of the individual commands.

14.1.1.2 Connection parameters

All commands need a connection to the StorMan Server. The necessary connection parameters are:

`–server` *stormanserver*

  Specifies the IP address (IPv4 or IPv6) or name of the StorMan Server (default: `localhost`).

`–port` *port*

  Specifies the port the server is listening on (default: 4178).

`–user` *user*

  Specifies the StorMan user for validation on the server (default: not specified).

`–pwd` *pwd*

  Specifies the StorMan password for validation on the server (default: not specified).

–connectparam

  substitutes these parameters in the syntax diagrams of the individual commands.

14.1.1.3 Help function

An internal help function is supported for every command by the `–help` option.

14.1.1.4 Additional common aspects

- If a parameter is specified more than once, the last value will be used.

- Parameters are not case-sensitive. Parameter values are case-sensitive except when specified otherwise. StorIDs and Pool IDs are stored as input with case (for display) but treated without case in all internal search functions.

- If a specified parameter is not supported, the command is rejected.

14.1.2 Common enumeration values

All StorMan commands use a common set of enumerations for input parameters and output values. These enumeration are only provided for show functions with parameter `-format xml/xmlp`. These are described below:

StorMan access mode types

The enumeration `STORMAN_ACCESS_MODE` defines how storage units may be accessed on a certain host: for exclusive or shared use. This property is used when selecting a pool for storage unit creation as well as when attaching (accessing) a storage unit which has already been created for a host.

Value	Meaning
STORMAN_ACCESS_MODE_ANY	Arbitrary access mode (only applicable with request)
STORMAN_ACCESS_MODE_NONE	Current state of storage unit: not yet attached to any host (only available as information value)
STORMAN_ACCESS_MODE_EXCLUSIVE	The host will be attached (by request) or is shown as attached exclusively. There is no further attach to another host allowed.
STORMAN_ACCESS_MODE_SHARED	The host can be attached together with other hosts which are already attached with shared access mode

StorMan resource types

The enumeration `STORMAN_RESOURCE_TYPE` defines the resource type property which designates a pool resource (to be) reserved.

Value	Meaning
STORMAN_RESOURCE_TYPE_ANY	Indicates that the storage unit requested for reservation may be predefined as well as dynamic
STORMAN_RESOURCE_TYPE_PREDEFINED	Indicates a certain storage unit as statically predefined
STORMAN_RESOURCE_TYPE_DYNAMIC	Indicates a certain storage unit as (to be) dynamically created

StorMan connection types

The enumeration `STORMAN_CONNECTION_TYPE` represents possible connection type values.

Value	Meaning
STORMAN_CONNECTION_TYPE_ANY	No specific storage type requested
STORMAN_CONNECTION_TYPE_FC	SAN connected by FC
STORMAN_CONNECTION_TYPE_ISCSI	SAN connected by iSCSI
STORMAN_CONNECTION_TYPE_SAS	SAN connected by SAS (serial attached SCSI)
STORMAN_CONNECTION_TYPE_FCoE	SAN connected by FCoE
STORMAN_CONNECTION_TYPE_NAS	NAS connected Storage

StorMan storage models

The enumeration `STORMAN_STORAGE_MODEL` represents a list of supported storage system models.

Value	Meaning
STORMAN_STORAGE_MODEL_ANY	No specific model requested
STORMAN_STORAGE_MODEL_SYMMETRIX	Symmetrix storage system from EMC
STORMAN_STORAGE_MODEL_CLARIION	CLARiiON CX storage system / FibreCAT CX storage system
STORMAN_STORAGE_MODEL_NETAPPFILER	NetApp Filer from NetApp
STORMAN_STORAGE_MODEL_ETERNUS	ETERNUS DX storage system
STORMAN_STORAGE_MODEL_OTHER	Storage system monitored by specific scripts
STORMAN_STORAGE_MODEL_USER	Storage system registered manually

Storman volume status

The enumeration `STORMAN_VOLUME_STATUS` represents a list of supported of volume status.

Value	Meaning
STORMAN_VOLUME_STATUS_ANY	No specific status requested
STORMAN_VOLUME_STATUS_UNKNOWN	State of volume is not known
STORMAN_VOLUME_STATUS_READY	Storage volume is in status READY
STORMAN_VOLUME_STATUS_NOT_READY	Storage volume is in status NOT_READY
STORMAN_VOLUME_STATUS_READ_ONLY	Storage volume is in status READ_ONLY
STORMAN_VOLUME_STATUS_ERROR	Storage volume is in status ERROR

Storman RAID levels

The enumeration `STORMAN_RAID_LEVEL` represents a list of supported raid levels.

Value	Meaning
STORMAN_RAID_LEVEL_ANY	No specific RAID requested
STORMAN_RAID_LEVEL_UNKNOWN	Unknown RAID level
STORMAN_RAID_LEVEL_DISK	No RAID level
STORMAN_RAID_LEVEL_0	Striped
STORMAN_RAID_LEVEL_1	mirrored disk
STORMAN_RAID_LEVEL_10	mirrored disk with striping
STORMAN_RAID_LEVEL_3	striping with parity
STORMAN_RAID_LEVEL_5	striping with distributed parity
STORMAN_RAID_LEVEL_6	striping with 2 distributed parities

Storman mirror volume usage

The enumeration `STORMAN_MIRROR_VOLUME` represents a list of supported values for the usage of a volume in mirroring relationships.

Value	Meaning
STORMAN_MIRROR_VOLUME_NONE	Not used for mirroring
STORMAN_MIRROR_VOLUME_ORIG_LOCAL	Used as original in local mirroring relationship
STORMAN_MIRROR_VOLUME_MIRROR_LOCAL	Used as local mirror
STORMAN_MIRROR_VOLUME_ORIG_REMOTE	Used as original in remote mirroring relationship
STORMAN_MIRROR_VOLUME_MIRROR_REMOTE	Used as remote mirror
STORMAN_MIRROR_VOLUME_UNKNOWN	Unknown mirroring relationship
STORMAN_MIRROR_VOLUME_ANY	No specific mirroring requested

Storman mirror type usage

The enumeration `STORMAN_MIRROR TYPE` represents a list of supported values for the mirroring type.

Value	Meaning
STORMAN_MIRROR_TYPE_NONE	No mirror type
STORMAN_MIRROR_TYPE SNAP	Used as snap
STORMAN_MIRROR_TYPE FULL	Used as full mirror
STORMAN_MIRROR_TYPE_UNKNOWN	Unknown mirror information

Storman mirror replica usage

The enumeration STORMAN_MIRROR_REPLICA represents a list of supported values for the replica location.

Value	Meaning
STORMAN_MIRROR_REPLICA_NONE	No replica
STORMAN_MIRROR_REPLICA_LOCAL	Used as local replica
STORMAN_MIRROR_REPLICA_REMOTE	Used as remote replica

Storman mirror remote replication mode

The enumeration STORMAN_REMOTE_COPY_MODE represents a list of supported values for the remote replication mode.

Value	Meaning
STORMAN_REMOTE_COPY_MODE_UNKNOWN	Unknown remote replication mode
STORMAN_REMOTE_COPY_MODE_ANY	Remote replication mode not specified
STORMAN_REMOTE_COPY_MODE_SYNC	Synchronous remote replication mode
STORMAN_REMOTE_COPY_MODE_ASYN	Asynchronous, conisitent remote replication mode
STORMAN_REMOTE_COPY_MODE_ASYNC_STAC	Asynchronous stack mode (ETERNUS DX)
STORMAN_REMOTE_COPY_MODE_ASYNC_THROUGH	Asynchronous through mode (ETERNUS DX)

Storman mirror pair status

The enumeration STORMAN_MIRROR_PAIR_STATUS represents a list of supported of volume status.

Value	Meaning
STORMAN_MIRROR_PAIR_STATUS_INITIALIZING	Mirror pair is in initialization
STORMAN_MIRROR_PAIR_STATUS_INITIALIZED	Mirror pair is initialized
STORMAN_MIRROR_PAIR_STATUS_FAILED	Mirror pair is broken after a failure
STORMAN_MIRROR_PAIR_STATUS_TERMINATING	Mirror pair is in termination
STORMAN_MIRROR_PAIR_STATUS_RESTORING	Mirror pair is restoring to original
STORMAN_MIRROR_PAIR_STATUS_RESYNCING	Mirror pair is resyncing to mirror
STORMAN_MIRROR_PAIR_STATUS_SPLIT	Mirror pair is split but out of synch, the mirror is accessible from host
STORMAN_MIRROR_PAIR_STATUS_SPLIT_AND_SYNC	Mirror pair is split and synchronized, the mirror is accessible from host
STORMAN_MIRROR_PAIR_STATUS_SUSPENDED	Mirror pair is suspended, the mirror is not accessible from host
STORMAN_MIRROR_PAIR_STATUS_FAILED_OVER	Mirror pair is failed over, for remote replication in case of access to the target volumes only
STORMAN_MIRROR_PAIR_STATUS_SYNCHRONIZING	Mirror pair is synchronizing
STORMAN_MIRROR_PAIR_STATUS_SYNCHRONIZED	Mirror pair is synchronized, the mirror is not accessible from host
STORMAN_MIRROR_PAIR_STATUS_NOT_APPLICABLE	pair status is not applicable to the pair
STORMAN_MIRROR_PAIR_STATUS_WAIT_FOR_SYNC	Mirror pair is waiting for synchronization
STORMAN_MIRROR_PAIR_STATUS_PARTITIONED	Mirror pair is partitioned

Value	Meaning
STORMAN_MIRROR_PAIR_STATUS_BROKEN	Mirror pair is broken
STORMAN_MIRROR_PAIR_STATUS_SPLIT_NOT_CONSISTENT	Mirror pair is split during a synchronization process, the mirror is accessible from host, but data is not consistent

14.1.3 Common supported value ranges

All StorMan commands have some common parameters with a defined range of supported values specified below:

Value	Meaning and supported values
ip | *hostip*	**IP address:** a string of 4 decimal numbers delimited by '.' each in the range from 0 to 255
hostname	**host name:** a string containing alphabetic characters ('A', 'B',... 'Z', 'a', 'b',... 'z', numbers and other signs except: '\', '/', '*', '? ', '"', '<', '>', '|', ' '. Internally it is not case-sensitive.
poolid	**pool ID:** a string starting with an alphabetic character ('A', 'B',... 'Z', 'a', 'b',... 'z' followed by up to 254 alphabetic characters, numbers, '_' or '.'. The pool ID must be unique. Internally it is not case-sensitive.
storid	**storage ID** (storID): a string starting with an alphabetic character ('A', 'B', ..., 'Z', 'a', 'b', ..., 'z') followed by up to 254 characters, numbers, '+', '-', '_' or '.' It should not start with prefix 'SM_' (regardless of case) as this is reserved for internal use. Internally it is not case-sensitive
wwpn	**WWPN:** a string of 16 or 32 hexadecimal numbers delimited by ':', '-' or without delimiter

The wildcard '*' is supported for alphabetic character strings as specified in the CLI description. Generally it can be used at any place of the string.

On Linux systems strings that do not contain only letters and numbers, must be enclosed in quotation marks (eg "#all ').

14.2 storadmin

The `storadmin` command provides specific functions to administrate StorMan on Linux platforms (including M2000 and MARS):

- `storadmin configsrv` collects host information information and saves it to StorMan repository for

    - SE Server Units and Application Units (on M2000 only)

    - SQ Server Units and Application Units (on MARS only)

- `storadmin getlogs` collects diagnostic information in an archive on Linux platforms

- `storadmin remlogs` removes all internal log files on Linux platforms(including M2000 and MARS)

On M2000 the `storadmin` functions are available for service accounts. On MARS the `storadmin` functions are available for administrator and service accounts.

The parameters `-user` and `-pwd` specify the StorMan user authorization.

–user *user*

Specifies the user for validation on the server (default: not specified).

–pwd *pwd*

Specifies the password for validation on the server (default: not specified).

14.2.1.1 storadmin configsrv

This function is supported M2000 for SE Servers and on MARS for SQ Server only:

Collects all server names and IP addresses of the Server Units and Application Units managed by M2000 or MARS respectively and adds the obtained host information to the StorMan repository.

This is especially helpful during the initial setup, but also to detect server configuration changes later on.

Syntax

```
storadmin configsrv [ –user user –pwd pwd ]
```

There are no further parameters.

Return codes

Error Code	Error Type
0	Function successful
1	Parameter Error
3	Function not supported
4	Function not successful

Example

```
storadmin configsrv
```

Check the result with:
```
storcfg host –show
```

14.2.1.2 storadmin getlogs

Collects all diagnostic information in a g-zipped tar archive. The resulting file is located at:

– `storManLogs.tar.gz` for Linux

– `/tmp/storManLogs.tar.gz` for M2000 or MARS.

Syntax

```
storadmin getlogs [ -user user -pwd pwd ]
```

There are no further parameters.

14.2.1.3 storadmin remlogs

Removes all internal log files.

Syntax

```
storadmin remlogs [ -user user -pwd pwd ]
```

There are no further parameters.

14.3 storattach

Makes one or multiple storage volumes visible for a host using the LUN masking capabilities of the storage system.

Syntax

```
storattach -storid storid[,...] [ -hostname hostname ] [ -hostip ip ]
            [ -hostwwpn wwpn ] [ -targetwwpn wwpn ] [ -hostlun lun[,...] ] -connectparam ...
```

Parameters

-storid *storid*[,...]

> Specifies the previously assigned storID(s) of one or a list of source volumes as unique identifier(s). This parameter is mandatory.

-hostname *hostname*

> Specifies name of the server to be provisioned with storage (attached). The volumes will be attached to all wwpns of the specified server.

-hostip *ip*

> Specifies the IP address of the server to be provisioned with storage (attached). The volumes will be attached to all WWPNs of the specified server.

-hostwwpn *wwpn*

> Specifies the server's HBA WWPN(s) to which the volume should be attached. This can be used to define single-path-connections.

> Only for ETERNUS DX storage systems, a list of more than 1 WWPN is supported, refer to chapter "Special Properties of Fujitsu ETERNUS DX storage systems" – "LUN mapping and LUN masking".

-targetwwpn *wwpn*

> Supported for ETERNUS DX storage systems only. Specifies the (list of) port WWPNs of the storage system and defines the sequence they are connected with the (list of) HBA ports specified by –hostwwpn.

> This parameter is only supported together with –hostwwpn to assure the correct connection sequence between host ports and target ports. It cannot be used if the server is specified by its host name or IP adress. If -targetwwpn is specified, their number must correspond to the number of specified elements of parameter -hostwwpn.

-hostlun *lun*[,...]

> Specifies one or a list for the device number(s) that can be seen by the host.
> Possible values: 0..255; e.g. –*hostlun* 128.
> For BS2000 connections an ETERNUS DX system the range of possible values is 0..4095.
> If this parameter is not specified, they are assigned by the storage system. If -hostlun is specified, its number must correspond to the number of specified elements of parameter -storid.

> If a SCSI host LUN is specified, make sure that this number is not already used for another device on the same host via the same storage ports – otherwise STORMAN_INV_PARAMETERS will be returned.

> This parameter is not supported for Symmetrix DMX systems (and rejected with STORMAN_FCT_NOT_POSSIBLE). In a Symmetrix DMX system, the host LUN number is only defined by the mapping of the volume.

-*connectparam*

> Refer to the description of the "Connection parameters".

Notes

1. `-storid` must be specified to specify the storage unit(s).

2. One of the parameters `hostname`, `hostip` or `hostwwpn` must be specified to specify the host exactly. If `hostname` or `hostip` is specified, StorMan must have the necessary information to determine the WWPNs of the host (e.g. supplied with `storcfg host` in its internal repository) and the storage is attached to all known host ports.

3. If the host specification is redundant, i.e. as a result of `-hostname` and / or `-hostip` and / or `-hostwwpn`, the specified parameters must be consistent.

Return codes

Error Code	Name	Error Type
0	STORMAN_OK	Function successful
1	STORMAN_NO_ACTION	Nothing to do
2	STORMAN_NOT_COMPLETE	Function only partly completed
3	STORMAN_VER_NOT_SUPP	Version not supported
7	STORMAN_INV_PARAMETERS	Specified parameters invalid
10	STORMAN_STORID_NOT_ FOUND	StorID does not yet exist
12	STORMAN_INV_ACCESS_MODE	Specified access mode not allowed or incompatible. E.g. a volume from a pool with access-mode=shared should be attached to more than one host.
13	STORMAN_COMMUNIC_ERROR	Error in StorMan communication between client and server.
14	STORMAN_STORAGE_LOCKED	Storage unit currently locked
16	STORMAN_FAULT	Unexpected error in function
18	STORMAN_NOT_FOUND	No wwpn for the specified host was found in the repository
22	STORMAN_RESOURCE_NOT_ AVAILABLE	CIM Server or database not available
25	STORMAN_CIMOM_AUTH_ FAILED	User / password for CIM Server not valid
36	STORMAN_AUTH_FAILED	User / password for StorMan not valid

14.4 storauth

The `storauth` command is used to manage the StorMan user authentification and StorMan Roles.

```
storauth object –function –functionparam ... –connectparam ...
```

The first parameter specifies the object group of the repository that is configured:

First parameter *object*	Object
user	StorMan user

The second parameter specifies the function. Most commands support at least the following functions:

Second parameter function	Meaning
–add	adds an object to the repository
–mod	modifies the properties of an object
–rem	removes an object from the repository
–show	shows the attributes of one or more object(s)

If the `–show` function is specified, the output format can be selected by specifying the `-format` parameter.

14.4.1 storauth user

`storauth user` is used to manage StorMan user authentification. This command is restricted to the administrator of StorMan.

After initialization, this command should be used to restrict the access tor StorMan server.

```
Storauth user –function –functionparam ... –connectparam ...
```

Functions

The *–function* parameter supplies one of the following functions:

–add

   defines a new user and his authorization in the StorMan administration

–mod

   modifies the attributes of an existing StorMan user

–rem

   removes a StorMan user from the administration

–show

   displays information related to StorMan users

storauth user –add

Adds a new user to the StorMan administration and defines its access password. Additionally it assigns the StorMan Roles that entitle the StorMan User for a specific set of functions and StorMan pools (storage resources).

Syntax

```
storauth user –add –name user –passwd passwd –role role [–poolids poolid [,...]]
              –connectparam ...
```

Parameters

–name *user*

　　Specifies the new StorMan user.

–passwd *passwd*

　　Specifies the password for the new StorMan user.

–role *role*

　　Specify the predefined StorMan Role (StorAdmin, PoolAdmin or Info).

–poolids *poolid*[,...]

　　Specifies a list of storage pool ID's. Refer to the common description.

　　The parameter is only supported for StorMan Role PoolAdmin, otherwise rejected.

　　If the operand is not specified no storage pools are assigned. The whole storage configuration visible by StorMan is enabled by default for StorMan Role StorAdmin.

　　The pool ID #all enables the user's StorMan Roles for all storage pools configured in (managed by) StorMan.

　　Wildcards (*) are supported as suffix to specify multiple poolids (e.g. HA_* specifies all poolids starting with HA_...).

–*connectparam*

　　Refer to the description of the "Connection parameters".

Return codes

Error Code	Name	Error Type
0	STORMAN_OK	Function successful.
3	STORMAN_VER_NOT_SUPP	Version not supported.
7	STORMAN_INV_PARAMETERS	Specified parameters invalid.
13	STORMAN_COMMUNIC_ERROR	Error in StorMan communication between client and server
16	STORMAN_FAULT	Unexpected error in function.
20	STORMAN_ALREADY_EXISTS	Object already exists in repository
22	STORMAN_RESOURCE_NOT_ AVAILABLE	CIM Server or database not available
36	STORMAN_AUTH_FAILED	User / password for StorMan not valid

storauth user –mod

Modifies the attributes and authorization of a user already contained in the StorMan repository.

This command is restricted to the administrator i.e. StorMan Users disposing of the role StorAdmin. Only the function to change the user's own password is supported for every StorMan User
If a parameter is not specified, the corresponding attribute is not changed.

Syntax

```
storauth user –mod –name user –passwd passwd [–newname newuser]
              [–newpasswd newpasswd ] [–newrole  role]
                    [–newpoolids poolid[,...]]   –connectparam ...
```

Parameters

–name *user*

    Specifies the StorMan user to be modified.

–passwd *passwd*

    Specifies the password of the StorMan user to be modified.

–newname *newuser*

    If specified, also a new user name is set for the user specified with –name.

–newpasswd *newpasswd*

    Specifies a new password for the specified StorMan user.

–newrole role

    Specify a new role for the StorMan User (StorAdmin, PoolAdmin or Info). Any eventually existing StorMan Role is replaced by the new one.

    StorMan Role StorAdmin cannot be replaced if there is only one StorMan User with this role.

–newpoolids poolid[,...]

    Replaces storage pool IDs from the user's manageable storage pools. Specifies a list of storage pool ID's.

    Any eventually existing storage pool IDs in the list of user's manageable storage pools are replaced.

    The pool ID *#all* enables the user's roles for all storage pools configured in StorMan.

Wildcards (*) are supported as suffix to specify multiple poolids (e.g. HA_* specifies all poolids starting with HA_...)–connectparam

    Refer to the description of the "Connection parameters".

Return codes

Error Code	Name	Error Type
0	STORMAN_OK	Function successful.
3	STORMAN_VER_NOT_SUPP	Version not supported.
7	STORMAN_INV_PARAMETERS	Specified parameters invalid.
13	STORMAN_COMMUNIC_ERROR	Error in StorMan communication between client and server
16	STORMAN_FAULT	Unexpected error in function.
18	STORMAN_NOT_FOUND	Specified host could not be found.
20	STORMAN_ALREADY_EXISTS	Object already exists in repository
22	STORMAN_RESOURCE_NOT_ AVAILABLE	CIM Server or database not available
36	STORMAN_AUTH_FAILED	User / password for StorMan not valid

storauth user –rem

Removes a StorMan user from the StorMan administration. All StorMan roles and storage pools assignments to the StorMan user are removed too. This command is restricted to the administrator i.e. StorMan Users disposing of the role `StorAdmin`.

Syntax

```
storauth user –rem –name user –connectparam ...
```

Parameters

–name *user*

Specifies the StorMan user to be removed from StorMan.

Removing the last StorMan User with StorMan Role `StorAdmin` is inhibited as long as there is still any other user in StorMan administration.

After removing the last user `storadmin` the complete StorMan functions are available again for all connecting StorMan servers. No more authorization checks are done.

–connectparam

Refer to the description of the "Connection parameters".

Return codes

Error Code	Name	Error Type
0	STORMAN_OK	Function successful.
3	STORMAN_VER_NOT_SUPP	Version not supported.
7	STORMAN_INV_PARAMETERS	Specified parameters invalid.
13	STORMAN_COMMUNIC_ERROR	Error in StorMan communication between client and server
16	STORMAN_FAULT	Unexpected error in function.
18	STORMAN_NOT_FOUND	Specified host could not be found.
22	STORMAN_RESOURCE_NOT_ AVAILABLE	CIM Server or database not available
36	STORMAN_AUTH_FAILED	User / password for StorMan not valid

storauth user –show

Displays information related to the specified StorMan user.

Syntax

```
storauth user –show [ –name user ] [ –format format ] –connectparam ...
```

Parameters

–name *user*

> Selects the StorMan user to be listed. If –name is not specified all StorMan users are listed.

All parameters above can be specified partially using '*' as wildcard.

–format *format*

> Specifies the output format. Permitted values: std (default) | short | xml | xmlp.

> std selects a multi-line output: Each attribute of an object is displayed on a separate line. This is also the default output format.

> short displays only the id / name of an object on a separate line.

> xml writes the output in xml format in a single line.

> xmlp writes the output in xml format, but in a multi-line format for better human readability (pretty print).

–*connectparam*

> Refer to the description of the "Connection parameters".

Return codes

Error Code	Name	Error Type
0	STORMAN_OK	Function successful.
3	STORMAN_VER_NOT_SUPP	Version not supported.
7	STORMAN_INV_PARAMETERS	Specified parameters invalid.
13	STORMAN_COMMUNIC_ERROR	Error in StorMan communication between client and server
16	STORMAN_FAULT	Unexpected error in function.
22	STORMAN_RESOURCE_NOT_ AVAILABLE	CIM Server or database not available
36	STORMAN_AUTH_FAILED	User / password for StorMan not valid

Example

Output for user with –format std (default)

```
storauth user -show -name my*

user      myself
  roles:
      StorPoolAdmin
  pool IDs:
      HA_sq210_x
      HA_sq210_y
```

Output for user with –format xmlp

```
storauth user –show –name d* –format xmlp
```

14.5 storcheck

Checks the availability of the StorMan Server. The check covers the availability of the StorMan Server itself, its database, and if at least one of the configured CIM-OMs is accessible.

The result message of the check is directed to stdout and the exit code is set (see return codes below).

Syntax

```
storcheck [ –timeout timeout ]–connectparam ...
```

Parameters

– timeout *timeout*

    Specifies a timeout value in seconds the command waits for a response of the checked CIM-OM. Default: 60 seconds.

–connectparam

    Refer to the description of the "Connection parameters".

Return codes

Error Code	Text	Error Type
0	OK	Everything OK
1	StorMan not reachable	The StorMan Server is not reachable
2	No Database	The StorMan Server is running but the database has failed
3	No CIM-OM	The StorMan Server is running and the database is up but no CIM-OM is reachable
7	STORMAN_INV_PARAMETERS	Specified parameters invalid
36	STORMAN_AUTH_FAILED	User / password for StorMan not valid

14.6 storcfg

The `storcfg` command is used to configure the StorMan repository.

`storcfg` *object* *–function* *–functionparam* `. . .` *–connectparam* `. . .`

The first parameter specifies the object group of the repository that is configured:

First parameter *object*	Object
`host`	hosts for provisioning or hosting a CIM-OM
`cimom`	CIM Object Manager
`system`	storage systems managed by StorMan
`pool`	pools to organize logical volumes
`volume`	logical volumes managed by StorMan

The second parameter specifies the function. Most commands support at least the following functions:

Second parameter *function*	Meaning
`–add`	adds an object to the repository
`–discover`	starts a discover for the object to obtain updated date
`–mod`	modifies the properties of an object
`–rem`	removes an object from the repository
`–show`	shows the attributes of one or more object(s)

If the `–show` function is specified, the output format can be selected by specifying the `–format` parameter.

14.6.1 storcfg host

`storcfg host` is used to manage host information in a StorMan configuration. The host information is necessary to attach a device to all FC ports of a host in one step and to display a storage volume with all its connection information related to a host.

`storcfg host` *–function* *–functionparam* `. . .` *–connectparam* `. . .`

Functions

The *–function* parameter supplies one of the following functions:

–add

  adds a new host to the StorMan repository

–mod

  modifies the attributes of a host already contained in StorMan repository

–rem

  removes a host from the StorMan repository

–show

  displays information related to hosts

14.6.1.1 storcfg host –add

Adds a new host to the StorMan repository.

Syntax

```
storcfg host -add { -name hostname | -ip ip[,...] }
                   [ -wwpn wwpn[,...] ] [ -iqn iqn[,...] ] -connectparam ...
```

Parameters

−name *hostname*

> Specifies the name of the host.
>
> Refer to the common description of "hostname" in chapter "Common supported value ranges".

−ip *ip*[,...]

> Specifies a list of IP addresses delimited by comma. If no hostname is specified, one of the IP addresses is used as hostid.

−wwpn *wwpn*[,...]

> Specifies a list of WWPNs delimited by comma.

−iqn *iqn*[,...]

> Specifies a list of IQNs for iSCSI connected storage delimited by comma.

At least one of the parameters -name or -ip must be specified and have a non-empty value.

−*connectparam*

> Refer to the description of the "Connection parameters".

Return codes

Error Code	Name	Error Type
0	STORMAN_OK	Function successful
3	STORMAN_VER_NOT_SUPP	Version not supported
7	STORMAN_INV_PARAMETERS	Specified parameters invalid
13	STORMAN_COMMUNIC_ERROR	Error in StorMan communication between client and server
16	STORMAN_FAULT	Unexpected error in function
20	STORMAN_ALREADY_EXISTS	Object already exists in repository
22	STORMAN_RESOURCE_NOT_ AVAILABLE	CIM Server or database not available
36	STORMAN_AUTH_FAILED	User / password for StorMan not valid

14.6.1.2 storcfg host –mod

Modifies the attributes of a host already contained in the StorMan repository.

If a parameter is not specified, the corresponding attribute is not changed.

Syntax

```
storcfg host –mod { –name hostname | –ip ip | –wwpn wwpn | –iqn iqn }
                  [ –newname hostname ]
                  { –newip ip[,...] | –addip ip | –remip ip }
                  { –newwwpn wwpn[,...] | –addwwpn wwpn | –remwwpn wwpn }
                  { –newiqn iqn[,...] | –addiqn iqn | –remiqn iqn }
                  –connectparam ...
```

Parameters

–name *hostname*

> Identifies the host by its name.

–ip *ip*

> Identifies the host by an IP address belonging to the host.

–wwpn *wwpn*

> Identifies the host by a WWPN belonging to the host.

–iqn *iqn*

> Identifies the host by a IQN belonging to the host.

Exactly one of the parameters –name, -ip, –wwpn or –iqn must be specified to identify the host.

–newname *hostname*

> Specifies the new name of the host. If the –newname is specified without a value, the name is removed.

> Refer to the common description of "hostname" in chapter "Common supported value ranges".

-newip ip[,...]

> Specifies a list of IP addresses delimited by comma. If no hostname is specified, one of the IP addresses is used as hostid.

–addip *ip*[,...]

> Assigns a single IP address to the host.

–remip *ip*

> Deassigns a single IP address from the host.

–newwwpn *wwpn*[,…]

> Assigns a list of WWPNs delimited by comma to assign to the host.

–addwwpn *wwpn*

> Assigns a single WWPN to the host.

–remwwpn *wwpn*

> Deassign a single WWPN from the host.

–newiqn *iqn*[,…]

> Assigns a list of IQN's delimited by comma to assign to the host.

–addiqn *iqn*

Assigns a single IQN to the host.

–remiqn *iqn*

Deassigns a single IQN from the host.

–connectparam

Refer to the description of the "Connection parameters".

Only one of the parameters –newip, –addip or –remip can be specified to modify the IP address assignment to the host. Combined input of these parameters is rejected with STORMAN_INV_PARAMETERS. If a specified IP address is already assigned to a different host, the (re)assignment of this IP address is rejected with return code STORMAN_ALREADY_EXISTS.

Only one of the parameters -newwwpn, –addwwpn or –remwwpn can be specified to modify the WWPN assignment to the host. Combined input of these parameters is rejected with STORMAN_INV_PARAMETERS. If a specified WWPN is already assigned to a different host, the (re)assignment of this WWPN is rejected with return code STORMAN_ALREADY_EXISTS.

Only one of the parameters -newiqn, –addiqn or –remiqn can be specified to modify the IQN assignment to the host. Combined input of these parameters is rejected with STORMAN_INV_PARAMETERS. If a specified IQN is already assigned to a different host, the (re)assignment of this IQN is rejected with return code STORMAN_ALREADY_EXISTS.

Return codes

Error Code	Name	Error Type
0	STORMAN_OK	Function successful
3	STORMAN_VER_NOT_SUPP	Version not supported
7	STORMAN_INV_PARAMETERS	Specified parameters invalid
13	STORMAN_COMMUNIC_ERROR	Error in StorMan communication between client and server
16	STORMAN_FAULT	Unexpected error in function
18	STORMAN_NOT_FOUND	Specified host could not be found
20	STORMAN_ALREADY_EXISTS	Object already exists in repository
22	STORMAN_RESOURCE_NOT_ AVAILABLE	CIM Server or database not available
36	STORMAN_AUTH_FAILED	User / password for StorMan not valid

14.6.1.3 storcfg host –rem

Removes a host from the StorMan repository. If this host is also acting as one or more CIM-OMs, these are also removed from repository.

Syntax

```
storcfg host –rem { –name hostname | –ip ip | –wwpn wwpn | –iqn iqn }
                     –connectparam ...
```

Parameters

–name *hostname*

> Identifies the host by its name. This parameter supports wildcards (*).

–ip *ip*

> Identifies the host by an IP address belonging to the host.

–wwpn *wwpn*

> Identifies the host by a WWPN belonging to the host.

–iqn *iqn*

> Identifies the host by a IQN belonging to the host.

Exactly one of the parameters -name, -ip, –wwpn or –iqn must be specified with a non-empty value to identify the host.

–*connectparam*

> Refer to the description of the "Connection parameters".

Return codes

Error Code	Name	Error Type
0	STORMAN_OK	Function successful
3	STORMAN_VER_NOT_SUPP	Version not supported
7	STORMAN_INV_PARAMETERS	Specified parameters invalid
13	STORMAN_COMMUNIC_ERROR	Error in StorMan communication between client and server
16	STORMAN_FAULT	Unexpected error in function
18	STORMAN_NOT_FOUND	Specified host could not be found
22	STORMAN_RESOURCE_NOT_ AVAILABLE	CIM Server or database not available
36	STORMAN_AUTH_FAILED	User / password for StorMan not valid

14.6.1.4 storcfg host –show

Displays information related to the specified hosts.

Syntax

```
storcfg host -show { -name hostname | -ip ip | -wwpn wwpn | -iqn iqn }
               [ -format format ] -connectparam ...
```

Parameters

–name *hostname*

> Selection by the name of the host.

–ip *ip*

> Selection by the IP address of the host.

–wwpn *wwpn*

> Selection by the WWPN of a host

–iqn *iqn*

> Selection by the IQN of a host.

All parameters above can be specified partially using * as wildcard. Using only * selects all hosts with this specific attribute supplied, regardless of its value. (e.g. –iqn * selects all hosts with iSCSI connectivity).

–format *format*

> Specifies the output format. Permitted values: std (default) | short | medium | xml | xmlp.
>
> std selects a multi-line output: Each attribute of an object is displayed on a separate line. This is also the default output format.
>
> short displays only the id / name of an object on a separate line.
>
> medium displays the id / name and some basic properties of an object on a separate line.
>
> xml writes the output in xml format in a single line.
>
> xmlp writes the output in xml format, but in a multi-line format for better human readability (pretty print).

–connectparam

> Refer to the description of the "Connection parameters".

Return codes

Error Code	Name	Error Type
0	STORMAN_OK	Function successful
3	STORMAN_VER_NOT_SUPP	Version not supported
7	STORMAN_INV_PARAMETERS	Specified parameters invalid
13	STORMAN_COMMUNIC_ERROR	Error in StorMan communication between client and server
16	STORMAN_FAULT	Unexpected error in function
22	STORMAN_RESOURCE_NOT_ AVAILABLE	CIM Server or database not available
36	STORMAN_AUTH_FAILED	User / password for StorMan not valid

Examples

1. Output for hosts with `-format std` (default)

   storcfg host –show –name d*

   ```
   hostID type STORMAN_HOSTID_TYPE_NAME
   hostID      ducksoup
     hostname    ducksoup
     no IPs found
     WWPNs:
         210000C09F956746
         210000C09F956747
     no IQNs found
   ```

2. Output for hosts with `-format medium`

   storcfg host –show –name d* -format medium
   ```
   ducksoup WWPNs:210000C09F956746,210000C09F956747
   ```

3. Output for hosts with `-format xmlp`

   storcfg host –show –name d* -format xmlp

   ```
   <Response>
     <ReturnCode>
       <Code>STORMAN_OK</Code>
       <CodeNumber>0</CodeNumber>
       <Text>Successful</Text>
       <SubCode>0</SubCode>
       <SubCodeText>STORMAN_SUB_NONE</SubCodeText>
     </ReturnCode>
     <Result>
       <Host>
         <HostIDType>STORMAN_HOSTID_TYPE_NAME</HostIDType>
         <HostID>ducksoup</HostID>
         <HostName>ducksoup</HostName>
         <IPs>
         </IPs>
         <WWPNs>
           <WWPN>210000C09F956746</WWPN>
           <WWPN>210000C09F956747</WWPN>
         </WWPNs>
         <IQNs>
         </IQNs>
       </Host>
        </Result>
   </Response>
   ```

14.6.2 storcfg cimom

`storcfg cimom` is used to manage CIM Object Manager information for SMI-S controlled storage systems and storage array information (API Server) for API controlled storage arrays in the StorMan repository. Please note that all following references to CIM-OMs are valid for the storage array information for API controlled storage arrays respective.

```
storcfg cimom –function –functionparam ... –connectparam ...
```

Functions

The *–function* parameter supplies one of the following functions:

–add

adds a new CIM-OM information or the new API Server to the StorMan repository

–mod

modifies the attributes of an existing CIM-OM or the API Server in the StorMan repository

–rem

removes CIM-OM information or the API Server from the StorMan repository

–discover

requests to discover the CIM-OM all available storage systems

–show

displays information related to hosts hosting CIM-OMs or API Servers

14.6.2.1 storcfg cimom –add

Adds the new CIM-OM information or the new API Server to the StorMan repository.

Syntax

```
storcfg cimom –add { –name hostname | –ip ip } [–http port ] [ –https port ]
                    [ –interface type ] [ –cimuser user ]
                    [ –cimpwd pwd ] [–discover discover] –connectparam ...
```

Parameters

–name *hostname*

Specifies the name of the host the CIM-OM is running on or the name of the API Server (NetApp filer).

Refer to the common description of "hostname" in chapter "Common supported value ranges".

–ip *ip*

Specifies the IP address of the host the CIM-OM is running on or the IP address of the API Server (NetApp filer).

–http *port*

Specifies the port the CIM-OM is listening on
(typically: if `-https` is not specified: 5988 for interface type smis, 80 for interface type netappapi).

−`https` *port*

> Specifies the secure port the CIM-OM is listening on (typically: 5989). Supported for `-interface smis` only.

−`interface` *type*

> Specifies the interface type for the *hostname*:
>
> `-interface smis` (default) for SMI-S Provider / CIM-OMs.
>
> `-interface netappapi` for API Server of NetApp filers.

−`cimuser` *user*

> Specifies the user for the CIM-OM connection (default: none).

−`cimpwd` *pwd*

> Specifies the password for the CIM-OM connection (default: none).

−`discover` *discover*

> Specifies if any discover should be done for the new CIM-OM:
>
> `-discover no` starts no discover for the CIM-OM.
>
> `-discover query` (default) checks the CIM-OM for managed storage systems and starts an update process in the background.
>
> `-discover deep` refreshes the base information for the managed storage systems and starts an update process in the background (may last some minutes depending on configuration).
>
> By specifying this operand an additional call of `storcfg cimom -discover` … can be omitted.

At least one of the parameters `-name` or `-ip` must be specified. If no host with the specified name or ip exists in the StorMan repository, a new host is added.

Only one of the port types `-http` or `-https` can be specified.

The cimuser and cimpwd must be specified for the StorMan repository with the function `storcfg cimom -add` or `storcfg cimom -mod.`

- For FUJITSU embedded SMI-S Provider `cimuser` / `cimpwd` are the user name and password of the storage system.

- For EMC's SMI-S Provider `cimuser` / `cimpwd` is a user/password combination that is configured and mandatory for the specified CIM Server (refer to the vendor-specific Installation Guides for information on how to add a cimuser). The cimuser and cimpwd must be specified for the StorMan repository with the function `storcfg cimom -add` or `storcfg cimom -mod.`

For API Server of NetApp filers `cimuser` / `cimpwd` are used to logon to the filer.

−*connectparam*

> Refer to the description of the "Connection parameters".

Return codes

Error Code	Name	Error Type
0	STORMAN_OK	Function successful
2	STORMAN_NOT_COMPLETE	Function only partly completed
3	STORMAN_VER_NOT_SUPP	Version not supported
7	STORMAN_INV_PARAMETERS	Specified parameters invalid
13	STORMAN_COMMUNIC_ERROR	Error in StorMan communication between client and server
16	STORMAN_FAULT	Unexpected error in function
20	STORMAN_ALREADY_EXISTS	Object already exists in repository
22	STORMAN_RESOURCE_NOT_ AVAILABLE	CIM server or database not available
36	STORMAN_AUTH_FAILED	User / password for StorMan not valid

14.6.2.2 storcfg cimom –mod

Modifies the attributes of an existing CIM-OM or an API Server.

If a parameter is not specified, the corresponding attribute is not changed.

Any combination of the parameters `name`, `ip`, `interface` and `http` or `https` which uniquely identifies the CIM-OM may be used to specify the CIM-OM to be modified.

To change the name or IP address of the CIM-OM or of the API Server, please use the function `storcfg host -mod`.

Syntax

```
storcfg cimom –mod { –name hostname | –ip ip }    [-http port ] [-https port]
                   [ –newhttp port ] [ –newhttps port ] [ –newuser user ]
                   [ –newpwd pwd ] [-discover discover] –connectparam ...
```

Parameters

–name *hostname*

    Specifies the name of the host the CIM-OM is running on or the name of the API Server.

–ip *ip*

    Specifies the IP address of the host the CIM-OM is running on or the IP address of the API Server.

–http *port*

    Specifies the port the CIM-OM is listening on.

–https *port*

    Specifies the secure port the CIM-OM is listening on.

−newhttp *port*

> Specifies the new port the CIM-OM is listening on.

−newhttps *port*

> Specifies the new secure port the CIM-OM is listening on.

−newuser *user*

> Specifies the new user for the CIM-OM connection.

−newpwd *pwd*

> Specifies the new password for the CIM-OM connection.

−discover *discover*

> Specifies if any discover should be done for the CIM-OM:
>
> −discover no starts no discover for the CIM-OM.
>
> −discover query (default) checks the CIM-OM for managed storage systems and starts an update process in the background.
>
> −discover deep refreshes the base information for the managed storage systems and starts an update process in the background (may last some minutes depending on configuration).
>
> By specifying this operand an additional call of storcfg cimom −discover … can be omitted.

At least one of the parameters -name or -ip must be specified to select the CIM-OM.

−*connectparam*

> Refer to the description of the "Connection parameters".

Return codes

Error Code	Name	Error Type
0	STORMAN_OK	Function successful
2	STORMAN_NOT_COMPLETE	Function only partly completed
3	STORMAN_VER_NOT_SUPP	Version not supported
7	STORMAN_INV_PARAMETERS	Specified parameters invalid
13	STORMAN_COMMUNIC_ERROR	Error in StorMan communication between client and server
16	STORMAN_FAULT	Unexpected error in function
18	STORMAN_NOT_FOUND	Specified host was not found
22	STORMAN_RESOURCE_NOT_ AVAILABLE	CIM server or database not available
36	STORMAN_AUTH_FAILED	User / password for StorMan not valid

14.6.2.3 storcfg cimom –rem

Removes the CIM-OM or the API Server from the StorMan repository. The Storage Arrays and their corresponding pools managed by the removed CIM-OM are not removed from the repository as a further CIM-OM may be or may become available for management. The host information remains in the StorMan repository; only the description of the CIM-OM is removed.

Any combination of the parameters `name`, `ip`, `interface` and `http` or `https` which uniquely identifies the CIM-OM may be used to specify the CIM-OM to be removed.

Syntax

```
storcfg cimom –rem { –name hostname | –ip ip } [–http port ] [ -https port]
                    [ –interface type ] –connectparam ...
```

Parameters

–`name` *hostname*

> Specifies the name of the host, the name of the host the CIM-OM is running on or the name of the API Server. This parameter supports wildcards (*).

–`ip` *ip*

> Specifies the IP address of the host the CIM-OM is running on or the IP address of the API Server.

–`http` *port*

> Specifies the port the CIM-OM is listening on.

–`https` *port*

> Specifies the secure port the CIM-OM is listening on.

–`interface` *type*

> Specifies the interface type for the *hostname*.
>
> –`interface smis` for CIM-OMs.
>
> –`interface netappapi` for API Server (NetApp filers).

–connectparam

> Refer to the description of the "Connection parameters".

Return codes

Error Code	Name	Error Type
0	STORMAN_OK	Function successful
3	STORMAN_VER_NOT_SUPP	Version not supported
7	STORMAN_INV_PARAMETERS	Specified parameters invalid
13	STORMAN_COMMUNIC_ERROR	Error in StorMan communication between client and server
16	STORMAN_FAULT	Unexpected error in function
18	STORMAN_NOT_FOUND	Specified host was not found
22	STORMAN_RESOURCE_NOT_ AVAILABLE	CIM server or database not available
36	STORMAN_AUTH_FAILED	User / password for StorMan not valid

14.6.2.4 storcfg cimom –discover

Requests the CIM-OM to discover all available storage systems, i.e. to get or update all the configuration data of all storage systems which can be accessed by the respective CIM-OM. This function may take some time depending on the number of connected storage systems and their complexity.

Any combination of the parameters `name`, `ip`, `interface` and `http` or `https` which uniquely identifies the CIM-OM may be used to specify that the CIM-OM should discover storage systems. If only one CIM-OM is in the StorMan repository these parameters can be omitted.

In general, a storage system should not be managed using CIM Servers on different hosts concurrently if the CIM-OMs have repositories of their own. If more than one CIM Host have connections to one storage system, they should not run at the same time.

> **i** When adding a new CIM-OM or modifying a CIM-OM's attributes with `storcfg cimom -add` or `storcfg cimom -mod` a discover parameter can directly be used to request the CIM-OM to discover all available storage systems.

Syntax

```
storcfg cimom –discover { –name hostname | –ip ip } [–http port ] [ –https port]
                        [ –interface type ] –connectparam ...
```

Parameters

–`name` *hostname*

> Specifies the name of the host, the CIM-OM is running on or the name of the API Server. This parameter supports wildcards (*).

–`ip` *ip*

> Specifies the IP address of the host the CIM-OM is running on or the IP address of the API Server.

–`http` *port*

> Specifies the port the CIM-OM is listening on.

–`https` *port*

> Specifies the secure port the CIM-OM is listening on.

–`interface` *type*

> Specifies the interface type for the *hostname*.
>
> –`interface` `smis` for CIM-OMs.
>
> –`interface` `netappapi` for API Server (NetApp filers).

–*connectparam*

> Refer to the description of the "Connection parameters".

> **i** If the CIM-OM or the API Server are specified by their name, the name must be valid in the local network otherwise `STORMAN_RESOURCE_NOT_AVAILABLE` is returned. If the name is modified or removed from the network, the name can be removed or modified in the StorMan repository using `storcfg host -mod`.

Return codes

Error Code	Name	Error Type
0	STORMAN_OK	Function successful
3	STORMAN_VER_NOT_SUPP	Version not supported
7	STORMAN_INV_PARAMETERS	Specified parameters invalid
13	STORMAN_COMMUNIC_ERROR	Error in StorMan communication between client and server
16	STORMAN_FAULT	Unexpected error in function
18	STORMAN_NOT_FOUND	Specified host was not found
22	STORMAN_RESOURCE_NOT_ AVAILABLE	CIM server or database not available
25	STORMAN_CIMOM_AUTH_ FAILED	User / password for CIM Server not valid. If the CIM Server supports authentication.
36	STORMAN_AUTH_FAILED	User / password for StorMan not valid

14.6.2.5 storcfg cimom –show

Displays information related to hosts hosting CIM-OMs.

Syntax

```
storcfg cimom –show [ -system system ] [ –name hostname ] [ –ip ip ] [ –http port ]
        [ –https port ] [ –interface type ] [ –showsystems ]-format format –connectparam ...
```

Parameters

–system *system*

> Specifies the storage system name and selects all CIM-OMs or API Servers managing the respective storage system(s). This parameter supports wildcards (*). If the parameter is not specified, all CIM-OMs and API Servers in StorMan repository are displayed.

-name hostname

> Selects the CIM-OM or the API Server by its host name. This parameter supports wildcards (*).

–ip *ip*

> Selects the CIM-OM by its IP address.

–http *port*

> Selects the CIM-OM by the listening port.

–https *port*

> Selects the CIM-OM by the listening secure port.

–interface *type*

> Selects the interface type for the *hostname or ip*.

> –interface smis for CIM-OMs.

> –interface netappapi for API Server (NetApp filers).

–showsystems

> Selects and list all storage systems managed by a CIM-OM.

−`format` *format*

Specifies the output format. Permitted values: `std` (default) | `short` | `medium` | `xml` | `xmlp`.

`std` selects a multi-line output: Each attribute of an object is displayed on a separate line. This is also the default output format.

`short` displays only the id/name of an object on a separate line.

`medium` displays the id / name and some basic properties of an object on a separate line.

`xml` writes the output in xml format in a single line.

`xmlp` writes the output in xml format, but in a multi-line format for better human readability (pretty print).

−*connectparam*

Refer to the description of the "Connection parameters".

Return codes

Error Code	Name	Error Type
0	STORMAN_OK	Function successful
3	STORMAN_VER_NOT_SUPP	Version not supported
7	STORMAN_INV_PARAMETERS	Specified parameters invalid
13	STORMAN_COMMUNIC_ERROR	Error in StorMan communication between client and server
16	STORMAN_FAULT	Unexpected error in function
18	STORMAN_NOT_FOUND	Specified host was not found
22	STORMAN_RESOURCE_NOT_ AVAILABLE	CIM server or database not available
36	STORMAN_AUTH_FAILED	User / password for StorMan not valid

Examples

1. Output for CIM-OMs with `-format std` (default)

   storcfg cimom –show

   ```
   hostID type IP
   hostID     192.1.0.0
     hostname
     IP        192.1.0.0
     interop    interop
     http port  5988
     user       root
     connection OK
     interface  SMIS
     vendor     FUJITSU
     version    V04L00-0000
   ```

2. Output for CIM-OMs with `-format medium`

   storcfg cimom –show –format medium

   ```
   192.1.0.0 connection=Ok version=V05L00-0000
   ```

3. Output for CIM-OMs with `-format=xmlp`

storcfg cimom –show –format xmlp

```xml
<Response>
  <ReturnCode>
    <Code>STORMAN_OK</Code>
    <CodeNumber>0</CodeNumber>
    <Text>successful</Text>
    <SubCode>0</SubCode>
    <SubCodeText>STORMAN_SUB_NONE</SubCodeText>
    <MessageID>3</MessageID>
  </ReturnCode>
  <Result>
    <CIMOMs>
      <CIMOM>
        <HostIDType>STORMAN_HOSTID_TYPE_IP</HostIDType>
        <HostID>172.17.67.121</HostID>
        <HostName/>
        <IP>172.17.67.121</IP>
        <OwnIP>10.172.102.158</OwnIP>
        <PortHttp>5988</PortHttp>
        <PortHttps/>
        <User>root</User>
        <InteropNamespace>interop</InteropNamespace>
        <CliName/>
        <CliRefreshTime/>
        <CliInitSuccessful>false</CliInitSuccessful>
        <ConnectStatus>STORMAN_CIMOM_CONNECT_OK</ConnectStatus>
        <InterfaceType>STORMAN_INTERFACE_TYPE_SMIS</InterfaceType>
        <Vendor>Fujitsu</Vendor>
        <VersionString>V05L00-0000</VersionString>
        <IsProxyProvider>no</IsProxyProvider>
        <Locality>STORMAN_CIMOM_LOCALITY_UNKNOWN</Locality>
        <StorageSystems>
        </StorageSystems>
      </CIMOM>
    </CIMOMs>
  </Result>
</Response>
```

14.6.3 storcfg system

`storcfg system` is used to manage the storage systems in the StorMan repository.

The –add and –mod functions are not supported because storage systems are automatically added to the StorMan repository only as a result of a `storcfg cimom -discover` call.

```
storcfg  system –function –functionparam ... –connectparam ...
```

Functions

The *–function* parameter supplies one of the following functions:

–rem

    removes the storage system from the StorMan repository

–discover

    requests to rediscover the storage system, i.e. to completely update the configuration data of the specified storage system

–mod

    sets or modifies the user access to the storage systems to access performance data of ETERNUS DX systems via PMCC

–show

    displays information related to the storage systems

14.6.3.1 storcfg system –rem

Removes the storage system from the StorMan repository.

If pools are defined for the storage system concerned, they are also completely removed from the StorMan repository together with their logical volumes. Therefore the contents of the pools (logical volumes and their assigned storIDs) can no longer be managed by StorMan.

The configuration of the storage system and its assignment to servers is not touched by this function.

Syntax

```
storcfg system –rem –system system –connectparam ...
```

Parameters

–system *system*

    Specifies the system name of the storage system to be removed. This parameter is mandatory and supports wildcards (*).

–connectparam

    Refer to the description of the "Connection parameters".

Return codes

Error Code	Name	Error Type
0	STORMAN_OK	Function successful
3	STORMAN_VER_NOT_SUPP	Version not supported
7	STORMAN_INV_PARAMETERS	Specified parameters invalid
13	STORMAN_COMMUNIC_ERROR	Error in StorMan communication between client and server
16	STORMAN_FAULT	Unexpected error in function
22	STORMAN_RESOURCE_NOT_ AVAILABLE	CIM server or database not available
36	STORMAN_AUTH_FAILED	User / password for StorMan not valid

14.6.3.2 storcfg system –discover

Storman starts a discovery of the storage system (i.e. the complete configuration data of the specified storage system is updated).
The discovery is processed by the appropriate CIM-OM or API-Server in StorMan's configuration running in Proxy mode. This function may take some time depending on the complexity of the storage system.
Please note that this function is NOT required and not supported for CIM-OM running embedded directly on the storage systems except if –full is specified to update the StorMan configuration data

Syntax

```
storcfg system –discover –system systen [-full]  –connectparam ...
```

Parameters

–system *system*

> Specifies the system name of the storage system to discover. This parameter is mandatory and does **not** support wildcards (*).

–full

> Discovers or re-descovers the storage system and perform a complete update of the configuration data.

–connectparam

> Refer to the description of the "Connection parameters".

Return codes

Error Code	Name	Error Type
0	STORMAN_OK	Function successful
3	STORMAN_VER_NOT_SUPP	Version not supported
7	STORMAN_INV_PARAMETERS	Specified parameters invalid
13	STORMAN_COMMUNIC_ERROR	Error in StorMan communication between client and server
16	STORMAN_FAULT	Unexpected error in function
18	STORMAN_NOT_FOUND	Matching object could not be provided
21	STORMAN_FCT_NOT_POSSIBLE	Function not supported
22	STORMAN_RESOURCE_NOT_ AVAILABLE	CIM Server or database not available
25	STORMAN_CIMOM_AUTH_ FAILED	User / password for CIM Server not valid

14.6.3.3 storcfg system –mod

Sets or modifies the access information for the ETERNUS DX storage system to retrieve performance and statistical information.

Syntax

```
storcfg system –mod –system system –statuser statuser –statpwd statpwd –connectparam ...
```

Parameters

–system *system*

Specifies the system name of the storage system to discover. This parameter is mandatory and does **not** support wildcards (*).

–statuser statuser

Specifies the user identification for PMCC access to ETERNUS DX.

–statpwd statpwd

Specifies the password for PMCC access to ETERNUS DX.

–*connectparam*

Refer to the description of the "Connection parameters".

Return codes

Error Code	Name	Error Type
0	STORMAN_OK	Function successful
3	STORMAN_VER_NOT_SUPP	Version not supported
7	STORMAN_INV_PARAMETERS	Specified parameters invalid
13	STORMAN_COMMUNIC_ERROR	Error in StorMan communication between client and server
16	STORMAN_FAULT	Unexpected error in function
18	STORMAN_NOT_FOUND	Matching object could not be provided
21	STORMAN_FCT_NOT_ POSSIBLE	Function not supported

14.6.3.4 storcfg system –show

Displays information related to the storage systems.

Selection criteria for the `-show` function can be specified using the parameters described below.

Syntax

```
storcfg system –show [ –system system ] [ –model model ] [ –name name ] [ –type type ]
                                        [ –cimip cimip ] [ –cimname cimname ]
                    [ –unmanaged ] [ –managed managed ] [ –showstoragepools ]
                    [ –showports ] [ –showraidtypes ] [ –showremote]
                    [-showenvironment] [ –refreshpreferredcimom ] [ –extended ]
                     [ –format format ] –connectparam ...
```

Parameters

–system *system*

> Specifies the system name of the storage system (default: *). Wildcards (*) are supported.

–name *name*

> Specifies the name of the storage system.This may be the customized name given by the administration or the name in the network.

–type *type*

> Select the storage systems by its type.If the parameter –type is omitted no selection is done (default).

> Supported values of –type:

> – disk: deliver Disk Storage (e.g. ETERNUS DX)

> – tape: deliver Tape Storage (e.g. ETERNUS CS)

–model *model*

> Only for –type disk: Specifies the model of the storage system.
> Supported values: eternus | symmetrix | clariion | netappfiler | any (default).

–cimname *cimname*

> Select the storage systems by the name of the managing CIMOM.

–cimip *cimip*

> Select the storage systems by the IP address of the managing CIMOM.

–unmanaged

> Displays only storage systems without an active management instance.

–managed

> Displays only storage systems with an active management instance.

> Permitted values: active | monitor | manual . If the parameter –managed is omitted no selection is done (default).

> active displays storage systems supporting active management (provisioning, replication, …)

> monitor displays storage systems supporting only information functions and monitoring.

> manual displays storage systems only statically configured in StorMan configuration.

–showstoragepools

Displays also the storage pools of the storage systems with `-format std`. The xml output always contains these information.

–showports

Displays also the storage port information of the storage systems with `-format std`. The xml output always contains these information.

–showraidtypes

Displays also the RAID types supported by the storage systems with `-format std`. The xml output always contains these information.

-showremote

Displays all remote connected storage systems to the storage system specified by parameter –system.

–showenvironment

Displays also the physical component information (e.g. controller, power supplies, fans, etc.) with `-format std`. The xml output always contains these information.

–refreshpreferredcimom

Refreshes the preferred CIMOM information and configuration for the selected storage system(s) before displaying the output data.

–extended

Displays extended logical and physical information about the storage system including FRU (Field Replaceable Unit) information. Two input modes are supported:

1. If only the `extended` parameter is specified:
   Displays an information overview about all storage systems with available extended information.

2. If the `extended` parameter and `–system` *system* are specified:
   Displays the complete and detailed information for the specified storage system.

All other parameters are rejected if the `extended` parameter is specified. If `–system` is specified input is supported as IP address only.

This option is currently supported for ETERNUS DX and CLARiiON storage systems.

–format *format*

Specifies the output format. Permitted values: `std` (default) | `short` | `medium` | `xml` | `xmlp`.

`std` selects a multi-line output: Each attribute of an object is displayed on a separate line. This is also the default output format.

`short` displays only the id / name of an object on a separate line.

`medium` displays the id / name and some basic properties of an object on a separate line.

`xml` writes the output in xml format in a single line.

`xmlp` writes the output in xml format, but in a multi-line format for better human readability (pretty print).

–*connectparam*

Refer to the description of the "Connection parameters".

Return codes

Error Code	Name	Error Type
0	STORMAN_OK	Function successful
3	STORMAN_VER_NOT_SUPP	Version not supported
16	STORMAN_FAULT	Unexpected error in function
18	STORMAN_NOT_FOUND	Matching object could not be provided
22	STORMAN_RESOURCE_NOT_ AVAILABLE	CIM Server or database not available
25	STORMAN_CIMOM_AUTH_FAILED	User / password for CIM Server not valid
36	STORMAN_AUTH_FAILED	User / password for StorMan not valid

Examples

1. Output for storage systems with `-format=std` (default)

```
storcfg system -show -model eternus -showports -showstoragepools
              -showraidtypes

storage system name = Eternus+4621347002
  serial number      = 4621347002
  storage vendor     = Fujitsu
  storage model      = Eternus
  storage model name = ETERNUS DX500 S3
  version            = V10L20-1000
  cache size         = 64 GB
  physical disks     = 48
  logical volumes    = 749
  masking enabled    = yes
  preferred CIMOM     = 172.17.67.121
  configuration state = Ok
  storage port(s)     = 500000E0DA804720/FC
                          PortName          = FCP_CM00CA00P00
                          masking enabled   = yes
                          speed             = 4GB
                          status            = OK
                        500000E0DA804721/FC
                          PortName          = FCP_CM00CA00P01
                          masking enabled   = yes
                          speed             = 8GB
                          status            = OK
                        500000E0DA804722/FC
                          PortName          = FCP_CM00CA00P02
                          masking enabled   = yes
                          speed             = 8GB
                          status            = OK
                        500000E0DA804723/FC
                          PortName          = FCP_CM00CA00P03
                          masking enabled   = yes
                          speed             = 8GB
                          status            = OK
                 ….
             500000E0DA804737/FC
                          PortName          = FCP_CM01CA01P03
                          masking enabled   = yes
                          speed             = 4GB
                          status            = OK
    storage pool(s)     = Primordial Storage pool for FUJITSU storage system
                          type              = primordial
```

```
                                status          = Ok
                                raid type       = 0
                                enabled size    = 37908 GB
                                free size       = 4919 GB
                                percent full    = 87
                            RG900_00
                                type            = normal
                                status          = Ok
                                raid type       = RAID1+0
                                enabled size    = 2458 GB
                                free size       = 8699 MB
                                percent full    = 99
…
                            LowPool_0
                                type            = normal
                                status          = Ok
                                raid type       = RAID1
                                enabled size    = 915 GB
                                free size       = 0 KB
                                percent full    = 100
                            MiddlePool_0
                                type            = normal
                                status          = Ok
                                raid type       = RAID1
                                enabled size    = 819 GB
                                free size       = 0 KB
                                percent full    = 100
                            HighPool_0
                                type            = normal
                                status          = Ok
                                raid type       = RAID1
                                enabled size    = 365 GB
                                free size       = 0 KB
                                percent full    = 100
                            RAID_GRP_#0
                                type            = normal
                                status          = error
                                raid type       = RAID1
                                enabled size    = 819 GB
                                free size       = 819 GB
                                percent full    = 0
                            SDP00
                                type            = snapshots
                                status          = Ok
                                raid type       = 0
                                enabled size    = 200 GB
                                free size       = 200 GB
                                percent full    = 0
…
  supp. raid type(s)  = UNPROTECTED
                        RAID0
                        RAID1
                        RAID1+0
                        RAID5
                        RAID5+0
                        RAID6
                        RAID0_TPP
                        RAID1_TPP
                        RAID1+0_TPP
                        RAID5_TPP
                        RAID6_TPP
```

2. Output for storage systems with `-format=medium`

storcfg system -show –model eternus –format medium

```
Eternus+4621347002 model=ETERNUS DX500 S3 OS-version=V10L20-10000
```

3. Output for storage systems `-format=xmlp`

storcfg system -show –model eternus -showraidtypes –format xmlp

```
<Response>
  <ReturnCode>
    <Code>STORMAN_OK</Code>
    <CodeNumber>0</CodeNumber>
    <Text>successful</Text>
    <SubCode>0</SubCode>
    <SubCodeText>STORMAN_SUB_NONE</SubCodeText>
    <MessageID>4</MessageID>
  </ReturnCode>
  <Result>
    <StorageSystems>
      <StorageSystem>
        <StorageSystemName>Eternus+4621347002</StorageSystemName>
        <StorageSerialNumber>4621347002</StorageSerialNumber>
        <StorageSystemID>DX000E220047</StorageSystemID>
        <StorageVendor>STORMAN_STORAGE_VENDOR_FUJITSU</StorageVendor>
        <Vendor>FUJITSU</Vendor>
        <StorageModel>STORMAN_STORAGE_MODEL_ETERNUS</StorageModel>
        <EternusModelType>STORMAN_ETERNUS_MODEL_DX500_S3</EternusModelType>
        <EternusModelTypeCode>14</EternusModelTypeCode>
        <StorageModelName>ETERNUS DX500 S3</StorageModelName>
        <StorageType>STORMAN_STORAGE_TYPE_DISK</StorageType>
        <LinkUI>172.17.67.121</LinkUI>
        <DataProviderType>STORMAN_STORAGE_PROVIDER_TYPE_INTERNAL</DataProviderType>
        <StorageWWNN>500000E0DA804700</StorageWWNN>
        <RemoteSAP>00ETERNUSDXMS3ET503SAU####0J4621347002##</RemoteSAP>
        <OSName/>
        <OSVersion>V10L20-1000</OSVersion>
        <CacheSize>68719476736</CacheSize>
        <CacheSizeNormalized>64 GB</CacheSizeNormalized>
        <NumOfPhysDisks>48</NumOfPhysDisks>
        <NumOfVolumes>749</NumOfVolumes>
        <NumberOfSpareDevices>2</NumberOfSpareDevices>
        <NumberOfUnmangedVolumes>749</NumberOfUnmangedVolumes>
        <MaskingEnabled>STORMAN_BOOL_TRUE</MaskingEnabled>
        <PrimaryHostAccessController/>
        <ConfigID>15034+47</ConfigID>
        <CopyIndicationCount>69737</CopyIndicationCount>
        <ConfigState>BOX_CONFIG_STATE_OK</ConfigState>
        <PreferredCim>172.17.67.121</PreferredCim>
        <EnclosureCount>3</EnclosureCount>
        <Product>ETERNUSDXMS3(ET503SAU)</Product>
        <Contact>Werner</Contact>
        <Location>ABG DC_6a Rack 168</Location>
        <CustomName>DX500_S3-01</CustomName>
        <Version>V10L20-1000</Version>
        <SCSIVendor>FUJITSU</SCSIVendor>
        <TotalManagedSpace>40703405064192</TotalManagedSpace>
        <TotalManagedSpaceNormalized>37908 GB</TotalManagedSpaceNormalized>
        <RemainingManagedSpace>5281736032256</RemainingManagedSpace>
        <RemainingManagedSpaceNormalized>4919 GB</RemainingManagedSpaceNormalized>
        <GUID/>
        <SupportsReplication>true</SupportsReplication>
        <IsUnifiedStorage>false</IsUnifiedStorage>
```

```
            <StatUser/>
            <OwningServer/>
            <EditableFields>
              <EditableField>StatUser</EditableField>
              <EditableField>StatPassword</EditableField>
            </EditableFields>
            <RemoteCopyModes>
              <RemoteCopyMode>sync</RemoteCopyMode>
              <RemoteCopyMode>async</RemoteCopyMode>
            </RemoteCopyModes>
            <SupportedRaidTypes>
              <SupportedRaidType>UNPROTECTED</SupportedRaidType>
              <SupportedRaidType>RAID0</SupportedRaidType>
              <SupportedRaidType>RAID1</SupportedRaidType>
              <SupportedRaidType>RAID1+0</SupportedRaidType>
              <SupportedRaidType>RAID5</SupportedRaidType>
              <SupportedRaidType>RAID5+0</SupportedRaidType>
              <SupportedRaidType>RAID6</SupportedRaidType>
              <SupportedRaidType>RAID0_TPP</SupportedRaidType>
              <SupportedRaidType>RAID1_TPP</SupportedRaidType>
              <SupportedRaidType>RAID1+0_TPP</SupportedRaidType>
              <SupportedRaidType>RAID5_TPP</SupportedRaidType>
              <SupportedRaidType>RAID6_TPP</SupportedRaidType>
            </SupportedRaidTypes>
            <StoragePools>
            </StoragePools>
            <NetappVolumes>
            </NetappVolumes>
            <RemoteSystems>
            </RemoteSystems>
            <StoragePorts>
            </StoragePorts>
            <PhysicalDisks>
            </PhysicalDisks>
            <Controllers>
            </Controllers>
            <Enclosures>
            </Enclosures>
            <BackendControllers>
            </BackendControllers>
            <PCIeFlashModules>
            </PCIeFlashModules>
            <PowerSupplys>
            </PowerSupplys>
            <Batterys>
            </Batterys>
            <Fans>
            </Fans>
            <OperationalStatus>OK</OperationalStatus>
            <OperationalStatusDetailList>
            </OperationalStatusDetailList>
          </StorageSystem>
        </StorageSystems>
      </Result>
    </Response>
```

14.6.4 storcfg pool

`storcfg pool` is used to manage the pools in the StorMan repository.

```
storcfg pool –function –functionparam ... –connectparam ...
```

Functions

The *–function* parameter supplies one of the following functions:

–add

>   adds a new pool to the StorMan repository

–mod

>   modifies the attributes of an existing pool or renames the pool

–rem

>   removes a pool from the StorMan repository

–show

>   displays information related to pools

14.6.4.1 storcfg pool –add

Creates and adds a new pool to the StorMan repository. The pool is defined for a specific storage system already contained in the repository.

After completing the `-add` function, the new pool is created and empty; it contains no logical volumes.

Syntax

```
storcfg pool –add –poolid poolid –system system [ –restype restype ]
                  [ –access access ]  –connectparam ...
```

Parameters

–poolid *poolid*

>   Specifies the ID of the pool. This parameter is mandatory.
>   Refer to the common description of "poolid" in chapter "Common supported value ranges".

>   The pool ID must be unique and is internally not case-sensitive. If the pool ID already exists in the repository, the command is rejected.

–system *system*

>   Specifies the system name of the storage system that contains the pool. If the specified storage system is not contained in the repository, the command is rejected.

>   This parameter is mandatory.

–restype *restype*

>   Specifies the type of resources in this pool, i.e. if dynamic creation of new logical volumes in the storage system is allowed for this pool.

>   `–restype predefined` (default value) allows no dynamic creation

>   `–restype dynamic` allows dynamic creation

–access *access*

Specifies whether volumes in this pool can be attached to one host only at a point in time (−access exclusive) or may be attached to multiple hosts
(default: −access shared).

–connectparam

Refer to the description of the "Connection parameters".

Return codes

Error Code	Name	Error Type
0	STORMAN_OK	Function successful
3	STORMAN_VER_NOT_SUPP	Version not supported
7	STORMAN_INV_PARAMETERS	Specified parameters invalid
13	STORMAN_COMMUNIC_ERROR	Error in StorMan communication between client and server
16	STORMAN_FAULT	Unexpected error in function
18	STORMAN_NOT_FOUND	Matching object could not be provided
19	STORMAN_POOLID_IN_USE	Pool ID is in use
20	STORMAN_ALREADY_EXISTS	Object already exists in the repository
22	STORMAN_RESOURCE_NOT_ AVAILABLE	CIM Server or database not available
36	STORMAN_AUTH_FAILED	User / password for StorMan not valid

14.6.4.2 storcfg pool –mod

Modifies the attributes of a specified existing pool or renames the pool.
If a parameter is not specified, the corresponding attribute is not changed.

Syntax

```
storcfg pool −mod −poolid poolid [ −newpoolid poolid ] [ −newrestype restype ]
                 [ −newaccess access ]  −connectparam ...
```

Parameters

–poolid *poolid*

Specifies the ID of the pool. This parameter is mandatory.
Refer to the common description of "poolid" in chapter "Common supported value ranges".

The pool ID must be unique and is internally not case-sensitive.

–newpoolid *poolid*

Specifies the new ID of the pool in the case of renaming.

For restrictions in naming the pool ID, refer to the common description of "poolid" in chapter "Common supported value ranges".

–newrestype *restype*

Specifies if dynamic creation of logical volumes is allowed for this pool:

−newrestype predefined allows no dynamic creation.

−newrestype dynamic allows dynamic creation.

–newaccess *access*

> Specifies whether volumes in this pool can be attached to one host only at a point in time (–newaccess exclusive) or may be attached to multiple hosts (default: –newaccess shared).

–*connectparam*

> Refer to the description of the "Connection parameters".

Return codes

Error Code	Name	Error Type
0	STORMAN_OK	Function successful
3	STORMAN_VER_NOT_SUPP	Version not supported
7	STORMAN_INV_PARAMETERS	Specified parameters invalid
13	STORMAN_COMMUNIC_ERROR	Error in StorMan communication between client and server
16	STORMAN_FAULT	Unexpected error in function
18	STORMAN_NOT_FOUND	Specified pool was not found
19	STORMAN_POOLID_IN_USE	Pool ID already used otherwise
22	STORMAN_RESOURCE_NOT_ AVAILABLE	CIM Server or database not available
36	STORMAN_AUTH_FAILED	User / password for StorMan not valid

14.6.4.3 storcfg pool –rem

Removes a pool from the StorMan repository.

If the pool contains logical volumes, these are also removed from the StorMan repository.

Therefore the contents of the pool (logical volumes with their assigned StorIDs) can no longer be managed by StorMan.

The configuration of the storage system and its assignment to servers is not touched by this function.

Syntax

```
storcfg pool –rem –poolid poolid   –connectparam ...
```

Parameters

–poolid *poolid*

> Specifies the ID of the pool to be removed. This parameter is mandatory.
> Refer to the common description of "poolid" in chapter "Common supported value ranges".
> Wildcards (*) are supported.

Return codes

Error Code	Name	Error Type
0	STORMAN_OK	Function successful
3	STORMAN_VER_NOT_SUPP	Version not supported
7	STORMAN_INV_PARAMETERS	Specified parameters invalid
13	STORMAN_COMMUNIC_ERROR	Error in StorMan communication between client and server
16	STORMAN_FAULT	Unexpected error in function
18	STORMAN_NOT_FOUND	Matching object could not be provided, i.e. pool ID not found
19	STORMAN_POOLID_IN_USE	Pool ID already used otherwise
22	STORMAN_RESOURCE_NOT_ AVAILABLE	CIM Server or database not available
36	STORMAN_AUTH_FAILED	User / password for StorMan not valid

14.6.4.4 storcfg pool –show

Displays information related to pools.

Syntax

```
storcfg pool –show [ –poolid poolid ] [ –system system ] [ –restype restype ]
                   [ –model model ] [ –access access ]
                   [ –format format ] –connectparam ...
```

Parameters

–poolid *poolid*

> Specifies the ID of the pool (default: *). Wildcards (*) are supported.

–system *system*

> Specifies the system name of the storage system (default: *).
> Wildcards (*) are supported.

–restype *restype*

> Selects by the type of resources in a pool, i.e. if dynamic creation of new logical volumes in the storage system is allowed for the pools.
> Supported values: predefined | dynamic | any (default)

–model *model*

> Selects by the model of the storage system.
> Supported values: eternus | symmetrix | netappfiler | any (default).

––access *access*

> Selects by the host access to the storage system.
> Supported values: exclusive | shared | any (default).

−`format` *format*

Specifies the output format. Supported values: `std` (default) | `short` | `medium` | `xml` | `xmlp`.

`std` selects a multi-line output: Each attribute of an object is displayed on a separate line. This is also the default output format.

`short` displays only the id / name of an object on a separate line.

`medium` displays the id / name and some basic properties of an object on a separate line.

`xml` writes the output in xml format in a single line.

`xmlp` writes the output in xml format, but in a multi-line format for better human readability (pretty print).

−*connectparam*

Refer to the description of the "Connection parameters".

Return codes

Error Code	Name	Error Type
0	STORMAN_OK	Function successful
3	STORMAN_VER_NOT_SUPP	Version not supported
7	STORMAN_INV_PARAMETERS	Specified parameters invalid
13	STORMAN_COMMUNIC_ERROR	Error in StorMan communication between client and server
16	STORMAN_FAULT	Unexpected error in function
22	STORMAN_RESOURCE_NOT_ AVAILABLE	CIM Server or database not available
36	STORMAN_AUTH_FAILED	User / password for StorMan not valid

Examples

1. Output for pools with `-format std` (default)

   storcfg pool −show −poolid SHC-OSD*

   ```
   pool ID             = SHC−OSD−DX−4541142001
      storage system name = Eternus+4541142001
      storage vendor      = Fujitsu
      storage model       = Eternus
      resource type       = predefined
      allowed access      = shared
   ```

2. Output for pools with `-format medium`

   storcfg pool −show −poolid SHC-OSD * -format medium

SHC-OSD-DX-4541142001 storage=Eternus+4541142001

3.   Output for pools with `-format xmlp`

storcfg pool –show –poolid SHC-OSD* -format xmlp

```
<Response>
  <ReturnCode>
    <Code>STORMAN_OK</Code>
    <CodeNumber>0</CodeNumber>
    <Text>successful</Text>
    <SubCode>0</SubCode>
    <SubCodeText>STORMAN_SUB_NONE</SubCodeText>
    <MessageID>115410</MessageID>
  </ReturnCode>
  <Result>
    <PoolDescriptions>
      <PoolDescription>
        <PoolID>SHC-OSD-DX-4541142001</PoolID>
        <StorageSystemName>Eternus+4541142001</StorageSystemName>
        <StorageVendor>STORMAN_STORAGE_VENDOR_FUJITSU</StorageVendor>
        <StorageModel>STORMAN_STORAGE_MODEL_ETERNUS</StorageModel>
        <ResourceType>STORMAN_RESOURCE_TYPE_PREDEFINED</ResourceType>
        <AllowedAccess>STORMAN_ACCESS_MODE_ANY</AllowedAccess>
        <NumVolumes>2457</NumVolumes>
        <NumPossibleTargetVolumes>2184</NumPossibleTargetVolumes>
        <NumPossibleTargetSnaps>260</NumPossibleTargetSnaps>
      </PoolDescription>
    </PoolDescriptions>
  </Result>
</Response>
```

14.6.5 storcfg volume

`storcfg volume` is used to manage the logical volumes in the pools of the StorMan repository.

`storcfg volume –function –functionparam ... –connectparam ...`

Functions

The *–function* parameter supplies one of the following functions:

–add

   adds new logical volumes to a specified pool of the StorMan repository

–mod

   changes the attributes of logical volumes of the pool or moves a logical volume into another pool

–rem

   removes a logical volume from the pool of StorMan repository

–discover

   requests to discover a logical volume on a storage system, i.e. to update the configuration data of the specified logical volume

–show

   displays information related to logical volumes

14.6.5.1 storcfg volume –add

Adds one or more new logical volumes to a specified pool of the StorMan repository. The pool must already exist in the repository. The logical volumes must be part of the same storage system as the pool.

The function supports the addition of single logical volumes to a pool or a mass operation adding all logical volumes of a storage system not yet contained in the StorMan repository.

Syntax

```
storcfg volume –add [ –storid storid ] –poolid poolid [–system system]
                    [–prefix prefix] [-startnumber startnumber]
                     –deviceid devid  –connectparam ...
```

Parameters

–storid *storid*

> Default: Parameter not specified.
>
> If the parameter is not specified (or no value is specified), there are two cases:
>
> − A storID is assigned automatically by StorMan if the logical volume is already attached to a server.
> − No storID is assigned if the logical volume is currently not attached to any server. It is considered as a free volume in the pool. This volume can be requested by a subsequent storcreate command.
>
> If the parameter is specified with a storID, the value storid specifies the unique identifier to be assigned to the logical volume for further management by StorMan which is selectable by the caller.
>
> It is possible to specify a list of storIDs separated by comma. The number of storIDs must equal the number of deviceIDs specified.
>
> − For naming conventions related to storIDs, refer to the common description of "storid" in chapter "Common supported value ranges".
> − If the storID already exists, the command is rejected.
> − If specified by caller, the storID is assigned regardless of the attachment of the logical volume to a server.
>
> If the value #auto is specified, StorMan automatically generates a storID for all logical volumes regardless of any attachment to a server.
>
> If the value #byrule is specified, StorMan automatically generates a storID for logical volumes (list of deviceids supported as defined by parameters –prefix and –startnumber regardless of any attachment to a server. If –deviceid #all is specified, all deviceIDs not yet assigned to a pool are added. Parameter –prefix must be specified.

–poolid *poolid*

> Specifies the ID of the pool the logical volume should be added to. This parameter is mandatory. If the pool ID does not exist in the repository, the command is rejected.

–system *system*

> Specifies the name of the storage system.
>
> This parameter is optional. It can be specified to uniquely define the volume in combination with –deviceid *devid*. If not specified the storage system defined by parameter –poolid is used.

–prefix *prefix*

> This parameter is only supported for –storid #byrule.
>
> –prefix defines a common prefix for the automatically generated storIDs of the deviceids specified in the call. The prefix must be conform with the naming conventions for the storID.
>
> If the parameter –startnumber is not specified the storIDs are automatically generated from the specified prefix and the complete deviceid used as suffix.

e.g. call with:

```
–prefix ABC_  creates storIDs: ABC_devid1, ABC_devid2,….,ABC_devidn
```

–startnumber *startnumber*

This parameter is only supported for `–storid #byrule` and `–prefix` *prefix*.

If specified the automatically generated storIDs are built by the prefix specified with `–prefix` and an incremented number as suffix starting with the number specified with `–startnumber` (with leading zeros) for all deviceids specified in the call.

e.g. call with:

```
–prefix ABC_ -startnumber 0022 creates storIDs: ABC_0022, ABC_0023,….,ABC_00nm
```

The specified startnumber is a decimal number greater or equal 0.

If the assigned number exceeds the size of the number of digits specified with –startnumber, including leading zeroes, the number of digits incremented for the numbers exceeding this size.

–deviceid *devid*

Specifies the device ID(s) of the logical volume. This parameter is mandatory. Possible values:

− The value is interpreted as a string (not an integer value, i.e. `1` is different from `0001`) and must be equal to the value shown by StorMan via `storcfg volume -show -system` *system* `-storid`. Otherwise `STORMAN_NOT_FOUND` is returned.

− It is possible to specify a list of deviceIDs separated by comma. If storIDs are specified the number of deviceIDs must be equal to the number of storIDs specified.

− If the value `#all` is specified, all logical volumes of the storage system not yet contained in the StorMan repository are added. In this case the parameter `-storid #auto` or `-storid #byrule` must be specified to automatically assign storIDs to the volumes. It is not possible to specify storIDs directly or by list in this case.

–*connectparam*

Refer to the description of the "Connection parameters".

Return codes

Error Code	Name	Error Type
0	STORMAN_OK	Function successful
3	STORMAN_VER_NOT_SUPP	Version not supported
7	STORMAN_INV_PARAMETERS	Specified parameters invalid
9	STORMAN_STORID_IN_USE	StorID already used otherwise
13	STORMAN_COMMUNIC_ERROR	Error in StorMan communication between client and server
14	STORMAN_STORAGE_LOCKED	Storage unit to be modified currently locked
16	STORMAN_FAULT	Unexpected error in function
18	STORMAN_NOT_FOUND	Matching object could not be provided
20	STORMAN_ALREADY_EXISTS	Object already exists in the repository
22	STORMAN_RESOURCE_NOT_ AVAILABLE	CIM Server or database not available
25	STORMAN_CIMOM_AUTH_ FAILED	User / password for CIM Server not valid
36	STORMAN_AUTH_FAILED	User / password for StorMan not valid

14.6.5.2 storcfg volume –mod

Changes attributes of logical volumes in a pool (e.g. rename) or moves a logical volume from a pool to another pool.

If a parameter is not specified, the corresponding attribute is not changed.

Syntax

```
storcfg volume –mod { [ –storid storid ] [ –deviceid devid ] }
                     [ –system system ] [ –newpoolid poolid ]
                     [ –newstorid storid ]  –connectparam ...
```

Parameters

–storid *storid*

> This parameter is mandatory for logical volumes with an assigned storID.
>
> For logical volume without assigned storID, the parameter must be omitted and the parameter –deviceid must be specified instead.

–deviceid *devid*

> Specifies the device ID of the logical volume. This parameter is mandatory if the logical volume does not have an assigned storID.
>
> It may be specified additionally for logical volumes with an assigned storID. If both storID and device ID are specified, the input must be consistent.

–system *system*

> Specifies the name of the storage system. This parameter must be specified if the –deviceid parameter is used to specify the volume.

–newpoolid *poolid*

> Specifies the target pool to which the logical volume should be moved by its pool ID. The current pool is implicitly specified by the storID or the device ID of the logical volume.
> Both pools must belong to the same storage system.

−newstorid *storid*

> If specified, renames the storID of the logical volume to the specified value.
> Refer to the common description of "storid" in chapter "Common supported value ranges".

–connectparam

> Refer to the description of the "Connection parameters".

Return codes

Error Code	Name	Error Type
0	STORMAN_OK	Function successful
3	STORMAN_VER_NOT_SUPP	Version not supported
7	STORMAN_INV_PARAMETERS	Specified parameters invalid
9	STORMAN_STORID_IN_USE	StorID already used otherwise
13	STORMAN_COMMUNIC_ERROR	Error in StorMan communication between client and server
14	STORMAN_STORAGE_LOCKED	Storage unit to be modified currently locked
16	STORMAN_FAULT	Unexpected error in function
18	STORMAN_NOT_FOUND	Matching object (Storage Volume) could not be provided
20	STORMAN_ALREADY_EXISTS	Object already exists in the repository
22	STORMAN_RESOURCE_NOT_ AVAILABLE	CIM Server or database not available
25	STORMAN_CIMOM_AUTH_ FAILED	User / password for CIM Server not valid
36	STORMAN_AUTH_FAILED	User / password for StorMan not valid

14.6.5.3 storcfg volume –rem

Removes a logical volume from a pool of StorMan repository.

Syntax

```
storcfg volume -rem  [ -storid storid ] { [ -deviceid devid ] }
                     [ -system system ] -connectparam ...
```

Parameters

−storid *storid*

> This parameter is mandatory for logical volumes with an assigned storID. Wildcards (*) are supported.
>
> For logical volume without assigned storID, the parameter must be omitted and the −deviceid parameter must be specified instead.

−deviceid *devid*

> Specifies the device ID of the logical volume. This parameter is mandatory if the logical volume does not have an assigned storID.
>
> The logical volume specified by its device ID must already be contained in a pool.

−system *system*

> Specifies the name of the storage system. This parameter must be specified if the −deviceid parameter is used to specify the volume.

–connectparam

> Refer to the description of the "Connection parameters".

Return codes

Error Code	Name	Error Type
0	STORMAN_OK	Function successful
3	STORMAN_VER_NOT_SUPP	Version not supported
7	STORMAN_INV_PARAMETERS	Specified parameters invalid
9	STORMAN_STORID_IN_USE	StorID already used otherwise
13	STORMAN_COMMUNIC_ERROR	Error in StorMan communication between client and server
14	STORMAN_STORAGE_LOCKED	Storage unit to be modified currently locked
16	STORMAN_FAULT	Unexpected error in function
18	STORMAN_NOT_FOUND	Matching object could not be provided
20	STORMAN_ALREADY_EXISTS	Object already exists in the repository
22	STORMAN_RESOURCE_NOT_ AVAILABLE	CIM Server or database not available
36	STORMAN_AUTH_FAILED	User / password for StorMan not valid

14.6.5.4 storcfg volume –discover

Starts a discovery call to the storage system for a specific logical volume to update its configuration and attributes. The specified volume(s) must be contained in a pool.

Syntax

```
storcfg volume –discover [ –storid storid ] [ –system system ]
                         [ –deviceid devid ] –connectparam ...
```

Parameters

–storid *storid*

> This parameter is used for logical volumes with an assigned storID.

> For logical volumes without assigned storID, the parameter must be omitted and the parameter –deviceid must be specified instead.

–system *system*

> Specifies the name of the storage system. This parameter must be specified if the –deviceid parameter is used to specify the volume.

–deviceid *devid*

> Specifies the device ID of the logical volume. This parameter is mandatory if the logical volume does not have an assigned storID.

> It may be specified additionally for logical volumes with an assigned storID. If both storID and device ID are specified, the input must be consistent.

–*connectparam*

> Refer to the description of the "Connection parameters".

Return codes

Error Code	Name	Error Type
0	STORMAN_OK	Function successful.
3	STORMAN_VER_NOT_SUPP	Version not supported.
7	STORMAN_INV_PARAMETERS	Specified parameters invalid.
9	STORMAN_STORID_IN_USE	StorID already used otherwise
13	STORMAN_COMMUNIC_ERROR	Error in StorMan communication between client & server
14	STORMAN_STORAGE_LOCKED	Storage unit to be discovered currently locked.
16	STORMAN_FAULT	Unexpected error in function.
18	STORMAN_NOT_FOUND	Matching object (Storage Volume) could not be provided.
20	STORMAN_ALREADY_EXISTS	Object already exists in the repository
22	STORMAN_RESOURCE_NOT_ AVAILABLE	CIM Server or database not available
25	STORMAN_CIMOM_AUTH_ FAILED	User / password for CIM Server not valid.
36	STORMAN_AUTH_FAILED	User / password for StorMan not valid

14.6.5.5 storcfg volume –show

Displays information related to logical volumes.

Selection criteria can be specified using the function parameters.

Syntax

```
storcfg volume –show [ –storid storid ] [ –poolid poolid ] [ –system system ]
                      [ –systemname systemname ]
                      [ –deviceid devid ] [ –vendor vendor ] [ –model model ] [ –type type ]
                      [ –restype restype ] [ –minsize minsize ] [ –maxsize maxsize ]
                      [ –hostname hostname ] [ –hostip ip ] [ –hostwwpn wwpn ]
                      [ –hostiqn iqn ]  [ –hostlun lun ] [ –access access ]
                      [ –hostaccess hostaccess ] [ –withunmanaged]
                      [ –showconnections] [ –showmirrors]
                      [ –mirrortype mirrortype] [ –replica replica] [ –mirrorusage mirrorusage]
                      [ –targetostype ostype ] [ –raidtype raidtype]
                      [ –format format ] –connectparam ...
```

Parameters

–storid *storid*

Specifies the storID for the logical volume (default: *). Wildcards (*) are supported.

To display all logical volumes of a storage system (even if they are not managed by StorMan) specify –storid #all and exactly one storage system using the system parameter. All other parameters are ignored in this case.

–poolid *poolid*

Specifies the ID of a pool the logical volume belongs to (default: *).
Wildcards (*) are supported.
–poolid #none displays all volumes of a storage system that are NOT assigned to any pool for the specifid system.

–system *system*

Specifies the system name of the storage system the logical volume belongs to (default: *). Wildcards (*) are supported.

−systemname *systemname*

Specifies the (custom) name of the storage system the logical volume belongs to (default: *).

−deviceid *devid*

Specifies the device ID of the logical volume. (default: *). Wildcards (*) are supported.

−vendor vendor

Specifies the vendor of the storage system.

Supported values: fujitsu | emc | netapp | any (default).

−model *model*

Selects by the model of the storage system.
Supported values: eternus | symmetrix || any (default).

-type type

Selects by the type of logical volume in the storage system (currently supported for ETERNUS DX only).

Supported values: –type. (default: no selection)

normal displays normal volumes.

thin displays thin provisioned volumes.

snap displays snap volumes used for snapshots.

flexible displays flexible volumes used for automated storage tiering. −restype *restype*

Selects by the resource type of logical volume from the specified pool.
Supported values: predefined | dynamic | any (default)

−minsize *minsize*

Selects by the minimum size of the logical volume in MB (default: no restriction).

−maxsize *maxsize*

Selects by the maximum size of the logical volume in MB (default: no restriction).

−hostname *hostname*

Selects by the name of the host the storID of is attached to (default: *).
Wildcards (*) are supported.

−hostip *ip*

Selects by the IP address of the host the storID of is attached to (default: *).
Wildcards (*) are supported.

−hostwwpn *wwpn*

Selects by the WWPN of the host port the storID of is attached to (default: *).
Wildcards (*) are supported.

−hostiqn *iqn*

Selects by the IQN of the host port the storID of is attached to (default: *).
Wildcards (*) are supported.

−hostlun *lun*

Selects by the device number of the host the storID of is attached to.
If −showconnections is specified only this connection is displayed.

–`access` *access*

> Selects by the storman setting for multiple host access to the storage volume.
> Supported values: `exclusive` | `shared` | `any` (default).

–`hostaccess` *hostaccess*

> Selects storage volumes according to their visibility to hosts (defined by lun mapping/ masking).
> Supported Values: restricted | allconnected | any (default). See section "LUN masking in Symmetrix systems".

-`withunmanaged`

> Displays also volumes which are not managed by StorMan (i.e. volumes which are not contained in the repository).

–`showconnections`

> Displays also the host connections of the volumes (with output format `std`).

–`showmirrors`

> Displays also assigned mirror volumes (with output format `std`).

–`mirrorusage` *mirrorusage*

> Selects the displayed volumes according to their usage as mirroring (replication) pairs (default: no selection). If the parameter is not specified, no selection of mirror usage is done.
>
> Supported values: `no` | `origlocal` | `mirrorlocal origremote` | `mirrorremote` | `any` (default)
>
> `any` selects all volumes used for mirroring regardless of the kind of mirror usage.
>
> `no` selects only volumes not used for mirroring.
>
> `origlocal` selects all volumes used as original volumes of local mirror pair(s).
>
> `mirrorlocal` selects all volumes used as mirror volumes of local mirror pair(s).
>
> `origremote` selects all volumes used as original volumes of remote mirror pair(s).
>
> `mirrorremote` selects all volumes used as mirror volumes of remote mirror pair(s).

–`mirrortype` *mirrortype*

> Specifies the mirror type of the volume for mirroring (replication) functions.
>
> Supported values: `snap` | `full` | `any` (default)
>
> `any` selects the mirroring feature defined by the specified mirror pair(s).
>
> `snap` selects snapshot (pointer based) mirror pair(s).
>
> `full` selects full copy mirror pairs.

–`replica` *replica*

> Specifies the replica type of the volume for mirroring (replication) functions.
>
> Supported values: `local` | `remote` | `any` (default)
>
> `any` selects the mirroring feature defined by the specified mirror pair(s).
>
> `local` selects local mirror pair(s).
>
> `remote` selects remote mirror pairs.

–`targetostype` *ostype*

> Selects by the operating system type of the provisioned host, for NetApp FAS storage systems only.
> Supported values: `local` | `windows` | `linux` | `solaris` | `any` (default)

−`raidtype` *raidtype*

Selects by the RAID type of the volume (default: any).

`any` selects the volumes regardless of the RAID type. The supported RAID types on a storage system can be shown by using `storcfg system −show −showraidtypes`.

−`format` *format*

Specifies the output format. Permitted values: `std` (default) | `short` | `medium` | `xml` | `xmlp`.

`std` selects a multi-line output: Each attribute of an object is displayed on a separate line. This is also the default output format.

`short` displays only the id / name of an object on a separate line.

`medium` displays the id / name and some basic properties of an object on a separate line.

`xml` writes the output in xml format in a single line.

`xmlp` writes the output in xml format, but in a multi-line format for better human readability (pretty print).

−*connectparam*

Refer to the description of the "Connection parameters".

Return codes

Error Code	Name	Error Type
0	STORMAN_OK	Function successful
3	STORMAN_VER_NOT_SUPP	Version not supported
7	STORMAN_INV_PARAMETERS	Specified parameters invalid
13	STORMAN_COMMUNIC_ERROR	Error in StorMan communication between client and server
14	STORMAN_STORAGE_LOCKED	Storage unit to be modified currently locked
16	STORMAN_FAULT	Unexpected error in function
18	STORMAN_NOT_FOUND	Matching object could not be provided
22	STORMAN_RESOURCE_NOT_ AVAILABLE	CIM Server or database not available

Examples

1. Output for logical volumes with `-format std` (default)

   `storcfg volume -show - storid UID_600000E* −showmirrors`

```
  storage ID         = UID_600000E00D1000000010301C00C10000
    pool ID          = SHC−OSD−DX−4541142001
    storage system name = Eternus+4541142001
    storage serial nr.  = 4541142001
    device ID        = 193
    GUID             = 600000E00D1000000010301C00C10000
    storage size     = 3600 MB
    type             = normal
    resource type    = predefined
    storage vendor   = Fujitsu
    storage model    = Eternus
    allowed access   = unknown
    current access   = shared
    target OS        = unknown
    status           = READY
    raid level       = mirrored
```

```
            raid type           = RAID1
            owner controller    = CM01
            storage pools       = RG03
            host access         = restricted
            host connections
              host ID type  = WWPN
              host ID       = 206800000EA09108
                 initiators
                   initiator ID    = 206800000EA09108
                   connection type = FC
                   storage connections
                     host device number = 1
                       target ID         = 500000E0D4301CA1
                       connection type   = FC
              host ID type  = WWPN
              host ID       = 20EC00000EA095A7
                 initiators
                   initiator ID    = 20EC00000EA095A7
                   connection type = FC
                   storage connections
                     host device number = 1
                       target ID         = 500000E0D4301C92
                       connection type   = FC
              mirror targets
            storage ID           =
              storage system name     = Eternus+4531107003
              pool ID                 =
              device ID               = 329
              GUID                    =
              mirror type             = full
              replication mode        = remote
              copy mode               = sync
              status                  = split
              isSynch                 = no
              percentSynced           = 32
          no mirror sources found
```

2. Output for logical volumes with `-format medium`

storcfg volume -show -poolid SHC-OSD -format medium

```
UID_600000E00D1000000010301C00C00000 storage=Eternus+4541142001 deviceID=192 size=3600
MB
UID_600000E00D1000000010301C00C10000 storage=Eternus+4541142001 deviceID=193 size=3600
MB
UID_600000E00D1000000010301C00C20000 storage=Eternus+4541142001 deviceID=194 size=3600
MB
```

UID_600000E00D1000000010301C00C30000 storage=Eternus+4541142001 deviceID=195 size=3600
MB 3.   Output for logical volumes with `-format xmlp`

storcfg volume -show -storid UID_600000E0* -format xmlp

```
<Response>
  <ReturnCode>
    <Code>STORMAN_OK</Code>
    <CodeNumber>0</CodeNumber>
    <Text>successful</Text>
    <SubCode>0</SubCode>
    <SubCodeText>STORMAN_SUB_NONE</SubCodeText>
    <MessageID>115415</MessageID>
    <TimeID>1407326266028459</TimeID>
  </ReturnCode>
  <Result>
    <VolumeDescriptions>
      <VolumeDescription>
        <StorID>UID_600000E00D1000000010301C00C10000</StorID>
```

```
            <PoolID>SHC-OSD-DX-4541142001</PoolID>
            <StorageSystemName>Eternus+4541142001</StorageSystemName>
            <StorageSerialNumber>4541142001</StorageSerialNumber>
            <StorageCustomName>ABGDX8700-S2-001</StorageCustomName>
            <GUID>600000E00D1000000010301C00C10000</GUID>
            <DeviceID>193</DeviceID>
            <DeviceNumber>193</DeviceNumber>
            <VolumeName>QA_VG011</VolumeName>
            <Size>3774873600</Size>
            <SizeNormalized>3600 MB</SizeNormalized>
            <StorageVendor>STORMAN_STORAGE_VENDOR_FUJITSU</StorageVendor>
            <StorageModel>STORMAN_STORAGE_MODEL_ETERNUS</StorageModel>
            <ResourceType>STORMAN_RESOURCE_TYPE_PREDEFINED</ResourceType>
            <AllowedAccess>STORMAN_ACCESS_MODE_ANY</AllowedAccess>
            <CurrentAccess>STORMAN_ACCESS_MODE_SHARED</CurrentAccess>
            <TargetOS>STORMAN_OS_ANY</TargetOS>
            <Status>STORMAN_VOLUME_STATUS_READY</Status>
            <VolumeType>STORMAN_VOLUME_TYPE_NORMAL</VolumeType>
            <HostAccess>STORMAN_HOST_ACCESS_RESTRICTED</HostAccess>
            <RaidLevel>STORMAN_RAID_LEVEL_1</RaidLevel>
            <RaidType>RAID1</RaidType>
            <CurrOwnerController>CM01</CurrOwnerController>
            <NetappVolumeName/>
            <SpaceConsumedInBox>3774873600</SpaceConsumedInBox>
            <MayBeSource>true</MayBeSource>
            <MayBeTarget>true</MayBeTarget>
            <StoragePoolNames>
               <StoragePoolName>RG03</StoragePoolName>
            </StoragePoolNames>
            <StoragePoolIDs>
               <StoragePoolID>FUJITSU:RSP0002</StoragePoolID>
            </StoragePoolIDs>
            <Type>Unrestricted</Type>
            <HostConnections>
            </HostConnections>
            <MirroringTargets>
            </MirroringTargets>
            <MirroringSources>
            </MirroringSources>
            <TargetPorts>
            </TargetPorts>
            <CustomProperties>
               <BS2CATID/>
               <BS2MN>3401</BS2MN>
               <BS2VSN/>
            </CustomProperties>
            <OperationalStatus>OK</OperationalStatus>
            <OperationalStatusDetailList>
            </OperationalStatusDetailList>
          </VolumeDescription>
        </VolumeDescriptions>
      </Result>
   </Response>
```

14.7 storcreate

Reserves a new storage volume from a pool or creates a new volume in the storage systems as specified by the caller or defined by the attributes of the specified storage pool.

Syntax

```
storcreate -storid storid [ -poolid poolid ] [ -system system ] -size size
          [ -raidtype raidtype] [ -restype restype ]
          [ -access access ] [ -vendor vendor ] [ -model model ] [ -storagepool poolname ]
          [ -timeout timeout ] -connectparam ...
```

Parameters

−storid *storid*

>   Specifies the storID as the unique identifier to be assigned to the new logical volume for further management by StorMan. This parameter is mandatory.

>   Refer to the common description of "storlid" in chapter "Common supported value ranges".

−poolid *poolid*

>   Specifies the ID of a pool the logical volume should be reserved or created for (default: *). Wildcards (*) are supported.

>   If the specified pool ID does not exist in the repository, the command is rejected.

−system *system*

>   Specifies the system name of the storage system the logical volume should be reserved or created for (default: *). Wildcards (*) are supported.

>   If both -poolid and -system are specified, the information must be consistent (the specified pool(s) must be part of the specified storage systems), otherwise the command is rejected.

−size *size*

>   Specifies by the minimal size of the logical volume in MB. This parameter is mandatory.

−raidtype *raidtype*

>   Specifies the RAID type of the logical volume that should be reserved or created as a string that is supported by the specified storage system (default: #any).

>   This parameter should be used only in conjunction with a pool ID or a storage system. The supported RAID types on a storage system can be shown by using storcfg system -show -showraidtypes.

−restype *restype*

>   Specifies the type of logical volume.
>   Supported values: predefined | dynamic | any (default)

−access *access*

>   Specifies the required type of access mode for the requested volume.
>   Supported values: exclusive | shared | any (default).
>   If the value any is specified, the access type of the pools from which a volume could be reserved is ignored.

−vendor *vendor*

>   Specifies the vendor of the storage system.
>   Supported values: fujitsu | emc | netapp | any (default).

–`model` *model*

> Specifies the model of the storage system.
> Supported values: `eternus` | `symmetrix` | `netappfiler` | `any` (default).

–`storagepool` *poolname*

> Supported for ETERNUS DX systems only:
> Specifies the name of a storagepool (Raidgroup) in a storage system, in which the requested volume should be located. If a volume that satisfies the required criteria does not yet exist, StorMan tries to create a new volume in the specified storage pool.
>
> Available storage pools of the storage system can be shown by using `storcfg system –show -showstoragepools`.
>
> The specified storage pool must be suitable for volume creationIn ETERNUS DX systems all pools shown by `storcfg system –show -showstoragepools` are suitable for volume creation, if the free size is greater than the required size for the new volume.

–`timeout` *timeout*

> Specifies the timeout in seconds if a volume is created dynamically. The default value is specified in the `smserver.ini` file. If the value 0 is specified, no timeout will be used.

–connectparam

> Refer to the description of the "Connection parameters".

Return codes

Error Code	Name	Error Type
0	`STORMAN_OK`	Function successful
3	`STORMAN_VER_NOT_SUPP`	Version not supported
7	`STORMAN_INV_PARAMETERS`	Specified parameters invalid
8	`STORMAN_INVALID_STORID`	Specified storID not allowed
9	`STORMAN_STORID_IN_USE`	StorID already use otherwise
11	`STORMAN_NO_STORAGE_ FOUND`	Matching storage could not be provided, i.e. if `–restype dynamic`: not enough space left in the storage system, if `–restype predefined`: no volume with the requested properties (size) available
13	`STORMAN_COMMUNIC_ERROR`	Error in StorMan communication between client and server
14	`STORMAN_STORAGE_LOCKED`	Storage unit to be modified currently locked
16	`STORMAN_FAULT`	Unexpected error in function
18	`STORMAN_NOT_FOUND`	Matching object could not be provided
21	`STORMAN_FCT_NOT_ POSSIBLE`	Function could not be performed
22	`STORMAN_RESOURCE_NOT_ AVAILABLE`	CIM Server or database not available
24	`STORMAN_CFG_CHANGE_ FAILED`	Volume creation was initialized but rejected by the storage system
25	`STORMAN_CIMOM_AUTH_ FAILED`	User / password for CIM Server not valid
36	`STORMAN_AUTH_FAILED`	User / password for StorMan not valid

14.8 stordelete

Releases a reserved device (i.e. one with a storID). If the storage volume is still attached to a host, the command will be rejected.

For a predefined device, normally only the storID will be removed from the StorMan repository. The storage volume may be reserved again by a subsequent `storcreate` command.
A dynamically created device will normally be removed from the StorMan repository and destroyed in the storage system.
This default behavior can be changed with the mode parameter.

Syntax

```
stordelete –storid storid [ –mode mode ] –connectparam ...
```

Parameters

–`storid` *storid*

> Specifies the assigned storID as unique identifier. This parameter is mandatory.

–`mode` *mode*

> Controls the behavior of the function. Supported values: `std` (default) | `keep` | `destroy`

> The default value `std` selects the default behavior as described above.
> If `keep` is specified, the storID is cleared and the device remains in the pool and in the storage system.
> The value `destroy` destroys the device in the storage system and in the pool.

> If `-mode destroy` is set, the volume (logical device) is completely removed and all data stored

> on that volume is lost.

> The deletion of the volume happens asynchronous and StorMan does not wait for completion. If the volume has host connections or is part of a mirror pair, it cannot be destroyed. In this case the `storid` is cleared but `STORMAN_NOT_COMPLETE` is returned.

–*connectparam*

> Refer to the description of the "Connection parameters".

Return codes

Error Code	Name	Error Type
0	STORMAN_OK	Function successful
2	STORMAN_NOT_COMPLETE	Function only partly completed
3	STORMAN_VER_NOT_SUPP	Version not supported
7	STORMAN_INV_PARAMETERS	Specified parameters invalid
10	STORMAN_STORID_NOT_ FOUND	StorID does not yet exist
13	STORMAN_COMMUNIC_ERROR	Error in StorMan communication between client and server
14	STORMAN_STORAGE_LOCKED	Storage unit currently locked
15	STORMAN_STORAGE_ ATTACHED	No release because storage unit still attached
16	STORMAN_FAULT	Unexpected error in function
21	STORMAN_FCT_NOT_ POSSIBLE	Function not supported
22	STORMAN_RESOURCE_NOT_ AVAILABLE	CIM Server or database not available
25	STORMAN_CIMOM_AUTH_ FAILED	User / password for CIM Server not valid
36	STORMAN_AUTH_FAILED	User / password for StorMan not valid

14.9 stordetach

Disconnects one or multiple storage volume(s) from a host using the LUN masking / unmasking capabilities of the storage system. Disconnected storage volumes are no longer visible to the host.

Syntax

```
stordetach –storid storid[,...][,...] | [ –system system] [ –hostname hostname ] [ –hostip ip ]
           [ –hostwwpn wwpn[,...] ] [ –targetwwpn wwpn[,...]
           –connectparam ...
```

Parameters

–storid *storid*[,...]

Specifies the assigned storID(s) of one or a list of volumes, separated by commas as unique identifier. This parameter is mandatory if parameter –system is not specified instead.

–system *system*

Specifies a storage system and can be specified instead of –storid. If specified StorMan will disconnect all volumes of this storage system from the specified server that are managed by StorMan i.e. that are contained in a pool.

If –storid parameter is specified, this parameter is ignored –hostname *hostname*

Specifies name of the server the storage volumes should be detached from. The volumes will be detached from all WWPNs of the specified server.

–hostip *ip*

Specifies the IP address of the server. The volumes will be detached from all WWPNs of the specified server.

–hostwwpn *wwpn*

Specifies the server's HBA WWPN where the device should be detached. This can be used to define single-path-connections. Only for ETERNUS DX storage systems, a list of hostwwpns is supported, refer to chapter "Special Properties of Fujitsu ETERNUS DX storage systems" – "LUN mapping and LUN masking".

–targetwwpn *wwpn*

Supported for ETERNUS DX systems only:
It specifies one or a list of storage system port WWPNs to be disconnected. In conjunction with parameter –hostwwpn it defines the sequence of connections to be detached for the specified volume(s). It cannot be used if the server is specified by its host name or IP adress.

If specified together with –system connections between a server/ hba and a storage system/ storage ports can be removed even if no volumes of the system are attached to the server any more. Only if unmanaged volumes (not contained in a StorMan pool) are connected to the host, the connections are not removed in this case and the command is refused.

This parameter is supported for Symmetrix DMX systems only.
On ETERNUS DX and Symmetrix VMAX systems it is ignored, because the lun masking connection between server and storage is removed implicitly when the last volumes are detached.

–connectparam

Refer to the description of the "Connection parameters".

Notes

1. –storid must be specified to specify the storage unit(s).

2. Exactly one of the parameters –hostname, –hostip or –hostwwpn must be specified to specify the host. If -hostname or -hostip is specified, StorMan must have the necessary information to determine the WWPNs of the host (e.g. supplied with storcfg host in its internal repository) and the storage is attached to all known host ports.

3. If the host specification is redundant, i.e. as a result of –hostname and / or –hostip and / or -hostwwpn, the specified parameters must be consistent.

Return codes

Error Code	Name	Error Type
0	STORMAN_OK	Function successful
1	STORMAN_NO_ACTION	Nothing to do
2	STORMAN_NOT_COMPLETE	Function only partly completed
3	STORMAN_VER_NOT_SUPP	Version not supported
7	STORMAN_INV_PARAMETERS	Specified parameters invalid
10	STORMAN_STORID_NOT_ FOUND	StorID does not yet exist
13	STORMAN_COMMUNIC_ERROR	Error in StorMan communication between client and server
14	STORMAN_STORAGE_LOCKED	Storage unit currently locked
16	STORMAN_FAULT	Unexpected error in function
22	STORMAN_RESOURCE_NOT_ AVAILABLE	CIM Server or database not available
25	STORMAN_CIMOM_AUTH_ FAILED	User / password for CIM Server not valid
31	STORMAN_REFUSED	Error Type –destroyconnection required but other unmanaged volumes attached
36	STORMAN_AUTH_FAILED	User / password for StorMan not valid

14.10 storemc

Provides vendor-specific functions to add / remove EMC storage systems to / from the EMC SMI-S Provider's database. This function can be used to handle CLARiiON CX systems with no FibreChannel connection to the host running the SMI-S Provider (CIM-OM). These so-called "out-of-band" storage systems are not discovered automatically using `storcfg cimom –discover` but have to be added manually using `storemc`.

```
storemc   –function –functionparam ... –connectparam ...
```

Functions

The *–function* parameter supplies one of the following functions:

`–add`

> adds a CLARiiON CX storage system to the StorMan repository and the SMI-S Provider's repository

`–rem`

> removes a CLARiiON CX storage system from an SMI-S Provider's repository

14.10.1 storemc –add

Add a CLARiiON CX storage system to the StorMan repository and the SMI-S Provider's repository. The system is specified by the `–sp` parameter as described below. The specified CIM-OM must be contained in the repository of StorMan and clearly defined by its hostname / ip address and the port (see section "storcfg cimom").

The processing of this functions automatically starts a discover process on the specified CIM-OM to create or update the information about the specified CLARiiON CX storage system. Additionally further storage systems visible to the CIM-OM may be discovered and added to the repository of StorMan.

> **i**
>
> **Note**
>
> If the authorization (userid and/or password) of the storage system has been changed or removed, you have to call `storemc –rem` (see section "storemc –rem") for this system and call `storemc –add` again with the new authorization.

Syntax

```
storemc –add [ –hostname hostname ] [ –hostip ip ] [–http port ] [ –https port ]
            [ –cimuser user ] [ –cimpwd pwd ] –sp spa,spb –cxuser user
            –cxpwd pwd –connectparam ...
```

Parameters

`–hostname hostname`

> Specifies the name of the host the CIM-OM is running on which will be directed to discover the specified storage system.

`–hostip ip`

> Specifies the IP address of the host the CIM-OM is running on.

`–https port`

> Specifies the secure port the CIM-OM is listening on (default: 5989).

`–cimuser user`

> Specifies the user for the CIM-OM connection (default: none). Mandatory if the CIM user requires authentication.

−cimpwd *pwd*

> Specifies the password for the CIM-OM connection (default: none). Mandatory if the CIM user requires authentication.

−sp *spa*,*spb*

> Specifies a list of two netnames or IP addresses of both storage processors (SPs) of the CLARiiON CX array delimited by comma.
> Make sure that the SPs are specified in the correct order: The first name / address must be *spa* and the second *spb*, otherwise the function does not work.

−cxuser *user*

> Specifies a valid userID on the CLARiiON CX (or the CLARiiON CX domain) with administrator or manager privilege.

−cxpwd *pwd*

> Specifies the password for the userID on CLARiiON CX.

−*connectparam*

> Refer to the description of the "Connection parameters".

Return codes

Error Code	Name	Error Type
0	STORMAN_OK	Function successful
7	STORMAN_INV_PARAMETERS	Specified parameters invalid
13	STORMAN_COMMUNIC_ERROR	Error in StorMan communication between client and server
16	STORMAN_FAULT	Unexpected error in function
21	STORMAN_FCT_NOT_ POSSIBLE	Function refused by SMI-S Provider
22	STORMAN_RESOURCE_NOT_ AVAILABLE	CIM Server or database not available
25	STORMAN_CIMOM_AUTH_ FAILED	User / password for CIM Server not valid
36	STORMAN_AUTH_FAILED	User / password for StorMan not valid

i	In High availability configurations i.e. if more than one CIM Server / SMI-S Provider can manage the same storage system, it is necessary to run this command for each CIM Server.

14.10.2 storemc –rem

Removes a CLARiiON CX storage system from an EMC SMI-S Provider's repository. The system is specified by its system name as delivered by the `storcfg system -show` command (e.g. `CLARiiON+ck200024200100`).

Syntax

```
storemc  –rem [ –hostname hostname ] [ –hostip ip ] [–http port ] [ –https port ]
              –system system [ –cimuser user ] [ –cimpwd pwd ]
              –connectparam ...
```

Parameters

`–hostname` *hostname*

> Specifies the name of the host the CIM-OM is running on which the storage system is removed from the SMI-S Provider's repository.

`–hostip` *ip*

> Specifies the IP address of the host the CIM-OM is running on.

`–http` *port*

> Specifies the port the CIM-OM is listening on (default: 5988).

`–https` *port*

> Specifies the secure port the CIM-OM is listening on (default: 5989).

`–system` *system*

> Specifies the system name of the storage system that is contained in the StorMan repository.

`–cimuser` *user*

> Specifies the user for the CIM-OM connection (default: none). Mandatory if the CIM user requires authentication.

`–cimpwd` *pwd*

> Specifies the password for the CIM-OM connection (default: none). Mandatory if the CIM user requires authentication.

–connectparam

> Refer to the description of the "Connection parameters".

If a CIM Server is specified by its hostname as described above, the system will be removed from the repository of this CIM Server only.

If no CIM Server is specified, the storage system will be removed from the repositories of all CIM Servers that actually have access to this system. This command is useful if a storage system is physically removed or if a storage system should be managed by another CIM Server.

If no CIM Server is specified, it is necessary that all CIM Servers related to the system are contained in the StorMan database with matching user and password for die CIM-OM connection (`–cimuser` and `–cimpwd`). If a system should be removed completely from StorMan's repository, use `storcfg system –rem` as described in section "storcfg system".

Return codes

Error Code	Name	Error Type
0	STORMAN_OK	Function successful
1	STORMAN_NO_ACTION	No CIM Server for this system was found.
2	STORMAN_NOT_COMPLETE	If no CIM Host was specified, the remove function could not be performed for all CIM Hosts, e.g. because not all are running.
7	STORMAN_INV_PARAMETERS	Specified parameters invalid
13	STORMAN_COMMUNIC_ERROR	Error in StorMan communication between client and server
16	STORMAN_FAULT	Unexpected error in function
21	STORMAN_FCT_NOT_ POSSIBLE	Function refused by SMI-S Provider
22	STORMAN_RESOURCE_NOT_ AVAILABLE	CIM Server or database not available
25	STORMAN_CIMOM_AUTH_ FAILED	User / password for CIM Server not valid
36	STORMAN_AUTH_FAILED	User / password for StorMan not valid

14.11 storinfo

Provides information about StorMan.

Syntax

```
storinfo [-server server]  –connectparam ...
```

Parameters

–server *server*

Specifies hostname or IP address of the StorMan server where the information is provided from (default: localhost).

–*connectparam*

Refer to the description of the "Connection parameters".

Return codes

Error Code	Name	Error Type
0	STORMAN_OK	Function successful
7	STORMAN_INV_PARAMETERS	Specified parameters invalid
13	STORMAN_COMMUNIC_ERROR	Error in StorMan communication between client and server
16	STORMAN_FAULT	Unexpected error in function
36	STORMAN_AUTH_FAILED	User / password for StorMan not valid

Example

Output for storinfo

```
storinfo

local StorMan version: V6.0
local StorMan build:   6.0.0.1
local StorMan buildDate:   Aug  1 2014 10:18:08
local hostname:   MCH9422N
local hostinfo:   Windows 7 Enterprise Edition 32-bit build 7601 Service Pack 1
```

14.12 stormandb

Saves or restores the StorMan repository. This command is only supported on the StorMan Server.

Syntax

```
stormandb { –dump / –load } [–noserver] –connectparam
```

Parameters

–dump

> Selects the backup function. The StorMan repository will be saved to the folder backup of the StorMan directory.

–load

> Selects the restore function. The StorMan repository will be restored from the backup folder of the StorMan directory.
>
> If this parameter is used to restore the StorMan repository after installing a new version of StorMan, with an empty repository StorMan has to be restarted because the format of the database may have changed.

–noserver

> Setting this parameter the command will work directly on the repository. Make sure there is really no server working with the repository and no concurrent stormandb command is running because else the repository may be corrupted.

–connectparam

> Refer to the description of the "Connection parameters".

Return codes

Error Code	Name	Error Type
0	STORMAN_OK	Function successful
7	STORMAN_INV_PARAMETERS	Specified parameters invalid
13	STORMAN_COMMUNIC_ERROR	Error in StorMan communication between client and server
16	STORMAN_FAULT	Unexpected error in function

14.13 stormirror

Manages volume based replication functions.

Local replication functions are supported as full copy mirrors of a source volume (for ETERNUS DX) or as snapshots (pointer based) local mirrors of a source volume (for ETERNUS DX).

Remote replication functions are supported as full mirrors in synchronuous mode for ETERNUS DX and EMC Symmetrix VMAX systems. Asynchronous mode for remote replication is supported for ETERNUS DX S3 series.

stormirror function	ETERNUS DX -mirror-type / -replica			Symetrix VMAX -mirror-type / -replica		
	full local	snap local	full remote	full local	snap	full remote
create	yes	yes	yes	no	no	yes
modify	no	no	yes	no	no	no
failover	no	no	no	no	no	yes
restart	yes	yes	yes	no	no	yes
restore	yes	yes	no	no	no	yes
suspend	yes	no	yes	no	no	yes
swap	yes	no	yes	no	no	yes
terminate	yes	yes	yes	no	no	yes
show	yes	yes	yes	no	no	yes

```
stormirror –function –functionparam ... –connectparam ...
```

Functions

The *–function* parameter supplies one of the following functions:

–create

creates new local or remote mirror pair(s) for logical volume(s) by establishing a mirroring relationship between source volume(s) and mirror volume(s)

–modify

Modifies the remote copy mode for remote mirror pair(s) from synchronous to asynchronous and back

–failover

performs a failover to remote mirror volumes for remote mirror pair(s)

–restart

restarts local or remote mirroring function for mirror pair(s)

–restore

restores local mirror volume(s) of mirror pair(s) to source volume(s)

–suspend

> suspends mirroring on local or remote mirror pair(s). Afterwards the mirror volume(s) are available for direct access from server.

–swap

> reverses the roles of local or remote mirror pair(s)

–terminate

> terminates local or remote mirroring relationship for mirror pair(s)

–show

> displays information related to local or remote mirror pair(s)

14.13.1 stormirror –create

Creates one or multiple local or remote mirroring pairs by assigning one or multiple mirror volumes to one or multiple source volumes. The mirror volumes can be specified as full mirror copies or local pointer based logical copies (snapshots).
For full mirror copies after establishing the mirroring relationship the synchronization process is started.
For local replication pairs, the source volumes and mirror volumes must all be part of the same storage system.

Syntax

```
stormirror –create –mirrortype mirrortype [ -replica replica ] [ –copymode copymode ]
                   –source storid[,…] –target storid[,…]
                    [ –waitforsync ] –connectparam ...
```

Parameters

–mirrortype *mirrortype*

> Specifies the type of mirror pair(s) to be created.

> Supported values: snap | full

> snap creates snapshot (pointer based) mirror pair(s). Snapshots on ETERNUS DX systems are immediately available and accessible after creation.

> full creates full copy mirror pairs.

–replica *replica*

> Specifies if local or remote mirror pair(s) should be created.

> Supported values: local | remote | bytarget (default)

> local creates local replica(s) in the same storage system.

> remote creates remote replica(s) in a different storage system of the same model. This can be on a different site. This value is supported for –mirrortype full only.

> bytarget selects the replica location defined by the specified mirror pair(s). The automatically selected replica must be the same for all specified mirror pairs.

`-copymode copymode`

>   Defines the copy mode synchronous or asynchronous: `sync | async | asyncstack`

>   The default value (when omitted) is `sync` (synchronous).

>   `-copymode sync` for ETERNUS DX S3 corresponds to sync REC mode.

>   `-copymode async` for ETERNUS DX S3 corresponds to async REC consistent mode.

>   `-copymode asyncstack` for ETERNUS DX S3 corresponds to async REC stack mode.

>   The parameter is only supported for `-replica remote` and for storage systems ETERNUS DX S3.

`–source` *storid*[,...]

>   Specifies the storID(s) of one or a list of source volumes for the mirror pair(s).

>   The number of specified storIDs for the source volumes must be equal to the number of specified storIDs for the target volumes.

`–target` *storid*[,...]

>   Specifies the storID(s) of one or a list of target volumes for the mirror pair(s).

>   The number of specified storIDs for the source volumes must be equal to the number of specified storIDs for the target volumes.

>   The capacity and type of the specified target volumes must match to the prerequisites of the specified mirroring feature of the storage system.

>   If the value *#auto* is specified, StorMan automatically assigns appropriate target volumes to create the mirror pair(s). This function can be supported for `–mirrortype snap` only.

`–waitforsync`

>   Option to wait for the completion of the synchronization process of the newly created mirror pair(s). This function can be supported for `–mirrortype full` only.

–connectparam

>   Refer to the description of the "Connection parameters".

Return codes

Error Code	Name	Error Type
0	STORMAN_OK	Function successful.
1	STORMAN_NO_ACTION	Nothing to do.
2	STORMAN_NOT_COMPLETE	Function only partly completed.
3	STORMAN_VER_NOT_SUPP	Version not supported
7	STORMAN_INV_PARAMETERS	Specified parameters invalid.
8	STORMAN_INVALID_STORID	Specified storID not allowed
10	STORMAN_STORID_NOT_FOUND	storID does not yet exist.
13	STORMAN_COMMUNIC_ERROR	Error in StorMan communication between client and server.
14	STORMAN_STORAGE_LOCKED	Storage unit currently locked.
16	STORMAN_FAULT	Unexpected error in function.
18	STORMAN_NOT_FOUND	No wwpn for the specified host was found in the repository.
22	STORMAN_RESOURCE_NOT_ AVAILABLE	CIM Server or database not available
25	STORMAN_CIMOM_AUTH_ FAILED	User / password for CIM Server not valid.
29	STORMAN_MIRROR_ERROR	Failure of the mirror operation

Error Code	Name	Error Type
35	`STORMAN_SYSTEM_LOCKED`	The ETERNUS DX system is locked for administration
36	`STORMAN_AUTH_FAILED`	User / password for StorMan not valid
37	`STORMAN_CIMOM_ERROR`	CIMOM returned an error

14.13.2 stormirror –modify

Modifies the copy mode of remote mirroring pairs by switching between synchronous mode and asynchronous modes..
This function is currently supported for remote replication as of ETERNUS DX S3.

Syntax

```
stormirror –modify [ –newcopymode newcopymode ] –mirrortype mirrortype [ -replica replica ]
                    –source storid[,...] –target storid[,...]
                     –connectparam ...
```

Parameters

`–copymode copymode`

>   Defines the copy mode synchronous or asynchronous: `sync` | `async` | `asyncstack`

>   The default value (when omitted) is `sync` (synchronous).

>   `–copymode sync` for ETERNUS DX S3 corresponds to sync REC mode.

>   `–copymode async` for ETERNUS DX S3 corresponds to async REC consistent mode.

>   `–copymode asyncstack` for ETERNUS DX S3 corresponds to async REC stack mode.

>   The parameter is only supported for `–replica remote` and for storage systems ETERNUS DX S3.

`–mirrortype mirrortype`

>   Specifies the type of mirror pair(s) to be created.

>   Supported values: `full`

>   `full` creates full copy mirror pairs.

`–replica replica`

>   Specifies if local or remote mirror pair(s) should be created.

>   Supported values: `remote` (default)

>   `remote` creates remote replica(s) in a different storage system of the same model. This can be on a different site. This value is supported for `–mirrortype full` only.

`–source storid[,...]`

>   Specifies the storID(s) of one or a list of source volumes for the mirror pair(s).

>   The number of specified storIDs for the source volumes must be equal to the number of specified storIDs for the target volumes.

−target *storid*[,...]

    Specifies the storID(s) of one or a list of target volumes for the mirror pair(s).

    The number of specified storIDs for the source volumes must be equal to the number of specified storIDs for the target volumes.

−connectparam

    Refer to the description of the "Connection parameters".

Return codes

Error Code	Name	Error Type
0	STORMAN_OK	Function successful.
1	STORMAN_NO_ACTION	Nothing to do.
2	STORMAN_NOT_COMPLETE	Function only partly completed.
3	STORMAN_VER_NOT_SUPP	Version not supported
7	STORMAN_INV_PARAMETERS	Specified parameters invalid.
8	STORMAN_INVALID_STORID	Specified storID not allowed
10	STORMAN_STORID_NOT_ FOUND	storID does not yet exist.
13	STORMAN_COMMUNIC_ERROR	Error in StorMan communication between client and server.
14	STORMAN_STORAGE_LOCKED	Storage unit currently locked.
16	STORMAN_FAULT	Unexpected error in function.
18	STORMAN_NOT_FOUND	No wwpn for the specified host was found in the repository.
22	STORMAN_RESOURCE_NOT_ AVAILABLE	CIM Server or database not available
25	STORMAN_CIMOM_AUTH_ FAILED	User / password for CIM Server not valid.
29	STORMAN_MIRROR_ERROR	Failure of the mirror operation
35	STORMAN_SYSTEM_LOCKED	The ETERNUS DX system is locked for administration
36	STORMAN_AUTH_FAILED	User / password for StorMan not valid
37	STORMAN_CIMOM_ERROR	CIMOM returned an error

14.13.3   stormirror –restart

Restarts the mirroring for the specified local or remote mirror pair(s). The mirror volumes are no longer accessible to the server and/or application and all changes done on the mirror volumes are discarded. For local mirror pairs a subsequent `stormirror –suspend` may activate the mirror volumes again to obtain a new point in time of the mirrored data.

The function is only supported for local and remote full mirror pairs. It is not supported for `–mirrortype snap`.

Syntax

```
stormirror –restart [ –mirrortype mirrortype ] [ -replica replica ]
                     –source storid[,...] –target storid[,...]
                     [ –waitforsync ] –connectparam ...
```

Parameters

–`mirrortype` *mirrortype*

> Specifies the type of mirror pair(s) to be restarted.
>
> Supported values: `full` | `any` (default)
>
> `full` selects full copy mirror pairs.
>
> `any` selects the mirroring feature defined by the specified mirror pair(s).

–replica *replica*

> Specifies if local or remote mirror pair(s) should be processed.
>
> Supported values: `local` | `remote` | `bytarget` (default)
>
> `local` creates local replica(s) in the same storage system.
>
> `remote` creates remote replica(s) in a different storage system of the same model. This can be on a different site. This value is supported for `–mirrortype full` only.
>
> `bytarget` selects the replica location defined by the specified mirror pair(s). The automatically selected replica must be the same for all specified mirror pairs.

–`source` *storid*[,...]

> Specifies the storID(s) of one or a list of source volumes for the mirror pair(s).
>
> The number of specified storIDs for the source volumes must be equal to the number of specified storIDs for the target volumes.

–`target` *storid*[,...]

> Specifies the storID(s) of one ore a list of target volumes for the mirror pair(s).
>
> The number of specified storIDs for the source volumes must be equal to the number of specified storIDs for the target volumes.

–`waitforsync`

> Option to wait for the completion of the re-synchronization process of the mirror pair(s).

–*connectparam*

> Refer to the description of the "Connection parameters".

Return codes

Error Code	Name	Error Type
0	STORMAN_OK	Function successful.
1	STORMAN_NO_ACTION	Nothing to do.
2	STORMAN_NOT_COMPLETE	Function only partly completed.
3	STORMAN_VER_NOT_SUPP	Version not supported
7	STORMAN_INV_PARAMETERS	Specified parameters invalid.
8	STORMAN_INVALID_STORID	Specified storID not allowed
10	STORMAN_STORID_NOT_ FOUND	storID does not yet exist.
13	STORMAN_COMMUNIC_ERROR	Error in StorMan communication between client and server.
14	STORMAN_STORAGE_LOCKED	Storage unit currently locked.
16	STORMAN_FAULT	Unexpected error in function.
18	STORMAN_NOT_FOUND	No wwpn for the specified host was found in the repository.
22	STORMAN_RESOURCE_NOT_ AVAILABLE	CIM Server or database not available
25	STORMAN_CIMOM_AUTH_ FAILED	User / password for CIM Server not valid.
29	STORMAN_MIRROR_ERROR	Failure of the mirror operation
35	STORMAN_SYSTEM_LOCKED	The ETERNUS DX system is locked for administration
36	STORMAN_AUTH_FAILED	User / password for StorMan not valid
37	STORMAN_CIMOM_ERROR	CIMOM returned an error

14.13.4 stormirror –failover

Performs a failover for remote mirror pairs of Symmetrix VMAX storage systems. During the failover the targets are set ready to the server and the sources are write disabled to the server. In result applications can continue to work on the targets instead of the sources. This is important to support HA and DR (Dynamic Reconfiguration) scenarios.

The failover function is supported for remote full mirror pairs on Symmetrix VMAX only.

Syntax

```
stormirror –failover –source storid[,…] –target storid[,…] –connectparam ...
```

Parameters

−`source` *storid*[,...]

    Specifies the storID(s) of one or a list of source volumes for the mirror pair(s).

    The number of specified storIDs for the source volumes must be equal to the number of specified storIDs for the target volumes.

−`target` *storid*[,...]

    Specifies the storID(s) of one or a list of target volumes for the mirror pair(s).

    The number of specified storIDs for the source volumes must be equal to the number of specified storIDs for the target volumes.

−connectparam

    Refer to the description of the "Connection parameters".

Return codes

Error Code	Name	Error Type
0	STORMAN_OK	Function successful.
1	STORMAN_NO_ACTION	Nothing to do.
2	STORMAN_NOT_COMPLETE	Function only partly completed.
3	STORMAN_VER_NOT_SUPP	Version not supported
7	STORMAN_INV_PARAMETERS	Specified parameters invalid.
10	STORMAN_STORID_NOT_ FOUND	storID does not yet exist.
13	STORMAN_COMMUNIC_ERROR	Error in StorMan communication between client and server.
14	STORMAN_STORAGE_LOCKED	Storage unit currently locked.
16	STORMAN_FAULT	Unexpected error in function.
22	STORMAN_RESOURCE_NOT_ AVAILABLE	CIM Server or database not available
25	STORMAN_CIMOM_AUTH_ FAILED	User / password for CIM Server not valid.
29	STORMAN_MIRROR_ERROR	Failure of the mirror operation
35	STORMAN_SYSTEM_LOCKED	The storage system is locked for administration
36	STORMAN_AUTH_FAILED	User / password for StorMan not valid
37	STORMAN_CIMOM_ERROR	CIMOM returned an error

14.13.5 stormirror –restore

Restores one or more mirror pairs. This function restores the contents of the local mirror volumes back to the original volumes of the mirrored pairs. All changes on the original volumes are discarded.
The mirror volumes of mirror type `snap` remain accessible to the server and/or application.
The mirror volumes of mirror type `full` are no longer accessible to the server and/or application. The status of mirror volumes after processing a restore operation depends on the specific storage system.

i	This function is supported for ETERNUS DX systems only for `–copymode sync`.
	For ETERNUS DX systems the restore operation is completed only after synchronization is also completed. The mirror volumes are accessible to the server and the mirror pair are 'split' and 'synchronized'.

Syntax

```
stormirror -restore [ -mirrortype ] ] [ -replica replica ]
                     -source storid[,...] -target storid[,...] -connectparam ...
```

Parameters

–`mirrortype` *mirrortype*

Specifies the type of mirror pair(s) to be restored.

Supported values: `snap` | `full` | `any` (default)

`snap` selects snapshot (pointer based) mirror pair(s).

`full` selects full copy mirror pairs.

`any` selects the mirroring feature defined by the specified mirror pair(s).

–`replica` *replica*

Specifies if local or remote mirror pair(s) should be restored.

Supported values: `local` | `remote` | `bytarget` (default)

`local` creates local replica(s) in the same storage system.

`remote` restores remote replica(s) in a different storage system of the same model. This can be on a different site. This value is supported for `–mirrortype full` only.

`bytarget` selects the replica location defined by the specified mirror pair(s). The automatically selected replica must be the same for all specified mirror pairs.

–`source` *storid*[,...]

Specifies the storID(s) of one or a list of source volumes for the mirror pair(s).

The number of specified storIDs for the source volumes must be equal to the number of specified storIDs for the target volumes.

–`target` *storid*[,...]

Specifies the storID(s) of one or a list of target volumes for the mirror pair(s).

The number of specified storIDs for the source volumes must be equal to the number of specified storIDs for the target volumes.

–connectparam

Refer to the description of the "Connection parameters".

Return codes

Error Code	Name	Error Type

Error Code	Name	Error Type
0	STORMAN_OK	Function successful.
1	STORMAN_NO_ACTION	Nothing to do.
2	STORMAN_NOT_COMPLETE	Function only partly completed.
3	STORMAN_VER_NOT_SUPP	Version not supported
7	STORMAN_INV_PARAMETERS	Specified parameters invalid.
8	STORMAN_INVALID_STORID	Specified storID not allowed
10	STORMAN_STORID_NOT_ FOUND	storID does not yet exist.
13	STORMAN_COMMUNIC_ERROR	Error in StorMan communication between client and server.
14	STORMAN_STORAGE_LOCKED	Storage unit currently locked.
16	STORMAN_FAULT	Unexpected error in function.
18	STORMAN_NOT_FOUND	No wwpn for the specified host was found in the repository.
22	STORMAN_RESOURCE_NOT_ AVAILABLE	CIM Server or database not available
25	STORMAN_CIMOM_AUTH_ FAILED	User / password for CIM Server not valid.
29	STORMAN_MIRROR_ERROR	Failure of the mirror operation. Details in subcodes: STEP_RESTART_FAILED STEP_SUSPEND_FAILED STEP_2ND_SWAP_FAILED
35	STORMAN_SYSTEM_LOCKED	The storage system is locked for administration
36	STORMAN_AUTH_FAILED	User / password for StorMan not valid
37	STORMAN_CIMOM_ERROR	CIMOM returned an error

14.13.6 stormirror –suspend

Suspends the mirroring between source and target volumes of one or more mirrors pairs. The function suspends the mirroring for the specified mirror pair(s).

The local volumes remain accessible to the server and/or application and all changes are tracked by the storage system.

The mirror volumes become available for a server and/or application. The activation of multiple mirror volumes is not supported consistently by means of IO sequence. Therefore consistency must be assured by caller.

The mirror pairs specified by the source and mirror volumes must be homogenous, i.e. they must have the same mirror type and must be specified in the correct order.

The function is supported for local and remote full mirror pairs.

Syntax

```
stormirror -suspend [ –mirrortype mirrortype ] [ -replica replica ]
                 -source storid[,…] –target storid[,…]] [ -force ]
                 –connectparam ...
```

Parameters

–mirrortype *mirrortype*

Specifies the type of mirror pair(s) to be supended.

Supported values: `full` | `any` (default)

`full` selects full copy mirror pairs.

`any` selects the mirroring feature defined by the specified mirror pair(s).

`–replica` *replica*

Specifies if local or remote mirror pair(s) should be processed.

Supported values: `local` | `remote` | `std` (default)

`–source` *storid*[,...]

Specifies the storID(s) of one or a list of source volumes for the mirror pair(s).

The number of specified storIDs for the source volumes must be equal to the number of specified storIDs for the target volumes.

`–target` *storid*[,...]

Specifies the storID(s) of one ore a list of target volumes for the mirror pair(s).

The number of specified storIDs for the source volumes must be equal to the number of specified storIDs for the target volumes.

`–force`

Suspends the mirroring regardless of the status. The function is performed as supported by underlaying functions of the storage system.

–connectparam

Refer to the description of the "Connection parameters".

Return codes

Error Code	Name	Error Type
0	STORMAN_OK	Function successful.
1	STORMAN_NO_ACTION	Nothing to do.
2	STORMAN_NOT_COMPLETE	Function only partly completed.
3	STORMAN_VER_NOT_SUPP	Version not supported
7	STORMAN_INV_PARAMETERS	Specified parameters invalid.
8	STORMAN_INVALID_STORID	Specified storID not allowed
10	STORMAN_STORID_NOT_FOUND	storID does not yet exist.
13	STORMAN_COMMUNIC_ERROR	Error in StorMan communication between client and server.
14	STORMAN_STORAGE_LOCKED	Storage unit currently locked.
16	STORMAN_FAULT	Unexpected error in function.
22	STORMAN_RESOURCE_NOT_AVAILABLE	CIM Server or database not available
25	STORMAN_CIMOM_AUTH_FAILED	User / password for CIM Server not valid.
29	STORMAN_MIRROR_ERROR	Failure of the mirror operation
35	STORMAN_SYSTEM_LOCKED	The storage system is locked for administration
36	STORMAN_AUTH_FAILED	User / password for StorMan not valid
37	STORMAN_CIMOM_ERROR	CIMOM returned an error

14.13.7 stormirror –swap

Swaps the mirror roles of source and target volumes of one or more mirrors pairs. The function reverses the roles of the volumes for the specified mirror pair(s) in the storage system, making the source volumes to target volumes and vice versa. The current local volumes are no longer accessible to the server and/or application and all.

The function is supported for local and remote full mirror pairs (clones).

Syntax

```
stormirror -swap [ -mirrortype mirrortype ] [ -replica replica ]
                  -source storid[,...] -target storid[,...]
                  -connectparam ...
```

Parameters

–mirrortype *mirrortype*

>   Specifies the type of mirror pair(s) to be swapped.

>   Supported values: full | any (default)

>   full selects full copy mirror pairs.

>   any selects the mirroring feature defined by the specified mirror pair(s).

–replica *replica*

>   Specifies if local or remote mirror pair(s) should be processed.

>   Supported values: local | remote | bytarget (default)

>   local selects local replica(s) in the same storage system.

>   remote selects remote replica(s) in a different storage system of the same model. This can be on a different site.

>   bytarget selects the replica location defined by the specified mirror pair(s). The automatically selected replica must be the same for all specified mirror pairs.

–source *storid*[,...]

>   Specifies the storID(s) of one or a list of source volumes for the mirror pair(s).

>   The number of specified storIDs for the source volumes must be equal to the number of specified storIDs for the target volumes.

–target *storid*[,...]

>   Specifies the storID(s) of one ore a list of target volumes for the mirror pair(s).

>   The number of specified storIDs for the source volumes must be equal to the number of specified storIDs for the target volumes.

–*connectparam*

>   Refer to the description of the "Connection parameters".

Return codes

Error Code	Name	Error Type
0	STORMAN_OK	Function successful.
1	STORMAN_NO_ACTION	Nothing to do.
2	STORMAN_NOT_COMPLETE	Function only partly completed.
3	STORMAN_VER_NOT_SUPP	Version not supported
7	STORMAN_INV_PARAMETERS	Specified parameters invalid.
8	STORMAN_INVALID_STORID	Specified storID not allowed
10	STORMAN_STORID_NOT_ FOUND	storID does not yet exist.
13	STORMAN_COMMUNIC_ERROR	Error in StorMan communication between client and server.
14	STORMAN_STORAGE_LOCKED	Storage unit currently locked.
16	STORMAN_FAULT	Unexpected error in function.
22	STORMAN_RESOURCE_NOT_ AVAILABLE	CIM Server or database not available
25	STORMAN_CIMOM_AUTH_ FAILED	User / password for CIM Server not valid.
29	STORMAN_MIRROR_ERROR	Failure of the mirror operation
35	STORMAN_SYSTEM_LOCKED	The storage system is locked for administration
36	STORMAN_AUTH_FAILED	User / password for StorMan not valid
37	STORMAN_CIMOM_ERROR	CIMOM returned an error

14.13.8 stormirror –terminate

Terminates one or more local or remote mirror pairs. This function removes the relationship between the original volumes and the mirror volumes.

Syntax

```
stormirror -terminate [ -mirrortype mirrortype ] [ replica replica ]
                      -source storid[,...] { -target storid[,...] |
                      [ -force ] -connectparam ...
```

Parameters

–mirrortype *mirrortype*

>   Specifies the type of mirror pair(s) to be terminated.
>>   Supported values: `snap` | `full` | `any` (default)
>>   `snap` selects snapshot (pointer based) mirror pair(s).
>>   `full` selects full copy mirror pairs.
>>   `any` selects the mirroring feature defined by the specified mirror pair(s).

–replica *replica*

>   Specifies if local or remote mirror pair(s) should be processed.

>   Supported values: `local` | `remote` | `bytarget` (default)

>   `local` selects local replica(s) in the same storage system.

>   `remote` selects remote replica(s) in a different storage system of the same model. This can be on a different site.

>   `bytarget` selects the replica location defined by the specified mirror pair(s). The automatically selected replica must be the same for all specified mirror pairs.

–source *storid*[,...]

>   Specifies the storID(s) of one ore a list of source volumes for the mirror pair(s).

>   The number of specified storIDs for the source volumes must be equal to the number of specified storIDs for the target volumes.

–target *storid*[,...]

>   Specifies the storID(s) of one ore a list of target volumes for the mirror pair(s).

>   The number of specified storIDs for the source volumes must be equal to the number of specified storIDs for the target volumes.

–force

>   Terminates the mirror pair(s) regardless of their status. The function is performed as supported by underlaying functions of the storage system.

–*connectparam*

>   Refer to the description of the "Connection parameters".

Return codes

Error Code	Name	Error Type
0	STORMAN_OK	Function successful.
1	STORMAN_NO_ACTION	Nothing to do.
2	STORMAN_NOT_COMPLETE	Function only partly completed.
3	STORMAN_VER_NOT_SUPP	Version not supported
7	STORMAN_INV_PARAMETERS	Specified parameters invalid.
8	STORMAN_INVALID_STORID	Specified storID not allowed
10	STORMAN_STORID_NOT_ FOUND	storID does not yet exist.
13	STORMAN_COMMUNIC_ERROR	Error in StorMan communication between client and server.
14	STORMAN_STORAGE_LOCKED	Storage unit currently locked.
16	STORMAN_FAULT	Unexpected error in function.
18	STORMAN_NOT_FOUND	No wwpn for the specified host was found in the repository.
22	STORMAN_RESOURCE_NOT_ AVAILABLE	CIM Server or database not available
25	STORMAN_CIMOM_AUTH_ FAILED	User / password for CIM Server not valid.
29	STORMAN_MIRROR_ERROR	Failure of the mirror operation
35	STORMAN_SYSTEM_LOCKED	The storage system is locked for administration
36	STORMAN_AUTH_FAILED	User / password for StorMan not valid
37	STORMAN_CIMOM_ERROR	CIMOM returned an error

14.13.9 stormirror –show

Displays information about the relationship and status of mirrored pairs. The selection is controlled by the specified parameters. E.g. use parameters `–system` *system* and `-mirrorusage origlocal` to show all volumes on a specific storage system that are used as source volumes for local mirroring. A subset of the selection parameters of `storcfg volume –show` is supported.

Syntax

```
stormirror -show [ -storid storid ] [ –poolid poolid ] [ –system system ]
                [ –mirrortype mirrortype ] [-copymode copymode ]
                  [ –mirrorusage mirrorusage ] [ –format format ] –connectparam ...
```

Parameters

–`storid` *storid*

   Specifies the storID for the logical volume (default: *). Wildcards (*) are supported.

   To display all logical volumes of a storage system (even if they are not managed by StorMan) specify `-storid #all` and exactly one storage system using the `system` parameter. All other parameters are ignored in this case.

–`poolid` *poolid*

   Specifies the ID of a pool the logical volume belongs to (default: *).
   Wildcards (*) are supported.

–`system` *system*

> Specifies the system name of the storage system the logical volume belongs to (default: *). Wildcards (*) are supported.

–`mirrortype` *mirrortype*

> Specifies the mirror type of the volume for mirroring (replication) functions.
>
> Supported values: `snap` | `full` | `any` (default)
>
> `snap` selects snapshot (pointer based) mirror pair(s).
>
> `full` selects full copy mirror pairs.
>
> `any` selects the mirroring feature defined by the specified mirror pair(s).

–`copymode` `copymode`

> Selects the displayed volumes according to their copy mode, supported only for mirrortype `origremote` and `mirrorremote`:
> `sync` | `async` | `asyncstack` | `asyncthrough` | `any` (default)
>
> –`copymode` `sync` selects the volumes with copy mode `sync` .
>
> –`copymode` `async` selects the volumes with copy mode `async`.
>
> –`copymode` `asyncstack` selects the volumes with copy mode `asyncstack`.
>
> –`copymode` `asyncthrough` selects the volumes with copy mode `asyncthrough`.

–`mirrorusage` *mirrorusage*

> Selects the displayed volumes according to their usage as mirroring (replication) pairs (default: any).
>
> Supported values: `origlocal` | `mirrorlocal` | `origremote` | `mirrorremote` | `any` (default)
>
> `any` selects the pairs regardless of their mirror usage.
>
> `origlocal` selects all volumes used as original volumes of local mirror pair(s).
>
> `mirrorlocal` selects all volumes used as mirror volumes of local mirror pair(s).
>
> `origremote` selects original volumes of remote mirror pair(s).
>
> `mirrorremote` selects mirror volumes of remote mirror pair(s).

–`format` *format*

> Specifies the output format. Permitted values: `std` (default) | `short` | `medium` | `xml` | `xmlp`.
>
> `std` selects a multi-line output: Each attribute of an object is displayed on a separate line. This is also the default output format.
>
> `short` displays only the id / name of an object on a separate line.
>
> `medium` displays the id / name and some basic properties of an object on a separate line.
>
> `xml` writes the output in xml format in a single line.
>
> `xmlp` writes the output in xml format, but in a multi-line format for better human readability (pretty print).

–*connectparam*

> Refer to the description of the "Connection parameters".

To display all logical volumes (even if they are not managed by StorMan) of a storage system you have to specify exactly one system using the -`system` parameter and specify -`storid` without a value. All other parameters are ignored in this case.

i	Additionally all selection parameters offered by `storcfg volume –show` are supported.

Return codes

Error Code	Name	Error Type
0	STORMAN_OK	Function successful.
3	STORMAN_VER_NOT_SUPP	Version not supported.
7	STORMAN_INV_PARAMETERS	Specified parameters invalid.
8	STORMAN_INVALID_STORID	Specified storID not allowed
13	STORMAN_COMMUNIC_ERROR	Error in StorMan communication between client and server
16	STORMAN_FAULT	Unexpected error in function.
18	STORMAN_NOT_FOUND	Matching object could not be provided.
22	STORMAN_RESOURCE_NOT_ AVAILABLE	CIM Server or database not available
35	STORMAN_SYSTEM_LOCKED	The storage system is locked for administration
36	STORMAN_AUTH_FAILED	User / password for StorMan not valid
37	STORMAN_CIMOM_ERROR	CIMOM returned an error

Examples

1. Output for logical volumes with `-format std` (default)

```
stormirror -show -storid UID_600000E00D1000000010301C00C10000

storage ID          = UID_600000E00D1000000010301C00C10000
  pool ID           = SHC-OSD-DX-4541142001
  storage system name = Eternus+4541142001
  storage serial nr.  = 4541142001
  device ID         = 193
  GUID              = 600000E00D1000000010301C00C10000
  storage size      = 3600 MB
  type              = normal
  resource type     = predefined
  storage vendor    = Fujitsu
  storage model     = Eternus
  allowed access    = unknown
  current access    = shared
  target OS         = unknown
  status            = READY
  raid level        = mirrored
  raid type         = RAID1
  owner controller  = CM01
  storage pools     = RG03
  host access       = restricted
  mirror targets
    storage ID        =
      storage system name    = Eternus+4531107003
      pool ID           =
      device ID         = 329
      GUID              =
      mirror type       = full
      replication mode  = remote
      copy mode         = sync
      status            = split
      isSynch           = no
      percentSynced     = 32
  no mirror sources found
  no unassigned point in times found
```

2. Output for logical volumes with `-format medium`

```
stormirror -show -storid UID_600000E00D1000000010301C00C10000 -format medium

UID_600000E00D1000000010301C00C10000 storage=Eternus+4541142001 deviceID=193
size=3600 MB
     targets:/Eternus+4531107003/329/split
```

3. Output for logical volumes with `-format xmlp`

```
stormirror -show -storid UID_600000E00D1000000010301C00C  -format xmlp
```

```xml
<Response>
  <ReturnCode>
    <Code>STORMAN_OK</Code>
    <CodeNumber>0</CodeNumber>
    <Text>successful</Text>
    <SubCode>0</SubCode>
    <SubCodeText>STORMAN_SUB_NONE</SubCodeText>
    <MessageID>115416</MessageID>
    <TimeID>1407326342692823</TimeID>
  </ReturnCode>
  <Result>
    <VolumeDescriptions>
      <VolumeDescription>
        <StorID>UID_600000E00D1000000010301C00C10000</StorID>
        <PoolID>SHC-OSD-DX-4541142001</PoolID>
        <StorageSystemName>Eternus+4541142001</StorageSystemName>
        <StorageSerialNumber>4541142001</StorageSerialNumber>
        <StorageCustomName>ABGDX8700-S2-001</StorageCustomName>
        <GUID>600000E00D1000000010301C00C10000</GUID>
        <DeviceID>193</DeviceID>
        <DeviceNumber>193</DeviceNumber>
        <VolumeName>QA_VG011</VolumeName>
        <Size>3774873600</Size>
        <SizeNormalized>3600 MB</SizeNormalized>
        <StorageVendor>STORMAN_STORAGE_VENDOR_FUJITSU</StorageVendor>
        <StorageModel>STORMAN_STORAGE_MODEL_ETERNUS</StorageModel>
        <ResourceType>STORMAN_RESOURCE_TYPE_PREDEFINED</ResourceType>
        <AllowedAccess>STORMAN_ACCESS_MODE_ANY</AllowedAccess>
        <CurrentAccess>STORMAN_ACCESS_MODE_SHARED</CurrentAccess>
        <TargetOS>STORMAN_OS_ANY</TargetOS>
        <Status>STORMAN_VOLUME_STATUS_READY</Status>
        <VolumeType>STORMAN_VOLUME_TYPE_NORMAL</VolumeType>
        <HostAccess>STORMAN_HOST_ACCESS_RESTRICTED</HostAccess>
        <RaidLevel>STORMAN_RAID_LEVEL_1</RaidLevel>
        <RaidType>RAID1</RaidType>
        <CurrOwnerController>CM01</CurrOwnerController>
        <NetappVolumeName/>
        <SpaceConsumedInBox>3774873600</SpaceConsumedInBox>
        <MayBeSource>true</MayBeSource>
        <MayBeTarget>true</MayBeTarget>
        <StoragePoolNames>
          <StoragePoolName>RG03</StoragePoolName>
        </StoragePoolNames>
        <StoragePoolIDs>
          <StoragePoolID>FUJITSU:RSP0002</StoragePoolID>
        </StoragePoolIDs>
        <Type>Unrestricted</Type>
        <HostConnections>
        </HostConnections>
        <MirroringTargets>
          <MirroringTarget>
            <StorID/>
            <DeviceID>329</DeviceID>
```

```
                <DeviceNumber>329</DeviceNumber>
                <GUID/>
                <StorageSystemName>Eternus+4531107003</StorageSystemName>
                <StorageSerialNumber>4531107003</StorageSerialNumber>
                <StorageCustomName>Eternus+4531107003</StorageCustomName>
                <PoolID/>
                <MirrorType>STORMAN_MIRROR_TYPE_FULL</MirrorType>
                <ReplicationMode>STORMAN_MIRROR_REPLICA_REMOTE</ReplicationMode>
                <Status>STORMAN_MIRROR_PAIR_STATUS_SPLIT</Status>
                <CopyMode>STORMAN_REMOTE_COPY_MODE_SYNC</CopyMode>
                <SplitMode>STORMAN_MIRROR_SPLIT_MODE_UNKNOWN</SplitMode>
                <RecoveryMode>STORMAN_MIRROR_RECOVERY_MODE_UNKNOWN</RecoveryMode>
                <IsSynch>false</IsSynch>
                <PercentSynced>32</PercentSynced>
                <PairEstablishedSince/>
             </MirroringTarget>
          </MirroringTargets>
          <MirroringSources>
          </MirroringSources>
          <TargetPorts>
          </TargetPorts>
          <CustomProperties>
             <BS2CATID/>
             <BS2MN>3401</BS2MN>
             <BS2VSN/>
          </CustomProperties>
          <OperationalStatus>OK</OperationalStatus>
          <OperationalStatusDetailList>
          </OperationalStatusDetailList>
        </VolumeDescription>
      </VolumeDescriptions>
    </Result>
  </Response>
```

14.14 storparam

Modifies the settings for the StorMan Server without restarting it. All settings are valid for the current StorMan Server session. They can optionally be maintained persistent for further sessions.

Syntax

```
storparam { –show | [ –debug level ] [ –debugcom comopt ] [ –traceworker level]
          [ –cimtimeout timeout ] [ –createtimeout timeout ]
          [ –pmcc pmcc_installation_directory ] [ –save ]
          [ –addpartner partner ] [ –rempartner partner ] }
          –connectparam ...
```

Parameters

–show

   Shows the settings of the StorMan Server.

–debug *level*

   Sets the debug level of the server. Supported values:

0	Trace severe errors only
1	Trace errors too
2	Trace even warnings
3	Trace information
4	Trace debug (default)
5	Trace extended debug

–debugcom *comopt*

   Sets the debug option for communication (only active if the overall debug level is 5). Supported values:

1	Trace tcp / ip calls of clients
2	Trace messages of clients
4	Trace tcp / ip calls of CIM communication
8	Trace messages of CIM communication

   Adding the values enables multiple traces.

–traceworker *level*

   Enables the tracing of background threads with debug trace level (not higher than the level set by debug) to the trace file. Supported values:

0	Trace off (default)
1	Trace worker thread
2	Trace updater threads

   Adding the values enables multiple levels.

–cimtimeout *timeout*

   Specifies the timeout for the CIM-OM connection (in seconds). Default: 600 seconds.

–createtimeout *timeout*

> Specifies the timeout for the creation of a volume. Default: unlimited.
> Any value lower than or equal to 0 is interpreted as unlimited.

–pmcc *pmcc_installation_directory*

> Set the installation folder for pmcc. This is only required to obtain performance data from ETERNUS DX storage systems.

–addpartner *partner*

> Adds a second StorMan Server on a partner system to duplicate all configuration changes in the StorMan repository. This feature is required to maintain a consistant StorMan repository to support redundant StorMan Server configurations, especially in SQ Server HA configuration running on MARS.
>
> This configuration setting may also be saved in the file smserver.ini using parameter –save.
>
> The value *partner* is a string in the format '<*ip-address-or-name*,*user*,*password*>'.

–rempartner *partner*

> Removes a second StorMan Server on a partner system to duplicate all configuration changes in the StorMan repository. This feature is required to maintain an a consistant StorMan repository to support redundant StorMan Server configurations especially in SQ Server HA configuration when running on MARS. This configuration setting may also be done in the file smserver.ini.

–save

> Saves the currently specified parameter values to the file smserver.ini to be preserved for subsequent sessions (restart of StorMan Server).

–connectparam

> Refer to the description of the "Connection parameters".

Return codes

Error Code	Name	Error Type
0	STORMAN_OK	Function successful
7	STORMAN_INV_PARAMETERS	Specified parameters invalid
13	STORMAN_COMMUNIC_ERROR	Error in StorMan communication between client and server
16	STORMAN_FAULT	Unexpected error in function
36	STORMAN_AUTH_FAILED	User / password for StorMan not valid

14.15 storstat

The `storstat` command is used to manage the collection of performance and statistical information. The command is restricted for internal use.

```
storstat object –function –functionparam ... –connectparam ...
```

The first parameter specifies the object group of the repository that is configured:

First parameter *object*	Object
`volume`	logical volumes of storage systems visible by StorMan

The second parameter specifies the function:

Second parameter function	Meaning
`–get`	get statistic information

If the `–get` function is specified, the output format can be selected by specifying the `-format` parameter.

14.15.1 storstat volume

`storstat volume` is used to get statistical and performance data for storage volumes by StorMan.

```
storstat volume –function –functionparam ... –connectparam ...
```

Functions

The `–function` parameter supplies the following functions:

`–get`

   Gets the statistical data for volumes of a storage system.

14.15.1.1 storstat volume –get

Gets the statistical and performance data related to storage volumes of a storage system. Selection criteria for the `–get` function can be specified using the parameters described below.

Syntax

```
storstat volume –get -system system [ -format format ] –connectparam ...
```

Parameters

`–system` *system*

>   Specifies the system name of the storage system the logical volume belongs to. Wildcards (*) are supported.

`–format` *format*

>   Specifies the output format. Permitted values: `std` (default) | `short` | `xml` | `xmlp`.

>   `std` selects a multi-line output: Each attribute of an object is displayed on a separate line. This is also the default output format.

>   `short` displays only the id / name of an object on a separate line.

>   `xml` writes the output in xml format in a single line.

>   `xmlp` writes the output in xml format, but in a multi-line format for better human readability (pretty print).

–connectparam

>   Refer to the description of the "Connection parameters".

Return codes

Error Code	Name	Error Type
0	STORMAN_OK	Function successful.
3	STORMAN_VER_NOT_SUPP	Version not supported.
7	STORMAN_INV_PARAMETERS	Specified parameters invalid.
13	STORMAN_COMMUNIC_ERROR	Error in StorMan communication between client and server.
16	STORMAN_FAULT	Unexpected error in function.
18	STORMAN_NOT_FOUND	Matching object could not be provided.
21	STORMAN_FCT_NOT_ POSSIBLE	Function rejected by SMI-S Provider.
22	STORMAN_RESOURCE_NOT_ AVAILABLE	CIM Server or database not available.
36	STORMAN_AUTH_FAILED	User / password for StorMan not valid

Output

The following volume specific metrics are provided:

Name	Description	Comments
TimeLastSampled	Absolute point in time of last sampled statistical data. Format: yyyy-mm-dd hh:mm:ss.sss	Example: 2014-01-13 13:56:15.000
SampleInterval	Period of time the statistical data are collected from the storage array. Format: in seconds	Default setting: 3 min. (ETERNUS DX) 5 min. (Symmetrix) Example: 180 (in seconds)
StatisticsTime	Absolute point in time of collecting statistical data. Format: CIM_DATETIME	Available for Symmetrix. Example: 20080319063103.781250+000 yyyymmddHHMMSS.mmmmmmsUUU
ReadIOs	Number of read IOs (Counter)	Available for ETERNUS DX and Symmetrix.

Name	Description	Comments
WriteIOs	Number of write IOs (Counter)	Available for ETERNUS DX and Symmetrix.
MBytesRead	Amount of MB read (Counter)	Available for ETERNUS DX and Symmetrix.
MBytesWritten	Amount of MB written (Counter)	Available for ETERNUS DX and Symmetrix.
ResponseTimeRead	Response time for Read IOs (msec). current value	Available for ETERNUS DX only.
ResponseTimeWrite	Response time for Write IOs (msec). current value	Available for ETERNUS DX only
ResponseTimeTotal	Total Response time for IOs (msec). current value	Available for ETERNUS DX only
CacheHitRateRead	Cache Hit Rate for Read IOs	Available for ETERNUS DX and Symmetrix.
CacheHitRateWrite	Cache Hit Rate for Write IOs	Available for ETERNUS DX and Symmetrix.
CacheHitRatePrefetch	Cache Hit Rate for Prefetch	Available for ETERNUS DX and Symmetrix.
CacheHitRateTotal	Total Cache Hit Rate	Available for ETERNUS DX and Symmetrix.
SampledReadsCounter	Number of read IOs sampled to obtain the sampledReadsTime	Available for Symmetrix only.
SampledReadsTime	Cumulated time value in µsec elapsed for the sampled read IOs	Available for Symmetrix only.
SampledWritesCounter	Number of write IOs sampled to obtain the sampledWritesTime	Available for Symmetrix only.
SampledWritesTime	Number of write IOs sampled to obtain the sampledWritesTime	Available for Symmetrix only.
IdleTimeCounter	Absolute counter representing all samples finding the IO processing idleing for the respective volume	refer to IOTimeCounter

- All Counters are defined as UINT64 (the maximum value being 18446744073709551615). However overflow is handled by the provider.

- All time values are provided in micro seconds (µsec).

Examples

1. Output for logical volumes on ETERNUS DX410 with `-format xmlp` (default)

```
storstat volume -get -system Eternus+4621347002 -format xmlp
  <Response>
    <ReturnCode>
      <Code>STORMAN_OK</Code>
      <CodeNumber>0</CodeNumber>
      <Text>successful</Text>
      <SubCode>0</SubCode>
      <SubCodeText>STORMAN_SUB_NONE</SubCodeText>
      <MessageID>115471</MessageID>
    </ReturnCode>
    <Result>
      <StorageSystem>
        <StorageSystemName>Eternus+4621347002</StorageSystemName>
        <StorageModel>STORMAN_STORAGE_MODEL_ETERNUS</StorageModel>
        <StorageVendor>STORMAN_STORAGE_VENDOR_FUJITSU</StorageVendor>
        <TimeLastSampled>2014-08-06 16:33:42.000</TimeLastSampled>
        <SampleInterval>180</SampleInterval>
        <UTCDiff>120</UTCDiff>
        <StatusCode>0</StatusCode>
        <Status>successful</Status>
        <StatisticsVolumes>
          <StatisticsVolume>
            <DeviceID>1</DeviceID>
            <ReadIOs>0</ReadIOs>
            <WriteIOs>0</WriteIOs>
            <MBytesRead>0</MBytesRead>
            <MBytesWritten>0</MBytesWritten>
            <ResponseTimeRead>0</ResponseTimeRead>
            <ResponseTimeWrite>0</ResponseTimeWrite>
            <ResponseTimeTotal>0</ResponseTimeTotal>
            <CacheHitRateRead>0</CacheHitRateRead>
            <CacheHitRateWrite>0</CacheHitRateWrite>
            <CacheHitRatePrefetch>0</CacheHitRatePrefetch>
            <CacheHitRateTotal>0</CacheHitRateTotal>
          </StatisticsVolume>...
        </StatisticsVolumes>
      </StorageSystem>
    </Result>
  </Response>
```

2. Output for logical volumes with `-format std` (default)

```
storstat volume -get -system Eternus+4621347002
  storage system name = Eternus+4621347002
    storage vendor      = STORMAN_STORAGE_VENDOR_FUJITSU
    storage model       = STORMAN_STORAGE_MODEL_ETERNUS
    status              = successful
    time last sampled   = 2014-08-06 16:39:42.000
    sample interval     = 180
    storage StatisticsVolume(s)
      DeviceID = 1
      ReadIOs = 0
      WriteIOs = 0
      MBytesRead = 0
      MBytesWritten = 0
      ResponseTimeRead = 0
      ResponseTimeWrite = 0
      ResponseTimeTotal = 0
      CacheHitRateRead = 0
      CacheHitRateWrite = 0
      CacheHitRatePrefetch = 0
      CacheHitRateTotal = 0
    ...
```

```
storstat volume -get -system *405 (Symmetrix)
  storage system name = SYMMETRIX+000287400405
    ...
    time last sampled  = 20080319063107.906000+000
    sample interval    = 00000000000500.000000:000
    storage StatisticsVolume(s)
      DeviceID = 0000
      StatisticTime = 20080319063103.781250+000
      ReadIOs = 18234
      WriteIOs = 0
      KBytesRead = 9120
      KBytesWritten = 0
      SampledReadsCounter = 4535
      SampledReadsTime = 845888
      SampledWritesCounter = 0
      SampledWritesTime = 0
```

2. Output for logical volumes with `-format xmlp`

```
storstat volume -get -system *405 -format xmlp (Symmetrix)
  <Response>
    <ReturnCode>
      <Code>STORMAN_OK</Code>
      <CodeNumber>0</CodeNumber>
      <SubCode>STORMAN_SUB_NONE</SubCode>
      <Text>Successful</Text>
    </ReturnCode>
    <Result>
      <StorageSystem>
        <StorageSystemName>SYMMETRIX+000287400405</StorageSystemName>
        <StorageModel>STORMAN_STORAGE_MODEL_SYMMETRIX</StorageModel>
        <StorageVendor>STORMAN_STORAGE_VENDOR_EMC</StorageVendor>
        <Status>succesful</Status>
        <TimeLastSampled>20080319134128.875000+000</TimeLastSampled>
        <SampleInterval>00000000000500.000000:000</SampleInterval>
        <StatisticsVolumes>
          <StatisticsVolume>
            <DeviceID>0000</DeviceID>
            <StatisticTime>20080319134112.656250+000</StatisticTime>
            <ReadIOs>18618</ReadIOs>
            <WriteIOs>0</WriteIOs>
            <KBytesRead>9312</KBytesRead>
            <KBytesWritten>0</KBytesWritten>
            <SampledReadsCounter>4620</SampledReadsCounter>
            <SampledReadsTime>861600</SampledReadsTime>
            <SampledWritesCounter>0</SampledWritesCounter>
            <SampledWritesTime>0</SampledWritesTime>
          </StatisticsVolume>
        ...
        </StatisticsVolumes>
      </StorageSystem>
    </Result>
  </Response>
```

15 Appendix

15.1 Fujitsu Technology Solutions:
Software License Agreement for End Users

1. Subject of this Agreement

1.1   For the purposes of this agreement "Software" shall mean the software with the object code, the version and the specification indicated in the software product data sheet of Fujitsu Technology Solutions.

The Software consists of machine-readable instructions and/or printed documentation and related licensed materials.

1.2   Please read this agreement carefully before you use the Software. If you do not agree with the license terms in this agreement, you are not permitted to use the Software and must immediately return all copies of the Software and all accompanying items to the Licensor/Sublicensor (either Fujitsu Technology Solutions or the reseller who supplied you with the Software) with proof of purchase for a full refund.

1.3   Any use of the Software requires the proper payment of the applicable license fees. By using the Software you agree to be bound by the terms of this agreement.

1.4   Fujitsu Technology Solutions reserves the right to implement at any time in the future an additional software license key and/or license certificates as countermeasures against software piracy.

1.5   Software components from third-party software suppliers which are part of the scope of the delivery are subject to separate license agreements that are included with the Software or that are transmitted by Fujitsu Technology Solutions upon request.

2. End User License

2.1   Fujitsu Technology Solutions grants you a non-exclusive and non-transferable license to use the Software on the number of workstations for which you have purchased licenses. Unless you purchase additional licenses, you are not permitted to operate the Software on more than the maximum number of licensed workstations or on hardware that exceeds the specified type.

You are permitted to make a backup copy of the Software for archiving purposes, provided you properly mark the copy or partial copy of the Software with the copyright notice and any other ownership information.

2.2   You are not permitted to copy, modify or distribute the Software. Furthermore, you are not permitted to re-compile, re-engineer, convert, revise, compile or modify the Software. You may not sub-license, without selling the related hardware, assign, rent, lease or transfer the Software except as expressly permitted by this agreement or due to mandatory legal regulations.

2.3   If you acquired the Software as a program upgrade, your license for the use of the old software version ends automatically with the installation of the upgrade version of the Software. If parts of the old software version are not replaced by the upgrade version, the license for the old version continues to be effective until the remnants of the old software version are also replaced or deactivated or shut down in any other way.

2.4   Unless specified otherwise in the respective software data sheet of Fujitsu Technology Solutions, the license for a software version or release does not give you any rights to new releases (updates), new versions (upgrades) or technical support services for the Software. Supplemental software support contracts and maintenance services, including or excluding new releases and new versions and additional technical support services, can be purchased separately either from Fujitsu Technology Solutions directly or from authorized software resellers.

3. Downloading

For Software supplied by Fujitsu Technology Solutions over a network or a similar distribution path, the following additional conditions shall apply:

All products supplied for downloading by Fujitsu Technology Solutions are selected, made available and - if supplied by third parties - provided without modification. However, you are fully responsible for ensuring the most current version and usability of downloadable material for your own purposes and on your own system. You download Software at your own risk. Fujitsu Technology Solutions will not accept any liability, particularly not for transmission errors or problems that arise during the downloading process (line failures, connection interruptions, server failures, data corruption, etc.).

The website of Fujitsu Technology Solutions is operated and administered only for those countries in which Fujitsu Technology Solutions has one or more offices. Fujitsu Technology Solutions accepts no responsibility that Software and/or documentation can or may be downloaded from a Fujitsu Technology Solutions website also in locations other than the countries mentioned above. If you access a website of Fujitsu Technology Solutions from abroad, you are fully responsible for complying with any local regulations. Fujitsu Technology Solutions expressly prohibits the downloading of Software and/or documentation from a Fujitsu Technology Solutions website in countries where such downloading is considered illegal.

4. Copyright

All rights and licenses, unless they are expressly granted to you in this license terms, as well as all property and usage rights related to the Software (including parts of the Software) remain fully with Fujitsu Technology Solutions and/or its third-party licensors.

The license terms do not authorize you to use the brands, logos or trademarks of Fujitsu Technology Solutions or its third-party licensors, nor are you permitted to use any other brands which are deceptively similar to the brands, logos or trademarks of Fujitsu Technology Solutions. Each and any use of brands, logos or trademarks with respect to the Software or Fujitsu Technology Solutions requires the express consent of Fujitsu Technology Solutions.

5. Licensor's warranty and liability disclaimer, if Software is sold and delivered by Reseller

If you acquire the Software directly from an authorized reseller (called "Reseller"), the right to install and use the Software may be subject to additional software license conditions agreed upon between you as the licensee and the respective reseller.

In all cases of an authorized software resale, the software is sublicensed and made available to the licensee directly by the Reseller. In such cases, Fujitsu Technology Solutions is not a contractual party of the software license agreement between you, as licensee and the Reseller, as far as the procurement of the software licenses are concerned. Legal claims in connection with the software licensing can therefore be asserted only on the basis of the agreements with the Reseller. Under no circumstances, however, will the respective scope of the license for the licensee exceed the scope of the license agreements as specified in sections 1, 2, 3 and 4 of this agreement.

Subject to mandatory legal regulations, particularly those governing liability and/or warranties, which cannot be excluded in connection with end user license agreement regulations and with reference to the licensee's claims against the Reseller, Fujitsu Technology Solutions disclaims all warranties for the Software in this agreement. For the same reason, Fujitsu Technology Solutions disclaims any and all liability/claims for any violations of third parties' rights as well as any implied warranties for the software's marketability and its suitability for a particular purpose. This disclaimer of liability does not apply in cases of willful or malicious behavior by Fujitsu Technology Solutions.

In this End User License Agreement, Fujitsu Technology Solutions grants no warranties of any kind, either express or implied.

6. Disclaimer of liability with respect to shareware, freeware and/or open source software components

6.1 The Software may contain freeware or shareware which Fujitsu Technology Solutions received from a third party. Fujitsu Technology Solutions paid no license fees for the use of this freeware or shareware. Accordingly, the licensee is not charged any license fees for the use of the freeware or shareware. You recognize and accept that Fujitsu Technology Solutions therefore grants no warranties with respect to such freeware or shareware components and does not assume any liability in connection with the ownership, the distribution and/or the use of the respective freeware or shareware.

6.2 The Software may also contain open source software components that were developed according to the "open source model" and which are distributed exclusively on the basis of the GPL (General Public License: *http://www.gnu.org/copyleft/gpl.html*) terms and conditions or other standard open source standard license terms and conditions applicable to the respective open source components at the time of their dissemination. You recognize and accept that the licensing of such open source software components is governed exclusively by the abovementioned GPL terms or by the conditions which are otherwise included with the open source software components. Fujitsu Technology Solutions receives neither license fees nor any other compensation for the delivered open source software components. As far as Fujitsu Technology Solutions or a third party receives any compensation in connection with open source software components, it is received exclusively for additional delivery items and/or services.

Because of the special nature of the development and distribution of open source software components, Fujitsu Technology Solutions assumes no express or implied liability for such components and excludes any kind of warranty for such open source software components, particularly in connection with missing specifications, lack of functionality, programming errors or any other malfunctions.

7. General limitations of liability

7.1 Neither Fujitsu Technology Solutions nor its suppliers are liable for any consequential or indirect damages, including damages arising as a result of or in connection with an operational interruption, lost profits or sales, lost data, or costs of capital. Fujitsu Technology Solutions and its suppliers will not be liable for additional ancillary or consequential costs or for any other losses, costs or expenses of any kind which arise as a result of the holding, sale, use or impossibility of use of the Software, independent of whether such claims are asserted due to warranty rights, contracts, tort or any other legal theory.

7.2 The liability of Fujitsu Technology Solutions for direct damage caused as a result of a contract violation and/or other action or lapse on the part of Fujitsu Technology Solutions which have not been excluded or cannot be completely excluded due to mandatory law are limited to no more than €250,000.00. Any and all other liabilities for direct damage are excluded. Damage caused by Fujitsu Technology Solutions as a result of slight negligence are excluded to the extent permitted by applicable legal regulations.

7.3   Limitations and exclusions of liability resulting from this agreement do not apply to damage where Fujitsu Technology Solutions carries compulsory liability according to applicable laws and where such liability cannot be limited to a maximum amount (for example, liability for bodily damage; product liability or fraudulently incorrect information).

8. Export controls

Due to its components as well as the nature or purpose of these components, the export of the Software and/or its accompanying documents may be subject to official or regulatory approval. In cases where the Software is intended for export, you are obliged to get all approvals and authorizations required to comply with all relevant export regulations.

The Software may not be exported if there is reason to assume that the Software will be used in connection with nuclear, chemical or biological weapons or for missile technology. Furthermore, you may not deliver the Software - or have it delivered indirectly - to such companies or persons who are listed in the applicable U.S. export regulations (particularly the Table of Denial Orders/U.S. Denied Persons Lists (DPL) or in the E.U. export regulations (particularly the EU Terrorist List) or in the applicable warnings issued by the German export authorities or any other competent authorities in any country.

Under no circumstances is Fujitsu Technology Solutions obligated to deliver software, patches, updates or upgrades, to provide software for download or to fulfill any other contractual commitments if this would be a violation of the applicable export regulations of the Federal Republic of Germany, the European Union, the United States of America or of any other countries.

If you export or re-export the Software or a copy of it, this may be a violation of applicable export laws and a severe violation of the terms of this agreement.

9. Miscellaneous

9.1   If any term or condition in this agreement or any other contract that is subject to the terms and conditions of this agreement turns out to be invalid or unenforceable (partly or in full), the validity of all other terms and conditions remains unaffected, unless complying with the remaining terms and conditions would represent an unreasonable hardship for either contract party, even with the application of applicable legal regulations to close the legal gap.

9.2   If you/ the licensee do not pay the license fees due and/or if the licensee does not comply with essential terms and conditions of this license agreement, Fujitsu Technology Solutions reserves the right to cancel the license. In case of such cancellation, you must immediately return any and all copies of the software in your possession and confirm the complete return [of the software copies] or the destruction of these copies in writing.

9.3   Neither you nor Fujitsu Technology Solutions is responsible or liable for the respective party's non-compliance with its obligations if the reason for such non-compliance is outside the party's control due to force majeure.

9.4   Any and all modifications and/or amendments to these license terms and conditions are only valid if they are made in writing.

10. Applicable law

10.1   These license terms and conditions are governed by the laws of the Federal Republic of Germany.

10.2   In the event that provisions of clause 10.1 are unenforceable, these license terms and conditions shall be governed by the laws of the country in which you acquire the Software, with the following exceptions: 1) In Australia, the terms and conditions of this license are governed by the laws of the state or sovereign territory in which the business contract is being concluded; 2) in Albania, Armenia, Belarus, Bosnia-Herzegovina, Bulgaria, Croatia, the Czech Republic, Georgia, Hungary, Kazakhstan, Kirgizia, the former Yugoslavian Republic of Macedonia (FYROM), Moldavia, Poland, Romania, Russia, Slovakia, Slovenia, the Ukraine

and the Federal Republic of Yugoslavia, the terms and conditions of this license are governed by the laws of the Federal Republic of Germany; 3) in the United Kingdom [Great Britain], all disputes with respect to these license terms and conditions are governed by English law, and English courts have exclusive jurisdiction; 4) in Canada, the terms and conditions of this license are governed by the laws of the Province of Ontario; 5) in the United States of America and in Puerto Rico as well as in the People's Republic of China the terms and conditions of this license are governed by the laws of the U.S. State of New York.

15.2 Third Party Licenses

These document contains legal attribution for acknowledging open-source and third-party software copyright, and licensing requirements for StorMan V6.0.

15.2.1 OpenSSL License

http://www.sdisw.com/openssl.htm

Copyright (c) 1998-2000 The OpenSSL Project. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. All advertising materials mentioning features or use of this software must display the following acknowledgment: This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (http://www.openssl.org)

4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact openssl-core@openssl.org

5. Products derived from this software may not be called "OpenSSL" nor may "OpenSSL" appear in their names without prior written permission of the OpenSSL Project.

6. Redistributions of any form whatsoever must retain the following acknowledgment: "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org)"

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
This product includes cryptographic software written by Eric Young (eay@cryptsoft.com).
This product includes software written by Tim Hudson (tjh@cryptsoft.com).

15.2.2 SQLite License

All of the deliverable code in SQLite has been dedicated to the public domain (http://en.wikipedia.org/wiki/Public_domain) by the authors. All code authors and representatives of the companies for which they work have signed affidavits dedicating their contributions to the public domain and originals of those signed affidavits are stored in a firesafe at the main offices of Hwaci

(http://www.hwaci.com). Anyone is free to copy, modify, publish, use, compile, sell, or distribute the original SQLite code, either in source code form or as a compiled binary, for any purpose, commercial or non-commercial, and by any means.

15.2.3 jquery License

https://jquery.org/license/

Copyright 2013 jQuery Foundation and other contributors

http://jquery.com/

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

15.2.4 jqueryUI License

https://jquery.org/license/

Copyright 2013 jQuery Foundation and other contributors

http://jquery.com/

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including

without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

15.2.5 jquery-cookie MIT-License

https://github.com/carhartl/jquery-cookie/blob/master/MIT-LICENSE.txt

Copyright 2014 Klaus Hartl

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

15.2.6 DataTables License

https://datatables.net/license_mit

MIT license

Copyright (C) 2008-2014, SpryMedia Ltd.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

15.2.7 zTree License

http://www.ztree.me/v3/main.php#_license

zTree -- jQuery tree plug-ins.

License (MIT License)

copyright (c) <2011> < hunter.z >

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

15.2.8 7zip License

http://7-zip.org/license.txt

License for use and distribution
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

7-Zip Copyright (C) 1999-2014 Igor Pavlov.

Licenses for files are:

   1) 7z.dll: GNU LGPL + unRAR restriction

   2) All other files: GNU LGPL

   The GNU LGPL + unRAR restriction means that you must follow both

   GNU LGPL rules and unRAR restriction rules.

   Note:

      You can use 7-Zip on any computer, including a computer in a commercial organization. You don't need to register or pay for 7-Zip.

GNU LGPL information

   This library is free software; you can redistribute it and/or   modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

   This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

   You can receive a copy of the GNU Lesser General Public License from

   http://www.gnu.org/

unRAR restriction

   The decompression engine for RAR archives was developed using source code of unRAR program.

   All copyrights to original unRAR code are owned by Alexander Roshal.

   The license for original unRAR code has the following restriction:

The unRAR sources cannot be used to re-create the RAR compression algorithm, which is proprietary. Distribution of modified unRAR sources in separate form or as a part of other software is permitted, provided that it is clearlystated in the documentation and source comments that the code maynot be used to develop a RAR (WinRAR) compatible archiver.

--

Igor Pavlov

16 Related publications

The manuals are available as online manuals, see *http://manuals.ts.fujitsu.com*, or in printed form which must be paid and ordered separately at *http://manualshop.ts.fujitsu.com*.

[1] **SHC-OSD**
Storage Management for BS2000
User Guide

[2] **IMON** (BS2000)
Installation Monitor
User Guide

[3] **SQ Business Servers**
Operation and Administration
User Guide

[4] **SQ Business Servers**
SQ Manager Help
Online Help

[5] **SQ Business Servers**
Cluster Solutions for SQ Servers
User Guide

[6] **FUJITSU Server BS2000 SE Series**
Operation and Administration
User Guide

[7] **FUJITSU Server BS2000 SE Series**
SE Manager Help
Online Help

17 Index

V

virtualization layer 18
VolumeLogix 85

W

WebUI 55
WWPN
    common parameter 96