Before using this information and the product it supports, read the information in "Notices" on page 57.
Contents

Figures ......................................... v

Tables .......................................... vii

About this guide .............................. ix
Who should use this guide .............. ix
Conventions used in this guide .......... ix
Publications and related information .... ix
Getting information, help, and service ... xii
Ordering publications ...................... xii
Sending your comments .................... xii

Chapter 1. XIV Open API overview ..... 1
CIM agent overview ......................... 1
CIM agent components ................. 2
CIM concepts .................................. 3
CIM agent security ...................... 4

Chapter 2. CIM agent for the XIV system 5
CIM agent installation overview ...... 5
Configuring the CIM agent ............ 5
Enabling the CIM agent ............... 5
Verifying the CIM agent connection ... 5

Chapter 3. XIV Open API component
definitions ................................ 7
CIM agent communication concepts ... 8
CIM agent communication methods ... 8
GetClass .................................... 9
GetInstance ................................ 10
DeleteClass ................................ 10
DeleteInstance .............................. 10
CreateClass ................................ 11
CreateInstance ............................ 11
ModifyClass ................................ 11
ModifyInstance ............................ 11
EnumerateClasses ......................... 12
EnumerateClassNames .................... 13
EnumerateInstances ....................... 13
EnumerateInstanceNames ............... 14
ExecuteQuery ............................. 14
Associators ................................ 15
AssociateNames ........................... 16
References .................................. 17
ReferenceNames ............................ 17
GetProperty ................................ 18
SetProperty ................................ 19
GetQualifier ................................ 19
SetQualifier ................................ 20
DeleteQualifier ............................ 20
EnumerateQualifiers ...................... 20
Return error codes ......................... 20

Chapter 4. Functional diagrams of the
CIM agent .................................. 23
Block Server performance profile .... 23
IBMTS_D_S_PermformanceStatisticsService.
GetStatistics Collection ............... 25
Getting volume statistics with
GetStatisticsCollection ............... 25
Block Services profile .................... 26
IBMTS_D_StorageConfigurationService.
CreateOrModifyStoragePool ........... 30
IBMTS_D_StorageConfigurationService.
DeleteStoragePool ...................... 30
IBMTS_D_StorageConfigurationService.
CreateOrModifyElementsFromStoragePool ... 31
IBMTS_D_StorageConfigurationService.
CreateOrModifyElementFromStoragePool ... 31
IBMTS_D_StorageConfigurationService.
ReturnToStoragePool .................. 33
IBMTS_D_StorageConfigurationService.
ReturnElementsToStoragePool .......... 33
IBMTS_D_PrimordialStoragePool.
GetSupportedSizeRange ............. 33
IBMTS_D_VirtualPool.GetSupportedSizeRange ... 33
Creating a ConcreteStoragePool .... 33
Creating a StorageVolume ........... 34
iSCSI Target Ports profile .......... 34
IBMTS_D_iSCSIConfigurationService.
CreateIPProtocolEndpoint .......... 37
IBMTS_D_iSCSIConfigurationService.
ModifyIPProtocolEndpoint .......... 37
IBMTS_D_iSCSIConfigurationService.
DeleteIPProtocolEndpoint .......... 38
Creating an IP endpoint ............ 38
Modifying an IP endpoint .......... 38
Deleting an IP endpoint ........... 39
Masking and Mapping profile ....... 39
IBMTS_D_StorageHardwareIDManagement.
CreateStorageHardwareID ........... 41
IBMTS_D_StorageHardwareIDManagement.
DeleteStorageHardwareID ........... 43
IBMTS_D_StorageHardwareIDManagement.
CreateHardwareIDCollection ........ 43
IBMTS_D_StorageHardwareIDManagement.
AddHardwareIDsToCollection ....... 44
IBMTS_D_ControllerConfigurationService.
ExposePaths ............................ 44
IBMTS_D_ControllerConfigurationService.
HidePaths ............................... 45
IBMTS_D_ControllerConfigurationService.
CreateIPProtocolEndpoint .......... 45
Sample recipe ......................... 45

© Copyright IBM Corp. 2010, 2012
# Figures

1. How a CIM agent works .................................. 2
2. The MOF compiler stores the model in the CIMOM data store .................................. 4
3. Block Services SMI-S model for XIV systems .................................. 24
4. Block Services SMI-S model for XIV systems .................................. 27
5. Block Services Package with Settings and Capabilities model .................................. 28
6. iSCSI Target Ports SMI-S model for XIV systems .................................. 35
7. Masking and mapping physical model in the XIV system .................................. 40
8. Masking and mapping object model in SMI-S .................................. 41
9. Replication Services SMI-S model for XIV systems .................................. 48
10. Sample group information retrieval .................................. 50
## Tables

1. Functional groups for the CIM agent ........................................ 8
2. GetClass method parameters ................................................. 9
3. GetInstance method parameters ............................................... 10
4. DeleteInstance method parameters ........................................... 10
5. CreateInstance method parameters .......................................... 11
6. ModifyInstance method parameters ......................................... 12
7. EnumerateClasses method parameters ..................................... 12
8. EnumerateClassNames method parameters ................................. 13
9. EnumerateInstances method parameters .................................. 13
10. EnumerateInstanceNames method parameters ......................... 14
11. ExecuteQuery method parameters .......................................... 14
12. Associators method parameters ............................................. 15
13. Associators method parameters ............................................. 16
14. References method parameters .............................................. 17
15. ReferenceNames method parameters ....................................... 17
16. GetProperty method parameters ........................................... 18
17. SetProperty method parameters ............................................. 19
18. GetQualifier method parameters ............................................ 19
19. SetQualifier method parameters ............................................ 20
20. Return error codes for the CIMOM ......................................... 21
21. Block Server metrics .......................................................... 23
22. Synchronous actions .......................................................... 29
23. iSCSI terminology and CIM class names ................................. 36
24. Masking and mapping classes ............................................... 41
25. ExposePaths parameters and use cases .................................. 44
26. HidePaths parameters and use cases ..................................... 45
27. Indication types and object classes ....................................... 46
28. Mapping XIV terminology to SMI terminology ......................... 49
29. Replication Service methods ................................................. 49
30. Group management classes .................................................. 49
31. Extrinsic methods for group management ................................ 50
32. Replication management classes ........................................... 51
33. Extrinsic methods for replication management ......................... 51
34. SMI Indications ............................................................... 54
35. SNIA CTP results ............................................................. 55

© Copyright IBM Corp. 2010, 2012
About this guide

This publication introduces the IBM® XIV® Storage System Open Application Programming Interface (API), which is referred to as the Common Information Model (CIM) agent. This publication can assist you in writing your CIM-based applications for the XIV Open API.

This publication supports the IBM XIV Storage System microcode version 11.1.

Who should use this guide

This publication is for system administrators and system and application programmers, or whoever is responsible for implementing the XIV Open API and configuring the CIM agent.

This publication assumes that you understand the general concepts of the operating system and Internet capabilities for your enterprise.

Conventions used in this guide

These notices are used to highlight key information.

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

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

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

Publications and related information

Product manuals, other IBM publications, and websites contain information that relates to the IBM XIV Storage System.

To view a PDF file, you need Adobe Reader, which can be downloaded for free from the Adobe website (get.adobe.com/reader/).

Information centers

IBM System Storage® Information Center website (publib.boulder.ibm.com/infocenter/ibmxiv/r2/index.jsp)

From the IBM XIV Storage System Information Center, you can browse all product documentation.

Publications

Information that is available in the information center is also available in a set of publications, in PDF format. You can find the following PDFs in the information center on the Publications and related information web page (publib.boulder.ibm.com/infocenter/ibmxiv/r2/topic/com.ibm.help.xiv.doc/xiv_pubsrelatedinfoic.html).

IBM XIV Storage System
  • IBM XIV Storage System Product Overview
This document contains an overview of the IBM XIV Storage System hardware and software.

- **IBM XIV Storage System Planning Guide**
  This guide defines installation requirements for IBM XIV Storage System model A14 and model 114. It is important to ensure that you meet all requirements to guarantee a fast and reliable installation.

- **IBM XIV Storage System Application Programming Interface Reference**
  This reference assists you in developing CIM-based applications using the XIV Open API.

- **IBM XIV Storage System XCLI Reference Guide**
  This reference describes the IBM XIV command-line interface (XCLI) system and utility commands used to manage and maintain the XIV system, including the command syntax, parameter descriptions, output descriptions, and examples.

- **IBM XIV Storage System XCLI User Manual**
  This guide describes how to use the IBM XIV command-line interface (XCLI) to run XIV system and utility commands.

- **IBM XIV Storage System Safety Notices**
  This document describes all standard IBM XIV Storage System safety notices.

  This guide describes how to install, configure, and use the IBM XIV Storage System Remote Support Proxy to connect an XIV system to the XIV Remote Support Center.

### Host software solutions

- **Host Attachment Guide**
  This guide describes how to prepare for, install, and use the IBM XIV Host Attachment Kit (HAK). The guide supports AIX, HP-UX, Linux, Solaris, and Windows. It also includes an appendix for VMware ESX and ESXi users.

- **IBM Storage Management Pack For Microsoft System Center Operations Manager User Guide**
  This guide provides installation, configuration, and usage instructions for the IBM Storage Management Pack for Microsoft System Center Operations Manager (SCOM).

- **IBM XIV Storage System Remote Mirroring for VCS Installation Guide**
  The installation guide describes how to install and configure the Veritas Cluster Server (VCS) enterprise agent for IBM XIV Remote Mirroring.

- **IBM Storage Device Driver for VMware VAAI Installation Guide**
  This guide describes how to install and configure the IBM Storage Device Driver for VMware VAAI.

- **IBM Storage Provider for VMware VASA Installation Guide**
  This guide describes how to install and configure the IBM Storage Provider for VMware VASA.

- **IBM Storage Enabler for Windows Failover Clustering User Guide**
  This guide describes the IBM Storage Enabler for Windows Failover Clustering (formerly titled ‘MSCS Agent’) and provides detailed instructions and best practices for using it.

- **IBM Storage Management Console for VMware vCenter User Guide**
  This publication provides installation, configuration, and usage instructions for the IBM Storage Management Console for VMware vCenter.

- **IBM XIV Adapter for VMware vCenter Site Recovery Manager (SRM) Installation Guide**
  This guide describes how to prepare for, install, configure, and use the IBM XIV Adapter for VMware vCenter Site Recovery Manager.

- **IBM XIV Provider for Microsoft Windows Volume Shadow Copy Service Installation Guide**
  This guide describes how to install and configure the IBM XIV Provider for Microsoft Windows VSS.
IBM Redbooks publications and technical papers

Various IBM Redbooks® publications, Redpapers, and white papers are available for the IBM XIV Storage System. For additional papers, see the [IBM System Storage website](www.ibm.com/systems/storage/disk/xiv/).

- [IBM XIV Storage System: Architecture, Implementation, and Usage](www.redbooks.ibm.com/abstracts/sg247659.html)
  This Redbooks publication describes the concepts, architecture, and implementation of the XIV system, which is a scalable enterprise storage system based upon a grid array of hardware components.

- [IBM XIV Storage System with the Virtual I/O Server and IBM i](www.redbooks.ibm.com/redpieces/abstracts/redp4598.html)
  This Redbooks publication describes how you can connect the XIV system to the IBM i operating system through the Virtual I/O Server (VIOS). A connection through the VIOS is especially useful for IT centers that have many small IBM i partitions. When using the VIOS, the Fibre Channel host adapters can be installed in the VIOS and shared by many IBM i clients by using virtual connectivity to the VIOS.

- [XIV Storage System: Host Attachment and Interoperability](www.redbooks.ibm.com/redpieces/abstracts/sg247904.html)
  This Redbooks publication describes how to attach an XIV system to various hosting operating system platforms in combination with databases and other storage-oriented application software. It also provides solutions for combining the IBM XIV Storage System with other storage platforms, host servers, or gateways.

- [IBM XIV Storage System: Copy Services and Migration](www.redbooks.ibm.com/abstracts/sg247759.html)
  This Redbooks publication describes IBM XIV Storage System copy and migration functions for various data protection scenarios, to enhance your business continuance, data migration, and online-backup solutions. These include point-in-time copies (also known as snapshots and full volume copies) and remote-copy capabilities in synchronous or asynchronous mode. This book also describes how to integrate the snapshot function with the IBM Tivoli® FlashCopy® Manager, built-in migration capability, and migration alternatives based on the IBM SAN Volume Controller.

Related websites

View these websites to get more information about the XIV system.

- [IBM System Storage website](www.ibm.com/systems/storage/disk/xiv/)
  Use this website to learn about the XIV system, including features and hardware summary. This website also has links to white papers, Redbooks publications, and product documentation.

- [IBM Support Portal website](www.ibm.com/storage/support)
  Use this website to obtain downloadable files, links to submit and track problems, and support phone numbers and contacts.

  Use this forum to share ideas with knowledgeable experts and discover how the latest IBM storage solutions can address your business challenges. Forum topics include storage management, storage virtualization, business continuity, infrastructure simplification, disk storage systems, and storage software products and solutions.
Getting information, help, and service

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

- IBM website (ibm.com®)
- IBM Support Portal website (www.ibm.com/storage/support)
- IBM Directory of Worldwide Contacts website (www.ibm.com/planetwide)

Ordering publications

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

The IBM Publications Center website (www.ibm.com/shop/publications/order/) offers customized search functions to help you find the publications that you need. Some publications are available for you to view or download at no charge. You can also order publications. The publications center displays prices in your local currency.

Sending your comments

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

Procedure

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

- Go to the feedback form (publib.boulder.ibm.com/infocenter/ibmxiv/r2/topic/com.ibm.xiv.doc/icfeedback.htm) in the IBM System Storage information center. You can use this form to enter and submit comments.
- Send your comments by email to starpubs@us.ibm.com Be sure to include the following information:
  - Exact publication title and version
  - Publication form number (for example, GA32-0770-00)
  - Page, table, or illustration numbers that you are commenting on
  - A detailed description of any information that needs to be changed
Chapter 1. XIV Open API overview

The IBM XIV Storage System Open Application Programming Interface (API) is a nonproprietary storage-management client application.

The XIV Open API uses the Storage Management Initiative Specification (SMI-S), as defined by the Storage Networking Industry Association (SNIA) to view LUN information.

You can use the XIV Open API to integrate configuration-management support into storage resource management (SRM) applications, so that you can use your existing SRM application and infrastructures to configure and manage the XIV system. The XIV Open API presents another option for managing your systems by complementing the use of the IBM XIV Storage Management GUI and the IBM XIV command-line interface (XCLI). The XIV Open API is an embedded component of the XIV system.

Beginning with the IBM XIV Storage System V10.1.0, you can implement the XIV Open API without using a separate middleware application, like the IBM System Storage Common Information Model (CIM) agent, which provides a CIM-compliant interface. The XIV Open API uses the CIM technology to manage proprietary devices as open system devices through storage management applications. The XIV Open API is used by storage management applications to communicate with an XIV system.

CIM agent overview

A CIM agent provides a means by which a device can be managed by common building blocks rather than proprietary software. If a device is CIM-compliant, software that is also CIM-compliant can manage the device. Vendor applications can manage CIM-compliant devices in a common way, rather than using device-specific programming interfaces. You can perform tasks in a consistent manner across devices and vendor applications.

A CIM agent consists of the components shown in Figure 1 on page 2. The main components are the CIM object manager (CIMOM), the service location protocol (SLP), and the device provider. A device can be a storage system such as the XIV system. The CIM agent registers itself with the SLP Service Agent (SLP SA) to enable discovery by the client application. The SLP DA is a directory service daemon that a client application calls to locate the CIMOM. The client application and the CIMOM communicate through CIM Messages. The CIMOM and device provider communicate through method calls made from the CIMOM to the provider. The device provider communicates with the device through proprietary calls.
The CIMOM supports the following specifications and standards:

- Distributed Management Task Force (DMTF) Specification for CIM Operations over HTTP, Version 1.4
- Common Information Model (CIM) Specification, Version 2.3
- Storage Networking Industry Association (SNIA) Storage Management Initiative Specification (SMI-S) and the Shared Storage Model, a framework for describing storage architectures, Version 1.4

These specifications allow a CIM agent to act as an open-system standards interpreter, allowing other CIM-compliant storage resource management applications (IBM and non-IBM) to interoperate with each other.

When you have installed, configured, and enabled the CIM agent on a host server or an administrative workstation within your network, that host server or workstation can communicate with your storage system through the CIM agent. This allows CIM-compliant applications like the XIV Open API to manage the data on your XIV system.

The following Web sites provide more information about the CIM standards:

- DMTF Common Information Model (CIM) Standards
  [http://www.dmtf.org/standards/cim/](http://www.dmtf.org/standards/cim/)
- Storage Networking Industry Association Standards

### CIM agent components

With a CIM agent, application programmers can use common building blocks rather than proprietary software or device-specific programming interfaces to manage CIM-compliant devices. Standardization of the way that applications manage storage provides easier storage management.

The following list describes the components of a CIM agent:
CIM agent
An agent that interprets open-system data as it is transferred between the API and a device or a storage unit.

CIM object manager (CIMOM)
A common conceptual framework for data management. Receives, validates, and authenticates client application requests, and then directs requests to the appropriate functional component or to a device provider.

Client application
A storage management API that initiates a request to a device or a data storage unit such as an XIV system.

Note: A client application is not provided with the CIM agent, and it must be supplied by the customer.

Service Location Protocol (SLP)
The SLP DA is a directory service that a client application calls to locate the CIMOM. The SLP SA is a service agent to allow discovery by a client application.

Storage unit (also known as a storage server)
The final destination of a client application request and the processor of the request.

Storage unit provider
A storage unit-specific handler that receives client application requests that are destined for its device or storage unit.

CIM concepts
The common information model (CIM) is an open approach to the management of systems and networks.

The CIM provides a common conceptual framework applicable to all areas of management including systems, applications, databases, networks, and devices. The CIM specification provides the language and the methodology used to describe management data.

The CIM defines a set of classes with properties and associations that provide a conceptual framework. The framework can be used to organize data for a specific managed environment, such as data storage. CIM Schema 2.11 for Managing a Storage Array provides information about enabling management applications to manage data in a common way.

The CIM standards and the Distributed Management Task Force (DMTF) specification provide information about Web-based enterprise management (WBEM) operations over HTTP.

When the CIM object manager (CIMOM) first starts, it registers itself to the SLP and provides information about its location (IP address and port) and the type of service it provides. A client application finds the location of the CIMOM by calling an SLP directory service. After obtaining this information, the client application opens direct communication with the CIMOM.

A client sends requests to a CIMOM in the context of a CIM model. The model is defined by the CIM schema and loaded into the repository of the CIMOM. Figure 2 on page 4 shows how the schema is loaded into the data store of the CIMOM. The managed object format (MOF) compilation and creation of the data store is managed automatically during installation.

As requests arrive, the CIMOM validates and authenticates each request. Requests are either directed to the appropriate functional component of the CIMOM or directed to a device-specific handler called a provider.
A provider makes device-unique programming interface calls on behalf of the CIMOM to satisfy a client application request. Such requests generally map a CIM request to the API for a device. A request to get an instance of a class or a property of an instance, for example, might be directed to a provider and a provider might make one or many requests of a device using the unique API for the device. Figure 1 on page 2 shows the communication structure between the device and the client application.

**CIM agent security**

The CIM agent can operate in both secure and unsecure modes.

**Secure mode**

All requests between the client application and the CIM object manager (CIMOM) are XML encoded requests sent over HTTP or HTTP over Secure Sockets Layer (SSL). The CIMOM, upon receiving a request, parses the request and processes it. Responses, when they are returned to the client application, are transformed into XML-encoded CIM status and returned in HTTP responses to the client. The CIM agent runs in secure mode using SSL by default.

**Unsecure mode**

Some vendor software cannot communicate with the CIM agent in a secure mode. You can still use this vendor software by configuring the CIM agent to run with only basic user name and password security. See the configuration instructions for your operating system for the instructions for configuring the CIM agent for this less secure mode.
Chapter 2. CIM agent for the XIV system

Beginning with the IBM XIV Storage System V10.1.0, the CIM agent is preinstalled on the administrative module. The embedded CIM agent is auto enabled and preconfigured. You can manage the XIV system from the CIM agent that is bundled with the administrative module.

The CIM agent that is embedded on the administrative module has the following limitations:
- The CIM agent can support only the XIV system on which the administrative module is located. The CIM agent is not able to manage any other XIV system.
- The CIM agent must use secure connections over port 5989.
- The CIM agent uses the XIV system user account to perform authentication. To manage accounts, you must use the IBM XIV Storage Management GUI or the IBM XIV command-line interface (XCLI).

CIM agent installation overview

Beginning with the IBM XIV Storage System V10.1.0, the CIM agent is preinstalled on the administrative module. The CIM object manager (CIMOM) is installed and running on all three administrative modules. The clients can connect to any of the administrative modules and the same results are provided to the CIMOM.

Configuring the CIM agent

For IBM XIV Storage System V10.1.0 and later, the embedded CIM agent does not require configuration changes to manage the XIV system.

About this task

For the IBM XIV Storage System V10.1.0, the SMI-S Agent user names and passwords are used to authenticate CIM client requests. This type of user is different than an XIV system user. The XIV system administrator must use the XCLI to create the SMI-S Agent user names and passwords that CIM clients can use to attach to the CIM agent.

For the IBM XIV Storage System V10.2.0 and later, the XIV system user names and passwords are used to authenticate CIM client requests. The XIV system administrator must create user names and passwords that CIM clients can use to attach to the CIM agent. Each user that is created can be in any group. Users that are created in the technicians user group cannot perform any functions. The XIV system administrator can use the XCLI or the IBM XIV Storage Management GUI to create user names and passwords.

Enabling the CIM agent

Beginning with the IBM XIV Storage System V10.1.0, the CIM agent is enabled by default. A watchdog process is monitoring the CIM agent to make sure that it is always running. There are no options to enable, disable, or restart the CIMOM process through the XCLI or IBM XIV Storage Management GUI.

Verifying the CIM agent connection

You must verify that the XIV system can connect to the CIM agent software.
Before you begin

The slptool is included in the DSCIMCLI package. You can download the DSCIMCLI package from the IBM CIM Agent for DS Open API site at http://www.ibm.com/support/docview.wss?rs=0&uid=ssg1S4000595.

Procedure

The following procedure verifies that the XIV system can connect to the CIM agent:

Verify that the CIM agent has registered into the SLP by issuing the slptool findsrvs command.

```
slptool findsrvs service:wbem
```

The output is a list of CIM agent services in the following form:

```
service:wbem:https://Admin module IP:5989,Timeout
```

where Admin module IP represents the IP address of the administrative module (IPv6 is supported in IBM XIV Storage System V11.0 and later releases) and Timeout is the number of seconds that remain before the entry times out of SLP.

What to do next

You can also use the IBM Tivoli Storage Productivity Center to check if the CIM agent is running on the device. If the discovery process fails, you can perform one of the following actions:

- Enable the Internet Group Management Protocol (IGMP) on the Ethernet router. The discovery process can fail because SLP requires that IGMP is enabled on the Ethernet router.
- Move the IBM Tivoli Storage Productivity Center server and the XIV system to the same Ethernet switch.
- Use IBM Tivoli Storage Productivity Center to add the device. The IBM Tivoli Storage Productivity Center can use the IP address of the device to add the device.
Chapter 3. XIV Open API component definitions

The XIV Open API elements include schemas, classes, properties, methods, indications, associations, references, and qualifiers.

Elements

The following list describes each type of element:

Schema
A group of classes defined to a single namespace. Within the CIM agent, the schemas that are supported are the ones loaded through the managed object format (MOF) compiler.

Class
The definition of an object within some hierarchy. Classes can have methods and properties and be the target of an association.

Property
A value that is used to characterize instances of a class.

Method
An implementation of a function on a class.

Indication
An object representation of an event.

Association
A class that contains two references which define a relationship between two objects.

Reference
A unique identifier of an object that is based on its key properties.

Qualifier
Additional information about other elements, classes, associations, indications, methods, method parameters, instances, properties, or references.

Namespace

XIV Open API operations always run within the context of a namespace. A namespace defines the scope over which an XIV Open API schema applies. An XIV Open API schema or version is loaded into a namespace when that schema is compiled by the MOF compiler. The namespace must be specified within the message that the client sends to the CIM agent.

Clients cannot create new namespaces. Attempts to do so result in errors.

Object name

An object name consists of a namespace path and a model path. The namespace path provides access to the XIV Open API implementation that is managed by the CIM agent. The model path provides navigation within the implementation. The following example shows an object name:

http://cimom.host.com/root/ibm:CIM_Class.key1=value1,key2=value2

where http://cimom.host.com/root/ibm is the namespace path and :CIM_Class.key1=value1,key2=value2 is the model path.
CIM agent communication concepts

Client communication

A client application communicates with the CIM agent through operation request messages encoded within XML. The CIM agent returns responses with operation response messages. Requests and responses are subelements of the <CIM MESSAGE> element. A <MESSAGE> sent to the CIM agent must contain an ID attribute. A response from the CIM agent returns this value and enables the client to track requests and their responses. The CIM agent supports simple requests and simple responses. Simple requests are operation request messages that contain the <SIMPLEREQ> XML tag. Simple responses are operation response messages that contain the <SIMPLERSP> XML tag. A client application determines that the CIM agent only supports simple operation requests and responses by examining the output of the OPTIONS method.

Intrinsic and Extrinsic Methods

All operations on the CIM agent are performed by running one or more methods. A method is either an intrinsic method or an extrinsic method.

Intrinsic methods are supported by the CIM agent itself. These methods are included within <IMETHODCALL> XML tags that are sent in messages to the CIM agent.

Extrinsic methods are defined by the schema that is supported by the CIM agent. These methods are included within <METHODCALL> XML tags that are sent in messages to the CIM agent. Client applications can call on the CIM agent using these methods. These methods fall within certain functional groups that might be supported by the CIM agent.

CIM agent communication methods

Client application calls to intrinsic methods result in CIM agent calls to the device provider if the device provider surfaces the classes or instances that are referenced in the calls.

The CIM agent returns IMETHODRESPONSE or METHODRESPONSE elements to the client application when the intrinsic or extrinsic methods are used. These elements are contained within a MESSAGERESPONSE XML tag.

Functional groups

Table 1 describes the functional groups that are supported by the CIM agent. This information is also returned to the client that makes an OPTIONS request to the CIM agent.

Table 1. Functional groups for the CIM agent

<table>
<thead>
<tr>
<th>Functional group</th>
<th>Parameters</th>
<th>Supported or not supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic read</td>
<td>GetInstance, EnumerateInstances, EnumerateInstanceNames, GetProperty</td>
<td>Supported</td>
</tr>
<tr>
<td>Basic write</td>
<td>SetProperty</td>
<td>Not Supported</td>
</tr>
<tr>
<td>Schema manipulation</td>
<td>CreateClass, ModifyClass, DeleteClass</td>
<td>Not Supported</td>
</tr>
</tbody>
</table>
Table 1. Functional groups for the CIM agent (continued)

<table>
<thead>
<tr>
<th>Functional group</th>
<th>Parameters</th>
<th>Supported or not supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance manipulation</td>
<td>CreateInstance</td>
<td>Not Supported</td>
</tr>
<tr>
<td></td>
<td>ModifyInstance</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DeleteInstance</td>
<td></td>
</tr>
<tr>
<td>Association traversal</td>
<td>Associators</td>
<td>Supported</td>
</tr>
<tr>
<td></td>
<td>AssociatorNames</td>
<td></td>
</tr>
<tr>
<td></td>
<td>References</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RefernceNames</td>
<td></td>
</tr>
<tr>
<td>Qualifier read</td>
<td>GetQualifier</td>
<td>Supported</td>
</tr>
<tr>
<td></td>
<td>EnumerateQualifiers</td>
<td></td>
</tr>
<tr>
<td>Qualifier manipulation</td>
<td>SetQualifier</td>
<td>Not Supported</td>
</tr>
<tr>
<td></td>
<td>DeleteQualifier</td>
<td></td>
</tr>
<tr>
<td>Query execution</td>
<td>ExecQuery</td>
<td>Supported</td>
</tr>
</tbody>
</table>

The most current information for the communication methods is in the MOF documentation. The MOF documentation is in the `mof` folder in the CIM agent installation directory.

**GetClass**

The GetClass method returns a single class from the target namespace.

**Parameters**

*Table 2* describes the parameters of the GetClass method.

*Table 2. GetClass method parameters*

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClassName</td>
<td>String</td>
<td>Defines the name of the class you want to retrieve.</td>
</tr>
<tr>
<td>LocalOnly</td>
<td>Boolean</td>
<td>TRUE returns all properties, methods, and qualifiers overridden within the definition of the class.</td>
</tr>
<tr>
<td>IncludeQualifiers</td>
<td>Boolean</td>
<td>TRUE returns all qualifiers for the class, its properties, methods, or method parameters. FALSE returns no qualifiers.</td>
</tr>
<tr>
<td>IncludeClassOrigin</td>
<td>Boolean</td>
<td>TRUE returns the CLASSORIGIN attribute of the class.</td>
</tr>
</tbody>
</table>

**Return values**

If successful, a single class is returned. Otherwise, one of the following error codes is returned:

- CIM_ERR_ACCESS_DENIED
- CIM_ERR_INVALID_NAMESPACE
- CIM_ERR_INVALID_PARAMETER
- CIM_ERR_FAILED
GetInstance
The GetInstance method returns a single instance from the target namespace.

Parameters
Table 3 describes the parameters of the GetInstance method.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InstanceName</td>
<td>String</td>
<td>Defines the name of the instance to retrieve.</td>
</tr>
<tr>
<td>IncludeClassOrigin</td>
<td>Boolean</td>
<td>TRUE returns the CLASSORIGIN attribute of the class.</td>
</tr>
</tbody>
</table>

Return values
If successful, a single class is returned. Otherwise, one of the following error codes is returned:
- CIM_ERR_ACCESS_DENIED
- CIM_ERR_INVALID_NAMESPACE
- CIM_ERR_INVALID_PARAMETER
- CIM_ERR_INVALID_CLASS
- CIM_ERR_NOT_FOUND
- CIM_ERR_FAILED

DeleteClass
The DeleteClass method deletes a single class from the target namespace.

The DeleteClass method cannot be used. The CIM_ERR_NOT_SUPPORTED error code is returned to the client application if a request to process this operation is received.

DeleteInstance
The DeleteInstance method deletes a single instance from the target namespace.

The DeleteInstance method is a standard CIM method. The XIV Open API does not have any features that use this method.

Parameters
Table 4 describes the parameters of the DeleteInstance method.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InstanceName</td>
<td>String</td>
<td>Defines the name of the instance you want to delete.</td>
</tr>
</tbody>
</table>

Return values
The named instance is deleted or one of the following error codes is returned:
- CIM_ERR_ACCESS_DENIED
- CIM_ERR_INVALID_NAMESPACE
CreateClass
The CreateClass method creates a new class from the target namespace.

The CreateClass method cannot be used. The CIM_ERR_NOT_SUPPORTED error code is returned to the client application if a request to process this operation is received.

CreateInstance
The CreateInstance method creates an instance in the target namespace. To use this method, the instance cannot exist.

The CreateInstance method is a standard CIM method. The XIV Open API does not have any features that use this method.

Parameters
Table 5 describes the parameters of the CreateInstance method.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance</td>
<td>Object</td>
<td>The instance to be created. The instance must be based on a class that is already defined in the target namespace.</td>
</tr>
</tbody>
</table>

Return values
If successful, the specified instance is created. Otherwise, one of the following error codes is returned:
- CIM_ERR_ACCESS_DENIED
- CIM_ERR_INVALID_NAMESPACE
- CIM_ERR_INVALID_PARAMETER
- CIM_ERR_NOT_SUPPORTED
- CIM_ERR_CANNOT_ADD_CHILD
- CIM_ERR_ALREADY_EXISTS
- CIM_ERR_FAILED

ModifyClass
The ModifyClass method modifies an existing class.

The ModifyClass method cannot be used. The CIM_ERR_NOT_SUPPORTED error code is returned to the client application if a request to process this operation is received.

ModifyInstance
The ModifyInstance method modifies an existing instance in the target namespace. The instance must exist.

The ModifyInstance method cannot be used. The CIM_ERR_NOT_SUPPORTED error code is returned to the client application if a request to process this operation is received.
Parameters

Table 6 describes the parameters of the ModifyInstance method.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance</td>
<td>Object</td>
<td>Defines the modified instance.</td>
</tr>
</tbody>
</table>

Return values

If successful, the specified instance is updated. Otherwise, one of the following error codes is returned:

- CIM_ERR_ACCESS_DENIED
- CIM_ERR_INVALID_NAMESPACE
- CIM_ERR_INVALID_PARAMETER
- CIM_ERR_INVALID_CLASS
- CIM_ERR_NOT_FOUND
- CIM_ERR_FAILED

EnumerateClasses

The EnumerateClasses method returns a single class from the target namespace.

Parameters

Table 7 describes the parameters of the EnumerateClasses method.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClassName</td>
<td>String</td>
<td>Defines the name of the class for which subclasses are to be returned. If this field is NULL, all base classes within the target namespace are returned.</td>
</tr>
<tr>
<td>DeepInheritance</td>
<td>Boolean</td>
<td>TRUE returns all subclasses of the specified class. FALSE returns only immediate child subclasses.</td>
</tr>
<tr>
<td>LocalOnly</td>
<td>Boolean</td>
<td>TRUE returns all properties, methods, and qualifiers, that are overridden within the definition of the class.</td>
</tr>
<tr>
<td>IncludeQualifiers</td>
<td>Boolean</td>
<td>TRUE returns all qualifiers for the class, its properties, methods, or method parameters. FALSE returns no qualifiers.</td>
</tr>
<tr>
<td>IncludeClassOrigin</td>
<td>Boolean</td>
<td>TRUE returns the CLASSORIGIN of the class.</td>
</tr>
</tbody>
</table>

Return values

If successful, zero or more classes (CIMClass) are returned. Otherwise, one of the following error codes is returned:

- CIM_ERR_ACCESS_DENIED
- CIM_ERR_INVALID_NAMESPACE
EnumerateClassNames
The EnumerateClassNames method enumerates the names of subclasses of a class defined within the target namespace.

Parameters

Table 8 describes the parameters of the EnumerateClassNames method.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClassName</td>
<td>String</td>
<td>Defines the name of the class for which subclass names are to be returned. If this field is NULL, all base class names within the target namespace are returned.</td>
</tr>
<tr>
<td>DeepInheritance</td>
<td>Boolean</td>
<td>TRUE returns all subclass names of the specified class. FALSE returns only immediate child subclass names.</td>
</tr>
</tbody>
</table>

Return values

Return values: If successful, zero or more class names are returned. Otherwise, one of the following error codes is returned:

- CIM_ERR_ACCESS_DENIED
- CIM_ERR_INVALID_NAMESPACE
- CIM_ERR_INVALID_PARAMETER
- CIM_ERR_INVALID_CLASS
- CIM_ERR_FAILED

EnumerateInstances
The EnumerateInstances method enumerates instances of a defined class in the target namespace.

Parameters

Table 9 describes the parameters of the EnumerateInstances method.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClassName</td>
<td>String</td>
<td>Defines the name of the class for which instances are to be returned.</td>
</tr>
<tr>
<td>DeepInheritance</td>
<td>Boolean</td>
<td>TRUE returns all instances and all properties of the instance, including the properties added by subclassing. FALSE returns only properties that are defined for the specified class.</td>
</tr>
<tr>
<td>IncludeClassOrigin</td>
<td>Boolean</td>
<td>TRUE returns the CLASSORIGIN attribute of the class within the instance.</td>
</tr>
</tbody>
</table>
Return values

If successful, zero or more instances (Objects) are returned. Otherwise, one of the following error codes is returned:

- CIM_ERR_ACCESS_DENIED
- CIM_ERR_INVALID_NAMESPACE
- CIM_ERR_INVALID_PARAMETER
- CIM_ERR_INVALID_CLASS
- CIM_ERR_FAILED

EnumerateInstanceNames

The EnumerateInstanceNames method enumerates the names of the instances of a class within a target namespace.

Parameters

Table 10 describes the parameter of the EnumerateInstanceNames method.

<table>
<thead>
<tr>
<th>Header</th>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClassName</td>
<td>String</td>
<td>Defines the name of the class for which instance names are returned.</td>
</tr>
</tbody>
</table>

Return values

If successful, zero or more names of instances are returned. Otherwise, one of the following error codes is returned:

- CIM_ERR_ACCESS_DENIED
- CIM_ERR_INVALID_NAMESPACE
- CIM_ERR_INVALID_PARAMETER
- CIM_ERR_INVALID_CLASS
- CIM_ERR_FAILED

ExecuteQuery

The ExecuteQuery method processes a query against the target namespace.

Parameters

Table 11 describes the parameters of the ExecuteQuery method.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QueryLanguage</td>
<td>String</td>
<td>Defines the query language in which the query parameter is expressed.</td>
</tr>
<tr>
<td>Query</td>
<td>String</td>
<td>Defines the query to be executed.</td>
</tr>
</tbody>
</table>
Return values

If successful, the method returns a table definition, followed by zero or more rows that correspond to the results of the query. Otherwise, one of the following error codes is returned:

- CIM_ERR_ACCESS_DENIED
- CIM_ERR_NOT_SUPPORTED
- CIM_ERR_INVALID_NAMESPACE
- CIM_ERR_INVALID_PARAMETER
- CIM_ERR_QUERY_LANGUAGE_NOT_SUPPORTED
- CIM_ERR_QUERY_FEATURE_NOT_SUPPORTED
- CIM_ERR_INVALID_QUERY
- CIM_ERR_FAILED

Associators

The Associators method enumerates classes or instances that are associated with a specific CIM Object.

Parameters

Table 12 describes the parameters of the Associators method.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectName</td>
<td>String</td>
<td>Defines the class name or instance name that is the source of the association.</td>
</tr>
<tr>
<td>AssocClass</td>
<td>String</td>
<td>If not NULL, indicates that all objects must be associated with the source object through an instance of this class or one of its subclasses.</td>
</tr>
<tr>
<td>ResultClass</td>
<td>String</td>
<td>If not NULL, indicates that all returned objects must be instances of this class or one of its subclasses or be this class.</td>
</tr>
<tr>
<td>Role</td>
<td>String</td>
<td>If not NULL, indicates that each return object must be associated with the source object through an association in which the source object plays the specified role. The name of the property in the association class that refers to the source object must match the value of this parameter.</td>
</tr>
<tr>
<td>ResultRole</td>
<td>String</td>
<td>If not NULL, indicates that each returned object must be associated with the source object through an association in which the return object plays the specified role. That is, the name of the property in the association class that refers to the returned object must match the value of this parameter.</td>
</tr>
<tr>
<td>IncludeClassOrigin</td>
<td>Boolean</td>
<td>TRUE returns the CLASSORIGIN attribute of the class.</td>
</tr>
</tbody>
</table>
Return values

If successful, zero or more classes (CIMClass) or instances (Objects) are returned. Otherwise, one of the following error codes is returned:

- CIM_ERR_ACCESS_DENIED
- CIM_ERR_INVALID_NAMESPACE
- CIM_ERR_INVALID_PARAMETER
- CIM_ERR_INVALID_CLASS
- CIM_ERR_FAILED

AssociatorNames

The AssociatorNames method enumerates the names of the classes or instances that are associated with a particular CIM object.

Parameters

Table 13 describes the parameters of the AssociatorNames method.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectName</td>
<td>String</td>
<td>Defines the class name or instance name that is the source of the association.</td>
</tr>
<tr>
<td>AssocClass</td>
<td>String</td>
<td>If not NULL, indicates that all returned object paths returned identify an object that is associated with the source object through an instance of this class or one of its subclasses.</td>
</tr>
<tr>
<td>ResultClass</td>
<td>String</td>
<td>If not NULL, indicates that all returned object paths must identify instances of this class or one of its subclasses or must be this class.</td>
</tr>
<tr>
<td>Role</td>
<td>String</td>
<td>If not NULL, the name of the property in the association class that refers to the source object must match the value of this parameter.</td>
</tr>
<tr>
<td>ResultRole</td>
<td>String</td>
<td>If not NULL, the name of the property in the association class that refers to the return object must match the value of this parameter.</td>
</tr>
</tbody>
</table>

Return values

If successful, zero or more class paths (CIMObjectPath) are returned. Otherwise, one of the following error codes is returned:

- CIM_ERR_ACCESS_DENIED
- CIM_ERR_INVALID_NAMESPACE
- CIM_ERR_INVALID_PARAMETER
- CIM_ERR_INVALID_CLASS
- CIM_ERR_FAILED
**References**

The References method enumerates the association objects that refer to a particular target class or instance.

**Parameters**

Table 14 describes the parameters of the References method.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectName</td>
<td>String</td>
<td>Defines the class name or instance name whose referring objects are to be returned.</td>
</tr>
<tr>
<td>ResultClass</td>
<td>String</td>
<td>If not NULL, indicates that all returned objects must be instances of this class or one of its subclasses or must be this class.</td>
</tr>
<tr>
<td>Role</td>
<td>String</td>
<td>If not NULL, must be a valid property name. Each returned object must refer to the target object through a property whose name matches the value of this parameter.</td>
</tr>
<tr>
<td>IncludeClassOrigin</td>
<td>Boolean</td>
<td>TRUE returns the CLASSORIGIN attribute of the class.</td>
</tr>
</tbody>
</table>

**Return values**

If successful, zero or more classes (CIMClass) or instances (Objects) are returned. Otherwise, one of the following error codes is returned:
- CIM_ERR_ACCESS_DENIED
- CIM_ERR_INVALID_NAMESPACE
- CIM_ERR_INVALID_PARAMETER
- CIM_ERR_INVALID_CLASS
- CIM_ERR_FAILED

**ReferenceNames**

The ReferenceNames method enumerates the association objects that refer to a particular target class or instance.

**Parameters**

Table 15 describes the parameters of the ReferenceNames method.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectName</td>
<td>String</td>
<td>Defines the class name or instance name whose referring objects are to be returned.</td>
</tr>
<tr>
<td>ResultClass</td>
<td>String</td>
<td>If not NULL, indicates that all returned objects must be instances of this class or one of its subclasses or must be this class.</td>
</tr>
</tbody>
</table>

Chapter 3. XIV Open API component definitions 17
Table 15. ReferenceNames method parameters (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Role</td>
<td>String</td>
<td>If not NULL, must be a valid property name. Each returned object must refer to the target object through a property whose name matches the value of this parameter.</td>
</tr>
</tbody>
</table>

Return values

If successful, the return value specifies the value of the requested property. Otherwise, one of the following error codes is returned:

- CIM_ERR_ACCESS_DENIED
- CIM_ERR_INVALID_NAMESPACE
- CIM_ERR_INVALID_PARAMETER
- CIM_ERR_INVALID_CLASS
- CIM_ERR_NOT_FOUND
- CIM_ERR_NO_SUCH_PROPERTY
- CIM_ERR_FAILED

GetProperty

The GetProperty method retrieves a single property value from an instance in the target namespace.

Parameters

Table 16 describes the parameters of the GetProperty method.

Table 16. GetProperty method parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InstanceName</td>
<td>String</td>
<td>Defines the name of the instance.</td>
</tr>
<tr>
<td>Property</td>
<td>String</td>
<td>The name of the property whose value is to be returned from the instance.</td>
</tr>
</tbody>
</table>

Return values

If successful, the return value specifies the value of the requested property. Otherwise, one of the following return codes is returned:

- CIM_ERR_ACCESS_DENIED
- CIM_ERR_INVALID_NAMESPACE
- CIM_ERR_INVALID_PARAMETER
- CIM_ERR_INVALID_CLASS
- CIM_ERR_NOT_FOUND
- CIM_ERR_NO_SUCH_PROPERTY
- CIM_ERR_FAILED
SetProperty
The SetProperty method sets a single property value within an instance in the target namespace.

The XIV CIM agent does not have any features that use this method.

Parameters

Table 17 describes the parameters of the SetProperty method.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InstanceName</td>
<td>String</td>
<td>Defines the name of the instance.</td>
</tr>
<tr>
<td>Property</td>
<td>String</td>
<td>The name of the property whose value is to be returned from the instance.</td>
</tr>
</tbody>
</table>

Return values

If successful, the instance is updated. Otherwise, one of the following return codes is returned:

- CIM_ERR_ACCESS_DENIED
- CIM_ERR_INVALID_NAMESPACE
- CIM_ERR_INVALID_PARAMETER
- CIM_ERR_INVALID_CLASS
- CIM_ERR_NOT_FOUND
- CIM_ERR_NO_SUCH_PROPERTY
- CIM_ERR_TYPE_MISMATCH
- CIM_ERR_FAILED

GetQualifier
The GetQualifier method retrieves a single qualifier declaration from the target namespace.

Parameters

Table 18 describes the parameters of the GetQualifier method.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QualifierName</td>
<td>String</td>
<td>Defines the qualifier whose declaration is to be returned.</td>
</tr>
</tbody>
</table>

Return values

If successful, the value of the qualifier is returned. Otherwise, one of the following return codes is returned:

- CIM_ERR_ACCESS_DENIED
- CIM_ERR_INVALID_NAMESPACE
- CIM_ERR_INVALID_PARAMETER
- CIM_ERR_NOT_FOUND
- CIM_ERR_FAILED
SetQualifier
The SetQualifier method creates or updates a qualifier declaration in the target namespace.

Parameters
Table 19 describes the parameters of the SetQualifier method.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QualifierDeclaration</td>
<td>Void</td>
<td>Defines the qualifier declaration to be added to the target namespace.</td>
</tr>
</tbody>
</table>

Return values
If successful, the qualifier is updated in the target namespace. Otherwise, one of the following error codes is returned:

- CIM_ERR_ACCESS_DENIED
- CIM_ERR_INVALID_NAMESPACE
- CIM_ERR_INVALID_PARAMETER
- CIM_ERR_NOT_FOUND
- CIM_ERR_FAILED

DeleteQualifier
The DeleteQualifier method deletes a single class from the target namespace.

The DeleteQualifier method cannot be used. The CIM_ERR_NOT_SUPPORTED error code is returned to the client application if a request to process this operation is received.

EnumerateQualifiers
The EnumerateQualifiers method enumerates qualifier declarations from the target namespace.

There are no parameters for this method.

Return values
If successful, zero or more qualifier declarations are returned. Otherwise, one of the following error codes is returned:

- CIM_ERR_ACCESS_DENIED
- CIM_ERR_INVALID_NAMESPACE
- CIM_ERR_INVALID_PARAMETER
- CIM_ERR_FAILED

Return error codes
The CIMOM returns status to the client application.

The return status is sent to the client application in of the following thing ways:

- Through HTTP status messages
- Through error codes contained within <METHODRESPONSE> or <IMETHODRESPONSE> XML tags.
Table 20 describes the vendor-specific status codes that the CIMOM might return. For CIM standard return codes, see the CIM schema.

Table 20. Return error codes for the CIMOM

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbolic Name</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CIM_ERR_FAILED</td>
<td>A general error occurred that is not covered by a more specific error code.</td>
</tr>
<tr>
<td>2</td>
<td>CIM_ERR_ACCESS_DENIED</td>
<td>Access to a CIM resource was not available to the client.</td>
</tr>
<tr>
<td>3</td>
<td>CIM_ERR_INVALID_NAMESPACE</td>
<td>The target namespace does not exist.</td>
</tr>
<tr>
<td>4</td>
<td>CIM_ERR_INVALID_PARAMETER</td>
<td>One or more parameter values passed to the method were not valid.</td>
</tr>
<tr>
<td>5</td>
<td>CIM_ERR_INVALID_CLASS</td>
<td>The specified class does not exist.</td>
</tr>
<tr>
<td>6</td>
<td>CIM_ERR_NOT_FOUND</td>
<td>The requested object could not be found.</td>
</tr>
<tr>
<td>7</td>
<td>CIM_ERR_NOT_SUPPORTED</td>
<td>The requested operation is not supported.</td>
</tr>
<tr>
<td>8</td>
<td>CIM_ERR_CLASS_HAS_CHILDREN</td>
<td>The operation cannot be carried out on this class because it has children.</td>
</tr>
<tr>
<td>9</td>
<td>CIM_ERR_CLASS_HAS_INSTANCES</td>
<td>The operation cannot be carried out on this class because it has instances.</td>
</tr>
<tr>
<td>10</td>
<td>CIM_ERR_INVALID_SUPERCLASS</td>
<td>The operation cannot be carried out because the specified superclass does not exist.</td>
</tr>
<tr>
<td>11</td>
<td>CIM_ERR_ALREADY_EXISTS</td>
<td>The operation cannot be carried out because the object exists.</td>
</tr>
<tr>
<td>12</td>
<td>CIM_ERR_NO_SUCH_PROPERTY</td>
<td>The specified property does not exist.</td>
</tr>
<tr>
<td>13</td>
<td>CIM_ERR_TYPE_MISMATCH</td>
<td>The value supplied is not compatible with the type that is specified.</td>
</tr>
<tr>
<td>14</td>
<td>CIM_ERR_QUERY_LANGUAGE_NOT_SUPPORTED</td>
<td>The query language is not recognized or supported.</td>
</tr>
<tr>
<td>15</td>
<td>CIM_ERR_INVALID_QUERY</td>
<td>The query is not valid for the specified query language.</td>
</tr>
<tr>
<td>16</td>
<td>CIM_ERR_METHOD_NOT_AVAILABLE</td>
<td>The extrinsic method cannot be run.</td>
</tr>
<tr>
<td>17</td>
<td>CIM_ERR_METHOD_NOT_FOUND</td>
<td>The specified extrinsic method does not exist.</td>
</tr>
<tr>
<td>20</td>
<td>CIM_ERR_LOW_ON_MEMORY</td>
<td>There is not enough memory.</td>
</tr>
<tr>
<td>21</td>
<td>XMLERROR</td>
<td>An XML error has occurred.</td>
</tr>
<tr>
<td>22</td>
<td>CIM_ERR_LISTENER_ALREADY_DEFINED</td>
<td>The listener is already defined.</td>
</tr>
<tr>
<td>23</td>
<td>CIM_ERR_INDICATION_NOT_COLLECTED</td>
<td>The indications are not collected.</td>
</tr>
<tr>
<td>24</td>
<td>CIM_ERR_NO_METHOD_NAME</td>
<td>The method name is null.</td>
</tr>
<tr>
<td>25</td>
<td>CIM_ERR_INVALID_QUALIFIER_DATATYPE</td>
<td>The datatype qualifier is not valid.</td>
</tr>
<tr>
<td>26</td>
<td>CIM_ERR_NAMESPACE_NOT_IN_MANAGER</td>
<td>The namespace value is not found.</td>
</tr>
<tr>
<td>27</td>
<td>CIM_ERR_INSTANTIATE_FAILED</td>
<td>The instantiation failed.</td>
</tr>
<tr>
<td>28</td>
<td>CIM_ERR_FAILED_TO_LOCATE_INDICATION_HANDLER</td>
<td>The indication handler is not found.</td>
</tr>
<tr>
<td>29</td>
<td>CIM_ERR_IO_EXCEPTION</td>
<td>An I/O exception has occurred.</td>
</tr>
<tr>
<td>30</td>
<td>CIM_ERR_COULD_NOT_DELETE_FILE</td>
<td>The file could not be deleted.</td>
</tr>
<tr>
<td>31</td>
<td>INVALID_QUALIFIER_NAME</td>
<td>The qualifier name is null.</td>
</tr>
<tr>
<td>32</td>
<td>NO_QUALIFIER_VALUE</td>
<td>The qualifier value is null.</td>
</tr>
<tr>
<td>Code</td>
<td>Symbolic Name</td>
<td>Definition</td>
</tr>
<tr>
<td>------</td>
<td>------------------------------------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td>33</td>
<td>NO_SUCH_QUALIFIER1</td>
<td>There is no such qualifier.</td>
</tr>
<tr>
<td>34</td>
<td>NO_SUCH_QUALIFIER2</td>
<td>There is no such qualifier.</td>
</tr>
<tr>
<td>35</td>
<td>QUALIFIER_UNOVERRIDABLE</td>
<td>The qualifier is cannot be overwritten.</td>
</tr>
<tr>
<td>36</td>
<td>SCOPE_ERROR</td>
<td>A scope error has occurred.</td>
</tr>
<tr>
<td>37</td>
<td>TYPE_ERROR</td>
<td>A type error has occurred.</td>
</tr>
<tr>
<td>38</td>
<td>CIM_ERR_MISSING_KEY</td>
<td>The key is missing.</td>
</tr>
<tr>
<td>39</td>
<td>CIM_ERR_KEY_CANNOT_MODIFY</td>
<td>The key cannot be modified.</td>
</tr>
<tr>
<td>40</td>
<td>CIM_ERR_NO_KEYS</td>
<td>There are no keys found.</td>
</tr>
<tr>
<td>41</td>
<td>CIM_ERR_KEYS_NOT_UNIQUE</td>
<td>The keys are not unique.</td>
</tr>
<tr>
<td>100</td>
<td>CIM_ERR_SET_CLASS_NOT_SUPPORTED</td>
<td>The set class operation is not supported.</td>
</tr>
<tr>
<td>101</td>
<td>CIM_ERR_SET_INSTANCE_NOT_SUPPORTED</td>
<td>The set instance operation is not supported.</td>
</tr>
<tr>
<td>102</td>
<td>CIM_ERR_QUALIFIER_NOT_FOUND</td>
<td>The qualifier value is not found.</td>
</tr>
<tr>
<td>103</td>
<td>CIM_ERR_QUALIFIERTYPE_NOT_FOUND</td>
<td>The qualifier type is not found.</td>
</tr>
<tr>
<td>104</td>
<td>CIM_ERR_CONNECTION_FAILURE</td>
<td>The connection failed.</td>
</tr>
<tr>
<td>105</td>
<td>CIM_ERR_FAIL_TO_WRITE_TO_SERVER</td>
<td>There is a fail to write to the server.</td>
</tr>
<tr>
<td>106</td>
<td>CIM_ERR_SERVER_NOT_SPECIFIED</td>
<td>The server is not specified.</td>
</tr>
<tr>
<td>107</td>
<td>CIM_ERR_INDICATION_ERROR</td>
<td>There is an indication processing error.</td>
</tr>
<tr>
<td>108</td>
<td>CIM_ERR_FAIL_TO_WRITE_TO_CIMOM</td>
<td>A write operation to the CIMOM has failed.</td>
</tr>
<tr>
<td>109</td>
<td>CIM_ERR_SUBSCRIPTION_EXISTS</td>
<td>A subscription exists.</td>
</tr>
<tr>
<td>110</td>
<td>CIM_ERR_INVALID_SUBSCRIPTION_DEST</td>
<td>The subscription destination is not valid.</td>
</tr>
<tr>
<td>111</td>
<td>CIM_ERR_INVALID_FILTER_PATH</td>
<td>The filter path is not valid.</td>
</tr>
<tr>
<td>112</td>
<td>CIM_ERR_INVALID_HANDLER_PATH</td>
<td>The handler path is not valid.</td>
</tr>
<tr>
<td>113</td>
<td>CIM_ERR_NO_FILTER_INSTANCE</td>
<td>The filter instance is not found.</td>
</tr>
<tr>
<td>114</td>
<td>CIM_ERR_NO_HANDLER_INSTANCE</td>
<td>The handler instance is not found.</td>
</tr>
<tr>
<td>115</td>
<td>CIM_ERR_UNSUPPORTED_FILTER</td>
<td>The filter that is referenced in the subscription is not supported.</td>
</tr>
<tr>
<td>116</td>
<td>CIM_ERR_INVALID_TRUSTSTORE</td>
<td>The CIMOM cannot be connected to because there is a bad or missing truststore or an incorrect truststore password.</td>
</tr>
<tr>
<td>117</td>
<td>CIM_ERR_ALREADY_CONNECTED</td>
<td>The CIMOM cannot be connected to because it is already connected.</td>
</tr>
<tr>
<td>118</td>
<td>CIM_ERR_UNKNOWN_SERVER</td>
<td>The server is unknown. The CIMOM cannot accept connections.</td>
</tr>
<tr>
<td>119</td>
<td>CIM_ERR_INVALID_CERTIFICATE</td>
<td>The correct certificate cannot be found in the truststore. The CIMOM cannot accept connections.</td>
</tr>
</tbody>
</table>
Chapter 4. Functional diagrams of the CIM agent

The functional diagrams of the CIM agent object model show specific functions that the CIM agent provides. The diagrams also illustrate the architecture of the CIM agent for the XIV system.

Block Server performance profile

Block server performance is the SMI-S subprofile that describes how to present performance statistics, which uses a number of metrics with definitions that have been standardized for all storage systems that are represented by SMI-S. These metrics are defined in the CIM_BlockStorageStatisticalData class definition.

For the XIV CIM Agent, statistics are only provided for the volumes.

Table 21 provides details about the specific metrics that are supported by the XIV arrays and their components. In addition, the table provides a list of ElementType values that correspond to the components of XIV arrays.

Table 21. Block Server metrics

<table>
<thead>
<tr>
<th>Equivalent CIM class</th>
<th>Corresponding ElementType value</th>
<th>Properties supplied in associated CIM_BlockStorageStatisticalData instances</th>
</tr>
</thead>
</table>
| Volume IBMTSDS_SEVolume | 8                              | StatisticTime
|                       |                                 | TotalIOs
|                       |                                 | KBytesTransferred
|                       |                                 | KBytesRead
|                       |                                 | KBytesWritten
|                       |                                 | ReadIOs
|                       |                                 | ReadHitIOs
|                       |                                 | WriteIOs
|                       |                                 | WriteHitIOs
|                       |                                 | IOTimeCounter
|                       |                                 | ReadIOTimeCounter
|                       |                                 | WriteIOTimeCounter
|                       |                                 | ReadHitIOTimeCounter
|                       |                                 | WriteHitIOTimeCounter |

Block Server Performance object model

Two major categories of classes in the object model determine the way clients retrieve performance statistics.

The following categories are used:

- The first category is a set of classes where each instance of the class represents a single performance statistics record. For example, the statistics for a single volume.
- The second category is a set of classes that are required to use an extrinsic method to retrieve a string that contains a batch of performance statistics data.
The classes inheriting from CIM_BlockStorageStatisticalData IBMTSDS_SEVolumeStatistics is of the first category, where each instance contains properties that describe the performance statistics of a single instance of the IBMTSDS_SEVolume. IBMTSDS_PerformanceStatisticsCollection does not fall under either of the categories, but is a class that associates all of the Block Server Performance-related classes for a particular system.

The rest of the classes are under the second category. They are used for batch gathering of statistics. IBMTSDS_PerformanceStatisticsService contains the GetStatisticsCollection method, which you use to get a string representation of a batch of statistics instances. IBMTSDS_PerformanceStatisticsCapabilities defines what the performance statistics-related capabilities of the CIM agent are.

The classes that inherit from CIM_BlockStatisticsManifest define filters for the statistics returned by GetStatisticsCollection. IBMTSDS_BlockStatisticsManifestCollection is a group of BlockStatisticsManifests that gets passed into the GetStatisticsCollection method. The BlockStatisticsManifest classes contain Boolean properties, one for each property in the corresponding BlockStorageStatisticalData class. If a BlockStorageStatisticalData had a statistics property XXX, then the corresponding BlockStatisticsManifest class would have a boolean property called IncludeXXX. If IncludeXXX was set to true, GetStatisticsCollection would return the data for XXX. If IncludeXXX was set to false, GetStatisticsCollection would not return the data for XXX.

The XIV CIM agent does not allow clients to pick and choose which properties it is interested in, therefore all of the IncludeXXX properties for the XXX properties that the CIM agent supports are set to true.
All statistics attributes are in units of kilobytes, milliseconds, or just a count. For example, the number of I/O operations. However, the statistics are just running counters. They do not provide information on rates. For example, I/O operations per second. When the counters reach an internal limit, they roll back to zero. If the client application is monitoring these statistics at a constant rate in order to calculate I/O rates, it must know when the counters roll back to zero.

**IBMTSDS_PerformanceStatisticsService. GetStatistics Collection**


**ElementTypes (required)**

An array of values that indicates the type of element that returns statistics. Volumes use the standard value in the CIM schema. See BlockServerPerformance.mof for details.

**ManifestCollection (required)**

A reference to an instance of the collection that represents a collection of BlockStatisticsManifests to use to filter the output. This parameter is not a reference to BlockStatisticsManifest. This is a ManifestCollection because the method can return statistics for multiple ElementTypes at once, and each ElementType has its own BlockStatisticsManifest. The CIM agent currently only supplies one instance of ManifestCollection, which is the default ManifestCollection.

**Statistics**

An output parameter that is an array of strings that represents a batch of performance data. Each array element is an instance of a statistics class. Each array element is formatted as a semicolon-separated list of values. The order of the returned values matches the properties definition order in the corresponding CIM_BlockStatisticsManifest class in mof. For example, the statistics output parameter might display as follows:

```
IBM.2810-6000095-100916;8;20110118225313.890995+480;3111;166920;20864419;1426;
1422;1119849;1078071;67382;1685;19744570;19692176;99538;
IBM.2810-6000095-100930;8;20110118225313.891038+480;8386;504537;76662400;1426;
1422;1182723;1133669;67382;6940;6960;6940;75479677;75360317;437155;
```

where each row is an array element. The ElementType of each row is 8, which refers to Volumes. This means that there are two instances of VolumeStatistics, and the attributes of each instance are in the semicolon-separated strings, in the order they are defined in the IBMTSDS_SEVolumeManifest class in BlockServerPerformance.mof.

```
{ StatisticTime;TotalIOs;KBytesTransferred;IOTimeCounter;ReadIOs;ReadHitIOs;
  ReadIOTimeCounter;ReadHitIOTimeCounter;KBytesRead;WriteIOs;WriteHitIOs;
  WriteIOTimeCounter;WriteHitIOTimeCounter;KBytesWritten; }
```

**Getting volume statistics with GetStatisticsCollection**

You can use the GetStatisticsCollection method to configure a device.

**About this task**

This procedure provides an example recipe for configuring an XIV device. There are other ways you can use the XIV Open API to configure XIV devices.

**Procedure**

1. enumerateInstanceNames on IBMTSDS_PerformanceStatisticsService and then save the reference.
2. enumerateInstancesNames on IBMTSDS_BlockStatisticsManifestCollection and then save the reference.
3. enumerateInstances on IBMTSDS_VolumeManifest (or CIM_BlockStatisticsManifest and look for ElementType 8), and then save BulkFormat.

5. Check the output Statistics. For each array element in the statistics output, string tokenize on semicolon, and save the properties in a table. Check the properties.

---

**Block Services profile**

The XIV system architecture allows you to customize the underlying resources.

The following definitions describe the various layers of abstraction that make this customization possible:

**Primordial StoragePool**
Logical entities that contain all the available unformatted or unprepared disk capacity on an array. Device Concrete Storage Pools are allocated from the Primordial StoragePool. XIV arrays have only one Primordial StoragePool per array.

**Concrete StoragePools**
Logical entities that are allocated from the Primordial StoragePool. Concrete StoragePools enable storage administrators to manage relationships between volumes and snapshots and to define separate capacity provisioning and snapshot requirements for separate applications and departments. Storage pools are not tied to specific physical resources, nor are they part of the data-distribution scheme. XIV supports two types of concrete storage pools: VirtualPool and SnapshotPool. VirtualPool is allocated from the Primordial StoragePool directly and contains volumes and at most one SnapshotPool; SnapshotPool is allocated from VirtualPool and contains snapshots of volumes in the VirtualPool in which this SnapshotPool is located.

**Storage Volumes**
Logical units that can be mapped to a host or a cluster. Volumes are created from storage pools and are managed within the context of storage pools.

For the XIV system, volumes do not occupy physical capacity when created. Space is allocated when data is written to the volume. Moreover, volumes are not exclusively associated with a subset of physical resources, nor is there a permanent static relationship between logical volumes and specific physical resources.

See [IBM XIV Storage System: Architecture, Implementation, and Usage](http://www.redbooks.ibm.com) on the IBM Redbooks website (www.redbooks.ibm.com) and the [IBM XIV Storage System Product Overview](http://www.redbooks.ibm.com) for more information about the architecture of the XIV system.

**Block Services object model**

The CIM agent is designed to allow for the retrieval of information for each layer of abstraction.

For release 10.2.2 and earlier, the XIV Open API provides a read-only implementation of the Block Services package. This model supports read-only information about the capacity assignment and allocation. You cannot create, delete, or modify services.

For release 10.2.4, SMI-S specification 1.4 is fully supported. You can create, delete, or modify services.
As required by SMI-S, each instance of storage pool has a corresponding instance of Storage Capabilities. Each instance of Storage Capabilities is associated with the instances of Storage Setting that are valid for creating volumes from the associated storage pool.

Figure 4. Block Services SMI-S model for XIV systems
A StorageSetting is a class that contains properties to specify the Quality of Service, such as DataRedundancy or parity layout. When creating a Storage Volume or a Concrete StoragePool, a StorageSetting is supplied as the Goal parameter for the appropriate method.

XIV CIM Agent has three CIM_StorageSetting instances defined:
• **IBMTSDS_VirtualPoolSetting.InstanceID="IBMTSDS:IBM XIV Virtual Storage Pool Setting"**: This StorageSetting is supplied as the Goal parameter for the CreateOrModifyStoragePool method to create or modify VirtualPool.

• **IBMTSDS_SnapshotPoolSetting.InstanceID=" IBMTSDS:IBM XIV Snapshot Pool Setting ":** This StorageSetting is supplied as the Goal parameter for the CreateOrModifyStoragePool method to create or modify SnapshotPool.

• **IBMTSDS_DataTypeSetting.InstanceID="IBMTSDS:XIVBlockSize":** This StorageSetting is supplied as the Goal parameter for the CreateOrModifyElementFromStoragePool and CreateOrModifyElementsFromStoragePool method to create or modify Storage Volumes with size in terms of GB, which means that the system will allocate the soft volume size as the minimum number of discrete 17 GB increments needed to meet the requested volume size.

• **IBMTSDS_DataTypeSetting.InstanceID="IBMTSDS:SystemBlockSize":** This StorageSetting is supplied as the Goal parameter for the CreateOrModifyElementFromStoragePool and CreateOrModifyElementsFromStoragePool method to create or modify Storage Volumes with size in terms of blocks, which means that the volume's capacity is indicated as a discrete number of 512 byte blocks. The system will still allocate the soft volume size consumed within the Storage Pool as the minimum number of discrete 17 GB increments needed to meet the requested size (specified in 512 byte blocks); however, the size that is reported to hosts is equivalent to the precise number of blocks defined.

See the BlockServices.mof file for more information about the block services object model.

**Extrinsic Methods**

Use an associator’s CIM client request and the association class CIM_ElementCapabilities between an instance of a CIM_StorageConfigurationService and the result class CIM_StorageConfigurationCapabilities. The SupportedSynchronousActions and SupportedAsynchronousActions properties identifies whether the action is performed synchronously or asynchronously. XIV CIM Agent supports the following synchronous actions in IBMTSDS_StorageConfigurationService class:

<table>
<thead>
<tr>
<th>Synchronous actions</th>
<th>XIV CIM functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>StoragePool Creation</td>
<td>CreateOrModifyStoragePool</td>
</tr>
<tr>
<td>StoragePool Modification</td>
<td></td>
</tr>
<tr>
<td>StoragePool Deletion</td>
<td>DeleteStoragePool</td>
</tr>
<tr>
<td>StorageVolume Creation</td>
<td>CreateOrModifyElementFromStoragePool</td>
</tr>
<tr>
<td>StorageVolume Modification</td>
<td>CreateOrModifyElementsFromStoragePool</td>
</tr>
<tr>
<td>StorageVolume Expansion</td>
<td></td>
</tr>
<tr>
<td>StorageVolume Shrinking</td>
<td></td>
</tr>
<tr>
<td>StorageVolume Deletion</td>
<td>Return to StoragePool</td>
</tr>
<tr>
<td></td>
<td>Return Elements to StoragePool</td>
</tr>
</tbody>
</table>

The following functions can be used to determine what sizes of Concrete StoragePool and StorageVolumes can be created:

**IBMTSDS_PrimordialStoragePool.GetSupportedSizeRange** 
Determines the supported physical size of Concrete StoragePools that can be created in Primordial StoragePool.
IBMTSDS_VirtualPool.GetSupportedSizeRange
Determines the supported soft size of StorageVolumes that can be created in a Concrete StoragePool.

IBMTSDS_StorageConfigurationService.CreateOrModifyStoragePool
The CreateOrModifyStoragePool method creates or modifies a VirtualPool or a SnapshotPool. The Primordial StoragePool cannot be created or modified.

Pool (required for modification)
As an input parameter, this specifies whether you want to create or modify a pool. If you specify a reference to a pool, this indicates that you want to modify the pool. If the parameter is left null, this indicates that you want to create a new pool.

InPools
InPools specifies from which pool to create the new pool. To create a VirtualPool, only the object reference of IBMTSDS_PrimordialStoragePool is permitted as the only one item of this parameter if specified, because all VirtualPools are created in the Primordial StoragePool; to create a SnapshotPool, only the object reference of IBMTSDS_VirtualPool is permitted as the only one item of this parameter if specified, because SnapshotPool is created in a VirtualPool.

Note: The CIM schema defines this input parameter to be an array of strings that represent CIM Object Paths (COPs), and not actual references to objects.

Goal
The Goal represents the StorageSetting of the pool to be created. To create a VirtualPool, only the object reference of IBMTSDS_VirtualPoolSetting is permitted if specified; to create a SnapshotPool, only the object reference of IBMTSDS_SnapshotPoolSetting is permitted if specified.

ElementName
The ElementName property allows you to set a user-friendly name for the pool being created or modified. If specified, it should be limited to 63 characters and may contain letters, digits, blanks, ",", "," and "~" characters. And blanks should not be the beginning and ending characters. If not specified during pool creation, a random pool name will be generated in form of "pool<randon integer>.”

Note: The name of the Pool must be unique in the system, meaning it cannot be a name that is already assigned to one of the other Pools in the system.

Size (required for creation)
As an input parameter, the Size specifies the requested hard size of the Pool. Null or 0 is not permitted for pool creation. 0 for pool modification means that the pool’s size will not be changed. As an output parameter, Size specifies the hard size achieved. If it is not possible to create pool of the desired size, a size is returned with size value set to the nearest supported size.

VirtualSize
As an input parameter, the VirtualSize specifies the desired soft size. The VirtualSize specifies the requested soft size of the virtual pool. If this parameter is specified, its value should be larger than the Size parameter, and in this case, the pool will be a thin provisioning pool; if it is left null, its value will be set as the value of Size parameter, and the pool will be a non-thin provisioning pool. As an output parameter, VirtualSize specifies the soft size achieved. If it is not possible to create pool of the desired VirtualSize, a size is returned with value set to the nearest supported VirtualSize.

Note: The VirtualSize parameter is not in the CIM or SMI-S schema. This parameter is an IBM extension.

IBMTSDS_StorageConfigurationService.DeleteStoragePool
DeleteStoragePool only requires a single parameter. Specify the reference to the Concrete StoragePool that is to be deleted. Only a single StoragePool can be deleted at a time.
You must delete all volumes in the current pool before deleting a pool. The Primordial StoragePool cannot be deleted.

**IBMTSDS_StorageConfigurationService.**
**CreateOrModifyElementsFromStoragePool**

CreateOrModifyElementsFromStoragePool is a vendor-extension method that allows for the creation / modification / moving of multiple volumes in a single method call.

For large numbers of volumes, this method may be more efficient than calling CreateOrModifyElementFromStoragePool several times in a loop. The InPool, Goal, and ElementType input parameters are the same as in CreateOrModifyElementFromStoragePool.

The following definitions summarize the parameters that are different from CreateOrModifyElementFromStoragePool.

**TheElements (required for volume modification or volume moving)**
If this value is not null, it indicates that you want to modify each of the volumes specified.

**Quantity (required for volume creation)**
As input, represents the number of StorageVolumes to be created. This parameter must be left as null when modifying StorageVolumes. As output, represents the number of volumes created if successful. Or creation was not successful due to too many volumes and not enough capacity on the VirtualPool, represents the number of volumes that can be created.

**ElementNames**
An array of the ElementNames to assign to the various volumes that are being created or modified. Volumes names can be supplied in two different ways:

* The first way supplies each volume name with one element of the value. In this case, its length must be equal to Quantity when creating volumes; if not NULL, its length must be same as length of TheElements when modifying volumes. For example, in order to create two volumes 'testVolume_3' and 'testVolume_4' in this way, set the parameter's value as ['testVolume_3', 'testVolume_4'] and leave the value of FirstSuffix as null.

* The second way supplies the leading string of volumes names as the only one element. In this case, names of all volumes will have the same format [LeadingString]_[IncrementingNumber]. The [LeadingString] will be the value of this parameter and the first [IncrementingNumber] will be value of the FirstSuffix parameter. The incremental number is 1. For example, in order to create two volumes 'testVolume_3' and 'testVolume_4' in the first way, set this parameter's value as ['testVolume'] and set value of FirstSuffix parameter as 3.

**FirstSuffix**
A uint16 that represents the starting suffix number of volume names when using the second way described above to create volumes or modify volume names. Users can supply the starting [IncrementingNumber] with this parameter. If it is left as null, the default value is 1. When using the first way, it must be left as null.

**ReturnCodes**
Because each volume that is created or modified can have a different error associated with it, this output parameter is an array of each of the individual return codes for each attempt.

**IBMTSDS_StorageConfigurationService.**
**CreateOrModifyElementFromStoragePool**

CreateOrModifyElementFromStoragePool is the CIM and SMI-S standard method for creating or modifying a StorageVolume from a Concrete StoragePool.
This method can only create or modify one volume at a time. This method is also used for moving a volume from one pool to another when the InPool parameter is specified with reference of the target pool.

The following definitions summarize how the method will treat different values for the input parameters:

**TheElement (required for modification)**
As an input parameter, if the value is null, it indicates that you want to create a new volume. If the value is not null, it indicates that you are trying to modify the specified volume. As an output parameter, TheElement references the volume that was created or modified.

**ElementType (required for creation)**
An enumeration indicating what type of element being created or modified. Only volume is supported, so the value of ElementType must be 2 (StorageVolume).

**InPool (required for creation and moving)**
A reference to the StoragePool which the volume is to be created in or moved to. This must be an instance of IBMTSDS_VirtualPool.

**Goal**
A reference to the StorageSetting representing the DataType of the volume being created. This must be a reference to an instance of IBMTSDS_DataTypeSetting. IBMTSDS_DataTypeSetting has two instance IBMTSDS:SystemBlockSize and IBMTSDS:XIVBlockSize. In volume creation, if this parameter is left null, the default value IBMTSDS:XIVBlockSize will be used. In volume modification, if the Size parameter is specified but this parameter is left null, the default value IBMTSDS:XIVBlockSize will be used and volume will be resized.

**Size**
In creating or modifying a volume, this parameter specifies the desired size (as input) and the size achieved (as an output parameter). In modifying the size of a volume, only size expansion is supported by default; if you want to shrink a volume, you need set the ForceShrink parameter as true as well.

**ForceShrink**
In modifying a volume, this parameter specifies a volume is shrunk if the Size parameter is smaller than the actual size of the volume. If this parameter is left null or set to false, shrinking a volume fails. To shrink a volume, set this parameter to true. The default value is false.

**Locked**
In modifying a volume, this parameter specifies the volume's locking status to be locked or unlocked. If this parameter is left null, the volume's locking status will not be modified; if this parameter is set to 0, the volume will be unlocked; if this parameter is set to 1, the volume will be locked. Other values are invalid. When volume is to be locked as well as other properties are to be modified in the same method invoke, volume will be locked in the last step; when volume is to be unlocked as well as other properties are to be modified in the same method invoke, volume will be unlocked in the first step.

**Note:** The Locked parameter is not in the CIM or SMI-S schema. This parameter is an IBM extension.

**ElementName**
A user-friendly name to assign to the volume.

**Note:** The name of the volume must be unique in the system, meaning it cannot be a name that is already assigned to one of the other Volumes in the system.
**IBMTSDS_StorageConfigurationService. ReturnToStoragePool**

This method is used to delete a volume. The only input parameter required is a reference to the volume that you want to delete. Only one volume can be deleted at a time.

**IBMTSDS_StorageConfigurationService. ReturnElementsToStoragePool**

This method is used to delete multiple volumes concurrently. You can pass in an array of references of volumes to delete.

There is a ReturnCodes output parameter for the different errors that could generate for each individual volume being deleted.

**Note:** The ReturnElementsToStoragePool method is not in the CIM or SMI-S schema. This method is an IBM extension.

**IBMTSDS_PrimordialStoragePool. GetSupportedSizeRange**

CIM_StoragePool has standard methods GetSupportedSizes and GetSupportedSizeRange to find out what sizes to use to create elements from the StoragePool.

For IBMTSDS_PrimordialStoragePool, only GetSupportedSizeRange is supported. It can be used to retrieve the supported physical size of virtual pool which can be created. In this function, the Goal input parameter is used to represent the data type of the pool to use to calculate the possible valid sizes, and it returns a range and a divisor of valid sizes.

**Note:** If there is no spare space to create a minimum pool, the returned minimum and maximum size of GetSupportedSizeRange for storage pool will be zero.

**IBMTSDS_VirtualPool.GetSupportedSizeRange**

CIM_StoragePool has standard methods GetSupportedSizes and GetSupportedSizeRange to find out what sizes to use to create elements from the StoragePool.

This method can be used to retrieve the supported physical size of virtual pool which can be created. In this function, the Goal input parameter is used to represent the data type of the pool to use to calculate the possible valid sizes, and it returns a range and a divisor of valid sizes.

**Note:** If there is no spare space to create a minimum pool, the returned minimum and maximum size of GetSupportedSizeRange for storage pool will be zero.

**Creating a ConcreteStoragePool**

You can use the CreateOrModifyStoragePool method to create a ConcreteStoragePool.

**About this task**

This procedure provides an example recipe for configuring an XIV device. There are other ways you can use the XIV Open API to configure XIV devices.

**Procedure**

1. Call IBMTSDS_PrimordialStoragePool.GetSupportedSizeRange with ElementType set to Storage Pool. If the return code is 0 (success), present these values to the user and allow the user to pick a size.
2. invokeMethod on IBMTSDS_StorageConfigurationService.CreateOrModifyStoragePool and specify the ElementName, Size, VirtualSize and SnapshotSize input parameters.
3. Save the pool output parameter. That is the reference to the VirtualPool that was just created.
Creating a StorageVolume
You can use the CreateOrModifyElementFromStoragePool method to create a StorageVolume.

About this task
This procedure provides an example recipe for configuring an XIV device. There are other ways you can use the XIV Open API to configure XIV devices.

Procedure
1. associatorNames from the Concrete StoragePool that was just created to CIM_StorageCapabilities to get matching instance of IBM_TSDS_VirtualPoolCapabilities.
2. associatorNames from IBM_TSDS_VirtualPoolCapabilities to CIM_StorageSetting to get valid instances of IBM_TSDS_DataTypeSetting and allow user to pick the Data Type to use to create the volume.
3. Call IBM_TSDS_VirtualPool.GetSupportedSizeRange on the virtual pool created in Step 1 using the data type that the user picked as the Goal input parameter. If the return code is 0 (success), present these values to the user and allow the user to pick a size.
4. InvokeMethod on IBM_TSDS_StorageConfigurationService.CreateOrModifyElementFromStoragePool with the IBM_TSDS_VirtualPool created in Step 1, the IBM_TSDS_DataTypeSetting, Size and ElementName that the user specified as the InPools, Size, Goal, and ElementName input parameters.
5. Save the TheElement output parameter and display it to the user, letting the user know that this is the instance of the volume that was created.

iSCSI Target Ports profile
The iSCSI Target Ports subprofile provides a standard model for representing iSCSI elements as CIM objects.

Through this model, a client can obtain information about the iSCSI capabilities and settings of a XIV storage array, as well as the iSCSI Network Entities, Nodes, Network Portals, and SCSI Ports that are provided by iSCSI-capable arrays and the storage volumes exposed by those arrays on an iSCSI network.

iSCSI sessions and settings are not currently configurable through the XIV SMI-S Provider.
iSCSI Target Ports Object Model

The CIM agent is designed to allow for the retrieval of information for each layer of abstraction.

The following table provides a map of terminology from iSCSI standards and CIM class names on the above UML diagram. iSCSI Session, Connection and Portal Group are not supported by XIV SMI-S.

Figure 6. iSCSI Target Ports SMI-S model for XIV systems
### Table 23. iSCSI terminology and CIM class names

<table>
<thead>
<tr>
<th>iSCSI term</th>
<th>CIM class name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network entity</td>
<td>ComputerSystem</td>
<td>The Network Entity represents a device that is accessible from the IP network. A Network Entity shall have one or more Network Portals, each of which can be used to gain access to the IP network by some iSCSI Nodes contained in that Network Entity.</td>
</tr>
<tr>
<td>Node</td>
<td>SCSIProtocolController</td>
<td>The iSCSI Node represents a single iSCSI Target. There are one or more iSCSI Nodes within a Network Entity. The iSCSI Node is accessible via one or more Network Portals. An iSCSI Node is identified by its iSCSI Name. The separation of the iSCSI Name from the addresses used by and for the iSCSI Node allows multiple iSCSI nodes to use the same address, and the same iSCSI node to use multiple addresses.</td>
</tr>
<tr>
<td>SCSI port</td>
<td>iSCSIProtocolEndpoint</td>
<td>A SCSI Port using an iSCSI service delivery subsystem. A collection of Network Portals that together act as a SCSI Target or target.</td>
</tr>
<tr>
<td>Network Portal</td>
<td>TCPProtocolEndpoint, IPProtocolEndpoint, EthernetPort</td>
<td>The Network Portal is a component of a Network Entity that has a TCP/IP network address and that may be used by an iSCSI Node within that Network Entity for the connection(s) within one of its iSCSI sessions. A Network Portal in a Target is identified by its IP address.</td>
</tr>
</tbody>
</table>

**Network entities**

Each IBM_TSR_SStorageSystem is considered a separate Network Entity, which instance associated to an instance of CIM_iSCSICapabilities via the CIM_ElementCapabilities association is an iSCSI network entity visible on an iSCSI network. Instances of CIM_iSCSICapabilities contain the iSCSI specification versions and the authentication mechanisms supported by a network entity, which can be used to determine the capabilities of a storage array.

**iSCSI nodes**

The iSCSI Node is represented by instances of CIM_SCSIProtocolController. This class can have many subclasses. In a generic way the Name and NameFormat properties can be used to determine whether any given CIM_SCSIProtocolController is an iSCSI Node. All CIM_SCSIProtocolController instances representing iSCSI Nodes are required to have a Name property whose value is an iSCSI Name. Therefore the NameFormat property must have a value of 3 (which maps to iSCSI Name). On the XIV arrays, the iSCSI Node is represented by instances of IBM_TSR_iSCSIProtocolController, a subclass of CIM_SCSIProtocolController, which is associated to the top-level computer system (IBM_TSR_SStorageSystem) through the CIM_SystemDevice association. For XIV arrays, each IBM_TSR_SStorageSystem has one Node for each IP Interface. The iSCSI Names for those nodes follow the IQN (iSCSI qualified name) format. The NameFormat and Name properties of the IBM_TSR_iSCSIProtocolController instances are shown bolded in the example below. instance of IBM_TSR_iSCSIProtocolController {
SystemCreationClassName = "IBMTSR_SStorageSystem"; SystemName = "IBM.2810-6000095";
CreationClassName = "IBMTSR_iSCSIProtocolController"; Name = "iqn.2005-
iSCSI ports

Each iSCSI Node on a XIV array is accessed through a single CIM_iSCSIProtocolEndpoint instance. The CIM_iSCSIProtocolEndpoint instances can be discovered by finding all of the iSCSI nodes, as described in "iSCSI nodes" on previous page, then following the CIM_SAPAvailableForElement association from the iSCSI Node (IBMTSDS_iSCSIProtocolController) to the CIM_iSCSIProtocolEndpoint instance. To find all of the CIM_iSCSIProtocolEndpoint instances for a network entity, follow the CIM_HostedAccessPoint association from the IBMTSDS_StorageSystem to the CIM_iSCSIProtocolEndpoint instances.

Network Portal

There is a single network portal per IP Interface on a XIV array, just as there is one iSCSI Node and iSCSI Port per IP Interface. Network portals are represented using the CIM_TCPProtocolEndpoint, CIM_IPProtocolEndpoint and CIM_EthernetPort classes, associated through the CIM_BindsTo association:

- CIM_IPProtocolEndpoint contains the IP address of the Network Portal in the IPv4Address property. Note that the IPv6Address property is not currently supported.
- CIM_TCPProtocolEndpoint instance contains the TCP Port of the Network Portal in the PortNumber property.
- CIM_EthernetPort instance contains the physical properties of the Network Portal in the PermanentAddress, DeviceID, FullDuplex, Speed and OtherIdentifyingInfo properties.

IBMTSDS_iSCSIConfigurationService. CreateIPProtocolEndpoint

The IBMTSDS_iSCSIConfigurationService.CreateIPProtocolEndpoint method creates a new IP endpoint.

**ElementName (required)**
A user-friendly name for the IP endpoint. It cannot be empty or be left as null.

**ModuleNumber (required)**
An identifier indicating the module that will contain the IP endpoint. It cannot be 0 or be left as null.

**PortNumber (required)**
Port identifier for this IP endpoint on this module. The value cannot be a value other than 1, 2, 3, or 4 because each module only has four ports.

**IPv4Address (required)**
IPv4 address for this port.

**IPv4Gateway (required)**
IPv4 gateway for this port.

**IPv4SubnetMask (required)**
IPv4 subnet mask for this port.

**MTU**
MTU for this port. If it is left as null, default value 4500 will be used.

**IPEndpoint**
The reference to a point that was created.

IBMTSDS_iSCSIConfigurationService. ModifyIPProtocolEndpoint

The IBMTSDS_iSCSIConfigurationService.ModifyIPProtocolEndpoint method modifies an IP endpoint.

The method ModifyIPProtocolEndpoint is not in the CIM/SMI-S schema but an IBM extension.

**ElementName**
A new user-friendly name for the IP endpoint.
**ModuleName**
An identifier indicating the new module that will contain the IP endpoint.

**PortNumber**
The new port identifier for this IP endpoint. If it is set, the value cannot be a value other than 1, 2, 3, or 4 because each module only has four ports.

**IPv4Address**
A new IPv4 address for this port.

**IPv4Gateway**
A new IPv4 gateway for this port.

**IPv4SubnetMask**
A new IPv4 subnet mask for this port.

**MTU**
A new MTU for this port.

**IPEndpoint**
The reference to a point to modify.

**IBMDS_iSCSIConfigurationService. DeleteIPProtocolEndpoint**
This method is used to delete an IP PortocolEndPoint. The only input parameter required is a reference to the PortocolEndPoint that you want to delete. Only one PortocolEndPoint can be deleted at a time.

The method DeleteIPProtocolEndpoint is not in the CIM/SMI-S schema but an IBM extension.

**Creating an IP endpoint**
You can use CreateIPProtocolEndpoint method to create an IP endpoint

**About this task**
This procedure provides an example recipe for configuring an XIV device. There are other ways you can use the XIV Open API to configure XIV devices.

**Procedure**
1. Obtain the reference (CIMObjectPath) of an IBMDS_iSCSIConfigurationService instance that is associated with the IBMDS_StorageSystem in which you will get the performance statistics by traversing the IBMDS_HostedSCSIConfigurationService association.
2. Invoke the IBMDS_iSCSIConfigurationService.CreateIPProtocolEndpoint method to create an IP endpoint with specified values of parameters ElementName, ModuleNumber, PortNumber, IPv4Address, IPv4Gateway, IPv4SubnetMask, MTU.
3. After it is successfully invoked, check the out value of IPEndpoint representing the IP endpoint that was created.

**Modifying an IP endpoint**
You can use ModifyIPProtocolEndpoint method to create an IP endpoint

**About this task**
This procedure provides an example recipe for configuring an XIV device. There are other ways you can use the XIV Open API to configure XIV devices.
**Procedure**

1. Obtain the reference (CIMObjectPath) of an IBMTSDS_iSCSIConfigurationService instance that is associated with the IBMTSDS_StorageSystem in which you will get the performance statistics by traversing the IBMTSDS_HostedI SCSIConfigurationService association.

2. Obtain the reference (CIMObjectPath) of an IBMTSDS_IPEndPoint instance which is to be modified.

3. Invoke the IBMTSDS_iSCSIConfigurationService.ModifyIPProtocolEndpoint method to modify an IP endpoint with values of parameters ElementName, IPv4Address, IPv4Gateway, IPv4SubnetMask, MTU, IPEndPoint.

4. After it is successfully invoked, the IP endpoint will be modified.

**Deleting an IP endpoint**

You can use the DeleteIPProtocolEndpoint method to delete an IP endpoint.

**About this task**

This procedure provides an example recipe for configuring an XIV device. There are other ways you can use the XIV Open API to configure XIV devices.

**Procedure**

1. Obtain the reference (CIMObjectPath) of an IBMTSDS_iSCSIConfigurationService instance that is associated with the IBMTSDS_StorageSystem in which you will get the performance statistics by traversing the IBMTSDS_HostedI SCSIConfigurationService association.

2. Obtain the reference (CIMObjectPath) of an IBMTSDS_IPEndPoint instance which is to be deleted.

3. Invoke the IBMTSDS_iSCSIConfigurationService.DeleteIPProtocolEndpoint method and set IPEndPoint as the reference obtained in step 2.

4. After it is successfully invoked, the IP endpoint is deleted.

**Masking and Mapping profile**

You can control which client hosts can see and access storage volumes by mapping and masking client hosts. The SNIA SMI-S specifications define the Mapping and Masking profile to support this functionality.

**LUN masking**

The process of configuring software in SAN nodes to determine which hosts have access to exported drives (volumes).

LUN masking for the XIV system is storage-based port mapping.

**LUN mapping**

The process of creating a disk resource and defining its external access paths using a LUN. The LUN is then mapped to an external ID descriptor. For example, a SCSI port or a target ID. To ensure uninterrupted data availability, map a logical volume to allow access from multiple ports, target IDs or both.

The XIV system architecture for masking and mapping involves the following objects on two kinds of devices:

- Objects on the XIV system
  - Storage device I/O ports (FC ports and iSCSI ports)
  - Storage volumes
- Objects on the host system
  - Host ports (HBAs)
  - Hosts (host systems)
Clusters

Figure 7 provides an example of how the XIV system uses masking and mapping.

The following list defines each of the objects shown in Figure 7.

**Storage device I/O ports**

Ports on the XIV system that represent a fiber channel port or an iSCSI port that is used for host I/O operations. Storage device I/O ports are sometimes referred as target ports.

**Volumes**

Objects on the XIV system that are mapped to host or a cluster. One volume cannot be mapped to multiple hosts that are not part of the same cluster.

**Host ports**

Objects that represent a single fiber channel port or an iSCSI port on the host system. The architecture does not track host systems. If a host system has multiple host ports, a host port object must be created for each port.

**Hosts**

Systems with a set of initiator ports that are used to access storage on the device. For each host system, you can create a host object, add host port objects, and then map the host to volumes through the IBM XIV Storage Management GUI, XCLI, or CIM agent.

**Clusters**

Groups of hosts. For the XIV system, multiple hosts can see the same set of volumes. For a host system cluster, you can create a cluster object, add host objects to it, and then map the cluster object to multiple volumes through the IBM XIV Storage Management GUI, XCLI, or CIM agent. Clusters are required when more than one host can access the same volume.

After you have set up the physical network and created a host object, add Fibre Channel ports or iSCSI ports to the host. Map the volumes to the host so the volume can be used for I/O operations.

**Mapping hosts to a volume**

You can map a host to volumes. A host can contain several initiator ports including Fiber Channel ports and iSCSI ports. In this case, create a host, add the initiator ports, and then map the host to a volume.
Through this mapping, the host system is able to access the mapped volume through the included initiator ports.

**Mapping clusters to a volume**

You can map a cluster to a volume when several hosts need to be mapped to the same volume. In this case, create a cluster and add hosts, and then map the cluster to volumes. Through this mapping, the volume will be mapped to all hosts that are contained in the cluster. If a host is included in a cluster, it cannot be mapped to another volume separately.

See the *IBM XIV Storage System: Architecture, Implementation, and Usage* and the *IBM XIV Storage System Product Overview* for more information about the architecture of the XIV system.

**Masking and Mapping object model**

The Masking and Mapping profile provides an interface to specify which hosts can see which volumes and through which target ports.

For XIV host and cluster mappings, specifying target ports for the “view” is not supported, which means that target ports cannot be selected when configuring host mappings, but can be configured by zoning configurations on the switch. The target ports involved in all “view” can be displayed by enumerating the instances of CIM_ProtocolControllerForPort class.

### Table 24. Masking and mapping classes

<table>
<thead>
<tr>
<th>CIM class name</th>
<th>Description</th>
<th>XIV CIM class name</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIM_FCPort</td>
<td>A Fibre Channel port on the XIV system.</td>
<td>IBMTSDS_FCPort</td>
</tr>
<tr>
<td>CIM_EthernetPort</td>
<td>An Ethernet port on the XIV system.</td>
<td>IBMTSDS_EthernetPort</td>
</tr>
<tr>
<td>CIM_SCSIProtocolEndPoint</td>
<td>The fibre channel or iSCSI port that is used in the mapping</td>
<td>IBMTSDS_SCSIProtocolEndPoint</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IBMTSDS_iSCSIProtocolEndpoint</td>
</tr>
<tr>
<td>CIM_SCSIProtocolController</td>
<td>A logical entity that represents a host and the relationship to the volumes</td>
<td>IBMTSDS_SCSIProtocolController</td>
</tr>
</tbody>
</table>

*Figure 8. Masking and mapping object model in SMI-S*

Table 24 describes the masking and mapping classes that are available. See the MaskingMapping.mof file for more information about the masking and mapping object model. The MOF documentation is in the mof folder in the CIM agent installation directory.
Table 24. Masking and mapping classes (continued)

<table>
<thead>
<tr>
<th>CIM class name</th>
<th>Description</th>
<th>XIV CIM class name</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIM_StorageVolume</td>
<td>A storage volume on the XIV system</td>
<td>IBMTSDS_SEVolume</td>
</tr>
<tr>
<td>CIM_AuthorizedPrivilege</td>
<td>A logical entity that represents the access permissions for a set of volumes</td>
<td>IBMTSDS_Privilege</td>
</tr>
<tr>
<td>CIM_StorageHardwareID</td>
<td>An initiator port on the host system</td>
<td>IBMTSDS_StorageHardwareID</td>
</tr>
<tr>
<td>CIM_SystemSpecificCollection</td>
<td>A logical entity which represents a collection of StorageHardwareID (host)</td>
<td>IBMTSDS_SystemSpecificCollection</td>
</tr>
<tr>
<td>CIM_Cluster</td>
<td>A logical entity which represents a collection of SystemSpecificCollection (hosts)</td>
<td>IBMTSDS_Cluster</td>
</tr>
</tbody>
</table>

See the MaskingMapping.mof file for more information about the Masking and mapping object model. The MOF documentation is in the mof folder in the CIM agent installation directory.

XIV CIM Agent releases before 10.2.4 only provide a read-only implementation of the masking and mapping subprofile. This means that this model supports read-only information about the maskings and mappings, but does not offer manipulations such as creating, deleting, and modifying.

XIV CIM Agent release 10.2.4 provides a full implementation of masking and mapping subprofile including the following items:
- iSCSI port support
- StorageHardwareID creation and deletion
- HardwareIDCollection creation and modification
- Expose paths and hide paths
- SMI-S 1.4

**iSCSI Port support**

By default, the XIV Storage System ships equipped with six iSCSI ports. Three of the interface modules support iSCSI, with two ports in each module. You can refer to XIV Redbook for detailed information. XIV CIM Agent 10.2.4 adds the full iSCSI support by implementation of the iSCSI target ports profile and the masking and mapping profile.

**StorageHardwareID manipulation**

A StorageHardwareID represents the host-side initiator that logs into a storage area network. The XIV SMI-S provider supports host-side initiators that contain fibre channel or iSCSI host ports. A StorageHardwareIDManagementService allows a CIM client to locate StorageHardwareIDs known to a storage array, as well as create, modify, and delete StorageHardwareIDs.
- CreateStorageHardwareID() method creates a new StorageHardwareID. On XIV, a host will be created which contains the specified initiator port.
- DeleteStorageHardwareID() deletes a StorageHardwareID. On XIV, the initiator port will be removed from the host containing it.
HardwareIDCollection manipulation

A HardwareIDCollection represents a collection of StorageHardwareID (host-side initiators) that logs into a storage area network. A StorageHardwareIDMangementService allows a CIM client to locate HardwareIDCollections known to a storage array, as well as create HardwareIDCollection and remove HardwareIDs from HardwareIDCollection.

- CreateHardwareIDCollection method creates a new HardwareIDCollection with specified ElementName and Setting.
- AddHardwareIDsToCollection adds StorageHardwareIDs into a HardwareIDCollection.

Note: The methods ModifyHardwareIDCollection, DeleteHardwareIDCollection, and RemoveHardwareIDsFromCollection are not in the CIM or SMI-S schema. These methods are an IBM extension.

Expose paths and hide paths

A SCSIProtocolController represents a view of volumes that can be assigned access to a StorageHardwareID. You can use a ControllerConfigurationService to locate SCSIProtocolControllers as well as ExposePaths and HidePaths from StorageHardwareIDs to SCSIProtocolControllers.

A privilege determines the access rights between the StorageHardwareID and the SCSIProtocolController. The XIV SMI-S Provider only supports read/write access. A privilege can be located using the PrivilegeManagementService.

IBMTSDS_StorageHardwareIDManagement. CreateStorageHardwareID

You can use the CreateStorageHardwareID method to create a StorageHardwareID. This method will create a new host in the XIV device containing the initiator port specified by StorageID.

**Setting**
- An input parameter that represents the operating system of the port. This must be an instance of IBMTSDS_StorageClientSettingData. Each instance represents a different operating system. Only Standard, HPUX, and zVM are supported by XIV CIM Agent. If it is left null, Standard will be used by default.

**IDType**
- An input parameter that represents the type of port. To create a HardwareID representing a Fiber Channel port, specify 2; to create a HardwareID representing an iSCSI port, specify 5.

**StorageID**
- An input parameter that represents the ID of the port. If IDType is 2, specify it as the port’s WWN; if IDType is 5, specify it as the port’s IQN.

**HardwareID**
- An output parameter; the reference to the created StorageHardwareID.

IBMTSDS_StorageHardwareIDManagement. DeleteStorageHardwareID

Use the DeleteStorageHardwareID method to delete StorageHardwareIDs.

**HardwareID**
- An input parameter representing a reference to the StorageHardwareID to delete.

IBMTSDS_StorageHardwareIDManagement. CreateHardwareIDCollection

You can use the CreateHardwareIDCollection method to create HardwareIDCollections.

**ElementName (required)**
- An input parameter representing a user-friendly name for the HardwareIDCollection that is being
created. The ElementName should be unique in the system, meaning it cannot be a name that is already assigned to one of the other HardwareIDCollections in the system. If not specified, CIM will generate a random name.

**HardwareIDs**

An input parameter representing ID of the ports contained by the HardwareIDCollection. If specified, each ID must be the WWN of a Fiber Channel port or the IQN of an iSCSI port. If not specified, an empty HardwareIDCollection will be created.

**HardwareIDCollection**

An output parameter representing the reference to the created HardwareIDCollection.

**IBMTSDS_StorageHardwareIDManagement. AddHardwareIDsToCollection**

You can use the AddHardwareIDsToCollection method to add StorageHardwareIDs (representing initiator ports) into SystemSpecificCollection (representing a host). The operation will fail if the port already belongs to another host. The operation will succeed if the port already belongs to the specified host, but it will have no effect.

**HardwareIDs (required)**

An input parameter representing an array of strings containing StorageIDs of StorageHardwareID instances that will become members of the HardwareIDCollection.

**HardwareIDCollection (required)**

An input parameter representing the reference to the SystemSpecificCollection into which the HardwareIDs will be added.

**IBMTSDS_ControllerConfigurationService. ExposePaths**

Use the ExposePaths method to assign volumes to a StorageHardwareID representing a host or cluster in XIV.

ExposePaths performs the mapping and masking operation in one method call. It exposes a list of volumes to a list of initiators, through one or more SCSIProtocolControllers (SPCs). It supports creating or modifying SPCs depending on different specified parameters.

**Table 25. ExposePaths parameters and use cases**

<table>
<thead>
<tr>
<th>Parameters/Use cases</th>
<th>LUNames</th>
<th>InitiatorPortIDs</th>
<th>TargetPortIDs</th>
<th>DeviceNumbers</th>
<th>DeviceAccesses</th>
<th>ProtocolControllers (on input)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create a new view</td>
<td>Optional</td>
<td>Optional</td>
<td>NULL</td>
<td>Optional</td>
<td>Optional</td>
<td>NULL</td>
</tr>
<tr>
<td>Add LUNs to a view</td>
<td>Mandatory</td>
<td>NULL</td>
<td>Optional</td>
<td>Optional</td>
<td></td>
<td>contains a single SPC ref</td>
</tr>
<tr>
<td>Add initiator IDs to a view</td>
<td>NULL</td>
<td>Mandatory</td>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
<td>contains a single SPC ref</td>
</tr>
</tbody>
</table>

**LUNames (required)**

A string array input parameter representing the volumes to map to the SPC. These are not references to volume instances, but strings that match the Name property of the IBMTSDS_SELVolume (CIM_StorageVolume), which is the WWN of an XIV volume.

**InitiatorPortIDs (required if StorageHardwareID has no volumes previously mapped)**

A string array input parameter representing the initiator ports to be added to an SPC. These are not references to StorageHardwareID instances, but the StorageID property of IBMTSDS_StorageHardwareID.

**DeviceNumbers (required)**

A string array input parameter representing the LUN IDs. Each item in this array should be a
number between 0 and 511. The number of items is the same as the number of items specified by the LUNames. If not specified, the CIM Agent will assign unused LUN IDs for the operation.

**DeviceAccesses**
A unit16 array input parameter representing the access rights to give to the StorageHardwareID to the volumes specified in LUNames. Only read/write access is supported. So this must be an array the same size as LUNames where each value is 2.

**ProtocolControllers (required if adding volumes to a StorageHardwareID that was already previously mapped to volumes)**
As an input parameter, this represents the SCSIProtocolController to be modified. As an output parameter, this represents the SCSIProtocolController that is being modified or created.

### IBMDSControllerConfigurationService. HidePaths

The HidePaths method is the inverse of ExposePaths, in other words, removing access to volumes from a StorageHardwareID.

#### Table 26. HidePaths parameters and use cases

<table>
<thead>
<tr>
<th>Parameters/ Use cases</th>
<th>LUNames</th>
<th>InitiatorPortIDs</th>
<th>TargetPortIDs</th>
<th>ProtocolControllers (on input)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remove LUs from a view</td>
<td>Mandatory</td>
<td>NULL</td>
<td>NULL</td>
<td>contains a single SPC ref</td>
</tr>
<tr>
<td>Remove initiator IDs from a view</td>
<td>NULL</td>
<td>Mandatory</td>
<td>NULL</td>
<td>contains a single SPC ref</td>
</tr>
<tr>
<td>Hide full paths from a view</td>
<td>Mandatory</td>
<td>Mandatory</td>
<td>NULL</td>
<td>contains a single SPC ref</td>
</tr>
</tbody>
</table>

**LUNames (required for removing access to volumes)**
A string array input parameter representing the volumes to unmap from the StorageHardwareID. These are not references to volume instances, but are strings that match the name property of the IBMDS_SEVolume (CIM_StorageVolume).

**InitiatorPortIDs (required for removing access to initiator ports)**
A string array input parameter representing the initiator port that you want to remove from the SCSIProtocolController view.

**ProtocolControllers (required)**
As an input parameter, this represents the SCSIProtocolController to be modified. As an output parameter, this represents the SCSIProtocolController that was modified or deleted.

### IBMDSControllerConfigurationService. DeleteProtocolController

The DeleteProtocolController method deletes an existing ProtocolController. In XIV, the corresponding host will be deleted, as well as all the host ports contained by the host.

**ProtocolController (required)**
An input parameter representing the ProtocolController to be deleted.

**DeleteUnits**
An input parameter representing the requirement to delete the volumes mapped to the ProtocolController. If true is specified, the mapped volumes will be deleted.

### Sample recipe
The following task is a sample recipe for configuring an XIV device. This is only a sample; there are many ways to use the XIV Open API to configure an XIV device.

1. Map FC and iSCSI ports to volumes:
   a. Attain WWN of host FC ports, IQN of host iSCSI ports, and volume names.
   b. InvokeMethod on IBMDSControllerConfigurationService.ExposePaths with InitiatorPortIDs set to the IDs of the host ports, LUNames set to the volume names, and DeviceAccesses set to 2.
2. Add volumes to an existing SCSIProtocolController:
   a. Attain the host reference and volume names.
   b. Find the IBMTSDS_SCSIProtocolController instance matching the selected host.
   c. InvokeMethod on IBMTSDS_ControllerConfigurationService.ExposePaths with LUNames set to
      volume names, DeviceAccesses set to 2, and ProtocolControllers set to the SCSIProtocolController.

3. Add initiate ports to an existing SPC:
   a. Attain IDs of initiate ports.
   b. Find the IBMTSDS_SCSIProtocolController instance.
   c. InvokeMethod on IBMTSDS_ControllerConfigurationService.ExposePaths with InitiatorPortIDs set
      to the initiate port IDs, DeviceAccesses set to 2, and ProtocolControllers set to the
      SCSIProtocolController.

4. Remove volumes from an SPC:
   a. Attain the SPC reference and volume names.
   b. Find the IBMTSDS_SCSIProtocolController instance matching the selected host.
   c. InvokeMethod on IBMTSDS_ControllerConfigurationService.HidePaths with LUNames set to
      volume names, and ProtocolControllers set to the SCSIProtocolController.

5. Remove initiate ports from an SPC:
   a. Attain the SPC reference and IDs of the initiate port.
   b. Find the IBMTSDS_SCSIProtocolController instance.
   c. InvokeMethod on IBMTSDS_ControllerConfigurationService.HidePaths with InitiatorPortIDs set to
      the IDs of the initiate ports, and ProtocolControllers set to the SCSIProtocolController.

---

### Indication profile

SMI-S supports two types of indications: Life Cycle Indications and Alert Indications. Life Cycle Indications are used to convey changes in the model and are concerned only with the creation, modification, or deletion of CIM Instances. AlertIndications are used to draw attention of subscribing client applications to the occurrence of an event. Life Cycle Indications have been implemented in all XIV CIM Agent releases. Beginning with XIV CIM Agent release 11.0, AlertIndications have been implemented.

#### Supported lifecycle indications by class

Table 27 identifies the class of objects that can be monitored using the specified indication type for XIV arrays.

<table>
<thead>
<tr>
<th>Class name</th>
<th>Supported lifecycle indications</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBMTSDS_SEVolume</td>
<td>IBMTSDS_InstCreation</td>
</tr>
<tr>
<td>IBMTSDS_VirtualPool</td>
<td>IBMTSDS_InstDeletion</td>
</tr>
<tr>
<td>IBMTSDS_SCSIProtocolController</td>
<td>IBMTSDS_InstModification</td>
</tr>
<tr>
<td>IBMTSDS_SystemSpecificCollection</td>
<td>IBMTSDS_InstCreation</td>
</tr>
<tr>
<td>IBMTSDS_StorageHardwareID</td>
<td>IBMTSDS_InstDeletion</td>
</tr>
<tr>
<td>IBMTSDS_iSCSIProtocolEndpoint</td>
<td>IBMTSDS_InstCreation</td>
</tr>
</tbody>
</table>
The lifecycle indications are sent after an XIV CIM Agent configuration function is successfully performed. For example, if SMI-S client invoked the XIV CIM Agent ExposePaths method to map a volume to a host, a related InstCreation indication with the newly created IBMTSDS_ProtocolControllerForSEUnit instance is sent from the provider.

In addition, XIV CIM Agent periodically fetches XIV events and sends related lifecycle indications. For example, if a StorageVolume is created on an XIV array through the XIV GUI, an SMI-S client receives a related InstCreation indication with the newly created IBMTSDS_SEVolume instance from the provider. By default, the XIV events fetching interval is 30 seconds.

**Alert indications**

The XIV CIM Agent periodically fetches XIV non-internal events and sends one alert indication for each event. The alert indication comes in the form of class IBMTSDS_AlertIndication. By default, the XIV events fetching interval is 30 seconds.

**Note:** The XIV CIM Agent has a limitation of values of properties:

- IndicationTime of IBMTSDS.InstCreation class
- IndicationTime of IBMTSDS.InstDeletion class
- IndicationTime of IBMTSDS.InstModification class
- EventTime of IBMTSDS.AlertIndication class
- IndicationTime of IBMTSDS.AlertIndication class

The above properties are representing a timestamp in format yyyyymmddhhmss.mmmmmmsutc, where:

- yyyy is the 4-digit year
- mm is the month within the year (starting with 01)
- dd is the day within the month (starting with 01)
- hh is the hour within the day (24-hour clock, starting with 00)
- mm is the minute within the hour (starting with 00)
- ss is the second within the minute (starting with 00)
- mmmmmm is the microsecond within the second (starting with 000000)
- s is a "+" or "-", indicating that the value is a timestamp, and indicating the sign of the UTC (Universal Coordinated Time) correction field. A "+" is used for time zones east of Greenwich, and a "-" is used for time zones west of Greenwich.
- utc is the offset from UTC in minutes (using the sign indicated by s)

Fields other than utc are correct which are representing the UTC timestamp and the utc field as 000. Occasionally the utc property will have values other than 000 (such as 540). Values other than 000 should be ignored.

**Replication Services profile**

The Replication Services profile allows a storage system to copy data from a source element to a target element. The copy operations may be performed on elements from the same storage system or across a connection to a different storage system. Elements used in a copy operation may be grouped to facilitate the copy operation on many elements at the same time. Furthermore, the elements of a group may be declared as Consistent.

Two types of synchronization views are supported. A target element can be synchronized to the current view of the source element, or it can be synchronized to a point-in-time view. Snapshots and clones represent a point-in-time view, while a mirror represents a current view. Two copy modes are supported: synchronous and asynchronous. In synchronous mode, the writer waits for acknowledgement that a write to the source element was processed by the target element before accepting another I/O from the host. In
asynchronous mode, the writer does not wait for this acknowledgement and continues processing I/Os, enabling writes to be sent to the target element at a later time.

**Replication Services object model**

Data can be copied from a source element or group of source elements from the same storage system or across a connection to a different storage system target element.

![Replication Services SMI-S model for XIV systems](image)

*Figure 9. Replication Services SMI-S model for XIV systems*
Replication Service capabilities

Use XIV terminology to map to SMI-specific terminology:

Table 28. Mapping XIV terminology to SMI terminology

<table>
<thead>
<tr>
<th>XIV Terminology</th>
<th>SMI Terminology</th>
<th>Description</th>
<th>XIV CIM Agent support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Snapshot</td>
<td>Asynchronous Snapshot Local</td>
<td>A &quot;Point-in-Time&quot;, associated virtual copy of the source element.</td>
<td>version 11.1 or later</td>
</tr>
<tr>
<td>Volume copy</td>
<td>Synchronous Clone Local</td>
<td>A full, &quot;Point-In-Time&quot;, unassociated local copy of the source element.</td>
<td>version 11.1 or later</td>
</tr>
<tr>
<td>Synchronous Remote Mirroring</td>
<td>Synchronous Mirror Remote</td>
<td>A synchronized remote copy of the source element.</td>
<td>not supported</td>
</tr>
<tr>
<td>Asynchronous Remote Mirroring</td>
<td>Asynchronous Mirror Remote</td>
<td>An asynchronous remote copy of the source element.</td>
<td>not supported</td>
</tr>
</tbody>
</table>

The single instance of the class ReplicationServiceCapabilities and its methods describe the various capabilities of the service. You can examine the ReplicationServiceCapabilities instance and invoke its methods to determine the specific capabilities of a replication service implementation.

Table 29. Replication Service methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConvertSyncTypeToReplicationType</td>
<td>Translates CopyType, Mode, and Local/Remote to the corresponding ReplicationType</td>
</tr>
<tr>
<td>ConvertReplicationTypeToSyncType</td>
<td>Translates ReplicationType to the corresponding CopyType, Mode, and Local/Remote</td>
</tr>
<tr>
<td>GetSupportedFeatures</td>
<td>Determines supported features</td>
</tr>
<tr>
<td>GetSupportedGroupFeatures</td>
<td>Determines supported group features</td>
</tr>
<tr>
<td>GetSupportedCopyStates</td>
<td>Determines the supported CopyStates</td>
</tr>
<tr>
<td>GetSupportedGroupCopyStates</td>
<td>Determines supported group copy states</td>
</tr>
<tr>
<td>GetSupportedWaitForCopyStates</td>
<td>Determines supported waitforcopystates</td>
</tr>
<tr>
<td>GetSupportedConsistency</td>
<td>Determines supported consistency</td>
</tr>
<tr>
<td>GetSupportedOperations</td>
<td>Determines supported operations</td>
</tr>
<tr>
<td>GetSupportedGroupOperations</td>
<td>Determines supported group operations</td>
</tr>
<tr>
<td>GetSupportedListOperations</td>
<td>Determines supported list operations</td>
</tr>
<tr>
<td>GetSupportedMaximum</td>
<td>Determines supported maximum</td>
</tr>
<tr>
<td>GetDefaultConsistency</td>
<td>Determines default consistency</td>
</tr>
<tr>
<td>GetDefaultGroupPersistency</td>
<td>Determines default group persistency</td>
</tr>
</tbody>
</table>

Group management

The CIM agent supports group manipulation, including consistency group and snapshot group.

Table 30. Group management classes

<table>
<thead>
<tr>
<th>XIV</th>
<th>SMI class</th>
<th>XIV CIM class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consistency group</td>
<td>CIM_ReplicationGroup</td>
<td>IBMTSDS.ConsistencyGroup</td>
</tr>
</tbody>
</table>
Table 30. Group management classes (continued)

<table>
<thead>
<tr>
<th>XIV</th>
<th>SMI class</th>
<th>XIV CIM class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Snapshot group</td>
<td>CIM_ReplicationGroup</td>
<td>IBMTSDS_SnapshotGroup</td>
</tr>
<tr>
<td>Association between Consistency group and volumes in it</td>
<td>CIM_OrderedMemberOfCollection</td>
<td>IBMTSDS_ConsistencyGroupToOrderedMembers</td>
</tr>
<tr>
<td>Association between Snapshot group and snapshots in it</td>
<td>CIM_OrderedMemberOfCollection</td>
<td>IBMTSDS_SnapshotGroupToOrderedMembers</td>
</tr>
</tbody>
</table>

Table 31. Extrinsic methods for group management

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreateGroup</td>
<td>Creates a new consistency group</td>
</tr>
<tr>
<td>DeleteGroup</td>
<td>Deletes an existing consistency group</td>
</tr>
<tr>
<td>AddMembers</td>
<td>Adds volumes to a consistency group</td>
</tr>
<tr>
<td>RemoveMembers</td>
<td>Removes volumes from a consistency group</td>
</tr>
</tbody>
</table>

**IBMTSDS_ReplicationService. CreateGroup**

The CreateGroup method allows you to create a consistency group.

**GroupName**
As an input, refers to the name of the group being created. If not specified, a random name is generated.

**Members**
As an input, refers the members to be added to the group being created. If not specified, an empty group will be created.

**ReplicationGroup**
As an output, refers to the group that is created.

**IBMTSDS_ReplicationService. DeleteGroup**

The DeleteGroup method deletes an existing consistency group.

**RemoveElements**
As an input, a value of true indicates that members in this group will be removed before deleting the group. If one or more elements in the group are in a replication relationship, its value is ignored.

**ReplicationGroup**
As an input, refers to the consistency group that you want to delete.
**IBMTSDS_ReplicationService. AddMembers**

The AddMembers method allows you to add volumes to a consistency group.

**ReplicationGroup**
- As an input, refers to the consistency group to which members will be added.

**Members**
- As an input, refers to volumes to be added to the group. All specified volumes must be in the same pool of the consistency group.

**IBMTSDS_ReplicationService. RemoveMembers**

The RemoveMembers method removes volumes from a consistency group.

**ReplicationGroup**
- As an input, refers to the consistency group from which members will be removed.

**Members**
- As an input, refers to volumes to be removed from the group.

**DeleteOnEmptyElement**
- As an input, a value of true indicates that the consistency group will be deleted if all members are removed.

**Replication management**

The CIM agent version 11.1 supports local replication manipulation, including local snapshot and clone.

**Table 32. Replication management classes**

<table>
<thead>
<tr>
<th>XIV</th>
<th>SMI class</th>
<th>XIV CIM class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Volume, Snapshot</td>
<td>CIM_StorageVolume</td>
<td>IBMTSDS_SEVolume</td>
</tr>
<tr>
<td>Association between volume and its snapshot</td>
<td>CIM_StorageSynchronized</td>
<td>IBMTSDS_StorageSynchronized</td>
</tr>
<tr>
<td>Association between consistency group and its snapshot group</td>
<td>CIM_GroupSynchronized</td>
<td>IBMTSDS_GroupSynchronized</td>
</tr>
</tbody>
</table>

**Table 33. Extrinsic methods for replication management**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreateElementReplica</td>
<td>Create a snapshot of a volume, or a duplication snapshot of a snapshot, or a clone of a volume.</td>
</tr>
<tr>
<td>CreateGroupReplica</td>
<td>Create a snapshot group of a consistency group.</td>
</tr>
<tr>
<td>ModifyReplicaSynchronization</td>
<td>Restore a snapshot or a snapshot group; or delete a snapshot or a snapshot group.</td>
</tr>
<tr>
<td>ModifyListSynchronization</td>
<td>Restore a list of snapshots or snapshot groups; or delete snapshots or snapshot groups.</td>
</tr>
<tr>
<td>GetAvailableTargetElements</td>
<td>Return snapshots of a specified volume or snapshot.</td>
</tr>
<tr>
<td>GetReplicationRelationships</td>
<td>Get all of available synchronization relationships.</td>
</tr>
</tbody>
</table>

**IBMTSDS_ReplicationService. CreateElementReplica**

By specifying different parameters, you can use the CreateElementReplica method to create a snapshot of a volume, create a duplicate snapshot of a snapshot, create a clone of a volume, and create a clone of a snapshot.
**ElementName**
As an input, refers the name for the volume or clone being created. If not specified, a random name is generated.

**SyncType**
As an input, refers the type of replica to be created. To create a snapshot, specify 7; to create a clone, specify 8.

**Mode**
As an input, refers the mode of replica to be created. Only 3 is supported, which means an asynchronous replica will be created if specified.

**SourceElement**
As an input, refers the source volume or snapshot for which the replica will be created.

**TargetElement**
As an input, refers to a target element to use if specified. This parameter cannot be specified when creating a snapshot. As an output, refers to the replica that is created.

**TargetPool**
As an input, refers the pool in which the replica is being created. For snapshot creation, if specified, it must be an instance of IBMTSDS_SnapshotPool which is located in the same pool of the source element; for clone creation, if specified, it can be an instance of IBMTSDS_VirtualPool representing any pool in the device, and if not specified, CIM will select the first available pool on the device to create the clone.

**WaitForCopyState**
As an input, refers to the copy state the replica should reach before the method returns. Only 4 is supported if specified.

**IBMTSDS_ReplicationService. CreateGroupReplica**
The CreateElementReplica method creates a snapshot group for a consistency group.

**RelationshipName**
As an input, refers to the name of the snapshot group being created. If not specified, a random name is generated.

**SyncType**
As an input, refers the type of replica to be created. Only 7 is supported if specified because this method can only create a snapshot of a consistency group.

**Mode**
As an input, refers the mode of replica to be created. Only 3 is supported if specified, which means an asynchronous snapshot group will be created.

**SourceGroup**
As an input, refers to a consistency group for which the snapshot group is being created.

**TargetGroup**
As an output, refers to the snapshot group being created.

**Consistency**
As an input, refers to the group consistency. Only 3 is supported if specified.

**Synchronization**
As an output, refers to the created association between the source consistency group and the snapshot group that is created.

**TargetPool**
As an input, refers the pool in which the replica is being created. The pool should be the same as the pool of the SourceGroup if specified.
**WaitForCopyState**
As an input, refers to the copy state the replica should reach before the method returns. Only 4 is supported if specified.

**IBMDS_ReplicationService. ModifyReplicaSynchronization**
By specifying different parameters, you can use the ModifyReplicaSynchronization method to restore a snapshot to a volume, restore a snapshot group to a consistency group, delete a snapshot, and delete a snapshot group.

**Operation**
As an input, refers to the type of modification to be made to the replica. Only 15 and 19 are supported.

**Synchronization**
As an input, refers to the replica synchronization to be modified. It must be an instance of IBMTSDS_GroupSynchronized or IBMTSDS_StorageSynchronized class.

**IBMDS_ReplicationService. ModifyListSynchronization**
This is a batch operation of the IBMDS_ReplicationService:ModifyReplicaSynchronization method, which means that you can modify a list of synchronizations.

**IBMDS_ReplicationService. GetAvailableTargetElements**
The GetAvailableTargetElements method allows you to get all of the candidate target elements for the supplied source element.

**SourceElement**
As an input, refers to the source volume. It must be an instance of IBMDS_SEVolume class.

**SyncType**
As an input, refers to type of target elements. Only 7 is supported if specified.

**Mode**
As an input, refers to the mode of target elements. Only 3 is supported if specified.

**Candidates**
As an output, refers to the target elements of the source volume.

**IBMDS_ReplicationService. GetReplicationRelationships**
The GetReplicationRelationships method allows you to get all of the snapshot relationships on a XIV device.

**Type**
As an input, refers to the replication type. To get snapshots, specify 2; to get snapshot groups, specify 3; if not specified, all snapshots and snapshot groups will be returned.

**SyncType**
As an input, refers to type of target elements. Only 7 is supported if specified.

**Mode**
As an input, refers to mode of target elements. Only 3 is supported if specified.

**Locality**
As an input, refers to locality of target elements. Only 2 is supported if specified.

**CopyState**
As an input, refers to copy state of target elements. Only 4 is supported if specified.

**Synchronizations**
As an input, refers to references of the returned snapshots.
**IBMTSDS_ReplicationService. GetReplicationRelationshipInstances**

This method is similar to GetReplicationRelationships, except that it returns strings of references of snapshots.

*Table 34. SMI Indications*

<table>
<thead>
<tr>
<th>SMI Indication</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBMTSDS_StorageSynchronized InstCreation</td>
<td>After a snapshot is created or snapshot group is created</td>
</tr>
<tr>
<td>IBMTSDS_GroupSynchronized InstCreation</td>
<td>After a snapshot group is created</td>
</tr>
<tr>
<td>IBMTSDS_GroupSynchronized InstDeletion</td>
<td>After a snapshot group is deleted</td>
</tr>
<tr>
<td>IBMTSDS_StorageSynchronized InstDeletion</td>
<td>After a snapshot or snapshot group is deleted</td>
</tr>
</tbody>
</table>
Chapter 5. Conformance tests

All SNIA official Certification Test Programs (CTP) and Microsoft System Center Virtual Machine Manager (SCVMM) Storage Automation tests are complete.

**SNIA official CTP tests**

*Table 35. SNIA CTP results*

<table>
<thead>
<tr>
<th>XIV CIM Agent release</th>
<th>Official CTP Test Suite</th>
<th>Official CTP Test Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.2</td>
<td>• SMI-S version 1.2</td>
<td><a href="http://www.snia.org/ctp/conformingprovidersarchive/ibm.html#sftw2">http://www.snia.org/ctp/conformingprovidersarchive/ibm.html#sftw2</a></td>
</tr>
<tr>
<td></td>
<td>• Test version 1.2.0.69</td>
<td></td>
</tr>
<tr>
<td>10.2.4</td>
<td>• SMI-S version 1.4</td>
<td><a href="http://www.snia.org/ctp/conformingproviders/ibm.html#sftw3">http://www.snia.org/ctp/conformingproviders/ibm.html#sftw3</a></td>
</tr>
<tr>
<td></td>
<td>• Test version 1.4.0.24</td>
<td></td>
</tr>
<tr>
<td>11.0</td>
<td>• SMI-S version 1.4</td>
<td><a href="http://www.snia.org/ctp/conformingproviders/ibm.html#sftw3">http://www.snia.org/ctp/conformingproviders/ibm.html#sftw3</a></td>
</tr>
<tr>
<td></td>
<td>• Test version 1.4.0.24</td>
<td></td>
</tr>
<tr>
<td>11.1</td>
<td>• SMI-S version 1.4</td>
<td><a href="http://www.snia.org/ctp/conformingproviders/ibm.html#sftw3">http://www.snia.org/ctp/conformingproviders/ibm.html#sftw3</a></td>
</tr>
<tr>
<td></td>
<td>• Test version 1.4.0.24</td>
<td></td>
</tr>
</tbody>
</table>

**SCVMM Storage Automation tests**

All Microsoft SCVMM 2012 Storage Automation tests have been completed using the IBM XIV Storage System Gen3 with the XIV CIM Agent version 11.1 installed. For detailed information, please refer to [http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WeblIndex/WP102071](http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WeblIndex/WP102071)
 Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.
Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation  
Almaden Research  
650 Harry Road  
Bldg 80, D3-304, Department 277  
San Jose, CA 95120-6099  
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.
Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at Copyright and trademark information website (www.ibm.com/legal/copytrade.shtml).

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Intel, Intel Xeon, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux and the Linux logo is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other product and service names might be trademarks of IBM or other companies.
Glossary

This glossary includes terms and definitions for IBM XIV Storage System.

This glossary includes selected terms and definitions from:


- IBM Terminology, which is available online at the [IBM Terminology website](http://www.ibm.com/software/globalization/terminology/index.jsp). Definitions derived from this source have the symbol (GC) after the definition.

- The *Information Technology Vocabulary* developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1). Definitions derived from this book have the symbol (I) after the definition. Definitions taken from draft international standards, committee drafts, and working papers that the ISO/IEC JTC1/SC1 is developing have the symbol (T) after the definition, indicating that final agreement has not been reached among the participating National Bodies of SC1.

This glossary uses the following cross-reference forms:

**See** Refers the reader to one of two kinds of related information:
- A term that is the expanded form of an abbreviation or acronym. This expanded form of the term contains the full definition.
- A synonym or more preferred term

**See also** Refers the reader to one or more related terms.

**Contrast with** Refers the reader to a term that has an opposite or substantively different meaning.

**A**

**access** To obtain computing services or data.

In computer security, a specific type of interaction between a subject and an object that results in flow of information from one to the other.

**Active Directory**

Microsoft Active Directory (AD) provides directory (lookup), DNS and authentication services.

**alerting event**

An event that triggers recurring event notifications until it is cleared.

**allocated storage**

The space that is allocated to volumes but not yet assigned. Contrast with *assigned storage*.

**API** See *application programming interface (API)*.

**application programming interface (API)**

An interface that allows an application program that is written in a high-level language to use specific data or functions of the operating system or another program.

**assigned storage**

The space that is allocated to a volume and that is assigned to a port.

**Asynchronous interval**

Denotes, per given coupling, how often the master runs a new sync job.

**authorization level**

The authorization level determines the permitted access level to the various functions of the IBM XIV Storage Management GUI:

- **Read only**
  
  Only viewing is allowed.

- **Full**
  
  Access to all the configuration and control functions is allowed, including shutdown of the system. This level requires a password.
auto-delete priority
As the storage capacity reaches its limits, snapshots are automatically deleted to make more space. The deletion takes place according to the value set for each snapshot, as follows:
1 last to be deleted
4 first to be deleted
Each snapshot is given a default auto delete priority of 1 at creation.

B
basic mode
A means of entering XCLI commands on the XCLI client that requires specifying IP address and login information for each command. Additional output formatting options are available in basic mode.

best effort mode
A mode of remote mirroring in which I/O operation is not suspended when communication between a primary and secondary volume is broken.

C
call home
A communication link established between the storage system and a service provider. The storage product can use this link to call IBM or to another service provider when it requires service. With access to the storage system, service personnel can perform service tasks, such as viewing error logs and problem logs or initiating trace and dump retrievals.

clearing events
The process of stopping the recurring event notification of alerting events.

CLI
The IBM XIV command-line interface (XCLI). See command-line interface (CLI)

command-line interface (CLI)
The nongraphical user interface used to interact with the system through set commands and functions. The IBM XIV command-line interface (XCLI) for the IBM XIV Storage System.

completion code
The returned message sent as a result of running CLI commands.

consistency group
A cluster of specific volumes for which a snapshot can be taken simultaneously as a group, thus creating a synchronized snapshot. The volumes in a consistency group are grouped into a single volume set. Snapshots can be taken for the volume set in multiple snapshot sets under the specific consistency group. See also snapshot set, volume set.

coupling
A primary volume and a secondary volume connected together through mirroring definitions.

D
data availability
The degree to which data is available when needed. Availability is typically measured as a percentage of time in which the system is able to respond to data requests (for example, 99.999% available).

data module
A module dedicated to data storage. A fully populated rack contains nine dedicated data modules, each with 12 disks.

default storage pool
The default storage pool when a volume is created.

destination
See event destination.

E
escalation
A process in which event notifications are sent to a wider list of event destinations because the event was not cleared within a certain time.

event destination
An address for sending event notifications.

event notification rule
A rule that determines which users are to be notified, for which events and by what means.

event notification
The process of notifying a user about an event.
event  A user or system activity that is logged (with an appropriate message).

F fabric  The hardware that connects workstations and servers to storage devices in a SAN. The SAN fabric enables any-server-to-any-storage device connectivity through the use of Fibre Channel switching technology.

FC-AL  Also known as arbitrated loop. A Fibre Channel topology that requires no Fibre Channel switches. Devices are connected in a one-way loop fashion.

FC-HBA  Fibre Channel host bus adapter.

FC  See Fibre Channel.

Fibre Channel  Serial data transfer architecture developed by a consortium of computer and mass storage device manufacturers and now being standardized by ANSI.

functional area  One of the high-level groupings of icons (functional modules) of the left pane in the IBM XIV Storage Management GUI screen (for example, Monitor, Configuration, or Volume management). See functional module.

functional module  One of the icons of a functional area, on the left pane in the IBM XIV Storage Management GUI screen. For example, System (under Monitor) or Hosts and LUNs (under Configuration). See functional area.

G Graphical user interface (GUI)  On-screen user interface supported by a mouse and a keyboard.

GUI  See graphical user interface (GUI).

H H/W  Hardware.

HBA  Host bus adapter.

host interface module  The interface data module serves external host requests with the ability to store data. A fully populated rack has six interface data modules.

host  A port name of a host that can connect to the system. The system supports Fibre Channel and iSCSI hosts.

I I/O  input/output.

image snapshot  A snapshot that has never been unlocked. It is the exact image of the master volume it was copied from, at the time of its creation. See also snapshot.

interactive mode  A means of entering XCLI commands on the XCLI client in which the IP address, user, and password information does not need to be specified for each command.

Internet Protocol  Specifies the format of packets (also called datagrams), and their addressing schemes. See also Transmission Control Protocol (TCP).

IOPs  input/output (I/O) per second.

IP  See Internet Protocol.

iSCSI  Internet SCSI. An IP-based standard for linking data storage devices over a network and transferring data by carrying SCSI commands over IP networks.

L latency  Amount of time delay between the moment an operation is issued, and the moment it is committed.


LDAP attribute  A property of an LDAP object, with a single or multiple values. A special object attribute is designated by an LDAP administrator to hold user group memberships values corresponding to XIV roles.

LDAP authentication  A method for authenticating users by validating the submitted credentials against data stored on an LDAP directory.
**LDAP directory**
A hierarchical database stored on an LDAP server and accessed through LDAP calls.

**LDAP server**
A server that provides directory services through LDAP.

**LDAP status**
The status of an LDAP server.

**load balancing**
Even distribution of load across all components of the system.

**locking**
Setting a volume (or snapshot) as unwritable (read-only).

**LUN map**
A table showing the mappings of the volumes to the LUNs.

**LUN** Logical unit number. Exports a systems volume into a registered host.

**main-power cable**
The electrical connection between the ac power source and the automatic transfer switch (ATS).

**mandatory mode**
A mode of remote mirroring in which I/O operation stops whenever there is no communication to the secondary volume.

**master volume**
A volume that has snapshots is called the master volume of its snapshots.

**MIB** Management Information Base. A database of objects that can be monitored by a network management system. SNMP managers use standardized MIB formats to monitor SNMP agents.

**Microsoft Active directory**
See Active Directory

**mirror volume**
A volume that contains a backup copy of the original volume.

**mirroring**
See remote mirroring.

**modified State**
A snapshot state. A snapshot in modified state can never be used for restoring its master volume.

**multipathing**
Used for direct access from host-interface modules to any volume.

**P**
**peer** Denotes a constituent side of a coupling. Whenever a coupling is defined, a designation is specified for each peer - one peer is designated primary and the other is designated secondary.

**pool**
See storage pool.

**primary volume**
A volume that is mirrored for backup on a remote storage system.

**R**
**rack** The cabinet that stores all of the hardware components of the system.

**remote mirroring**
The process of replicating a volume on a remote system.

**remote target connectivity**
A definition of connectivity between a port set of a remote target and a module on the local storage system.

**remote target**
An additional storage system used for mirroring, data migration, and so on.

**role**
The actual role that the peer is fulfilling as a result of a specific condition, either a master or a subordinate.

**rule**
See event notification rule.

**S**
**SAN** Storage area network.

**SCSI** Small computer system interface.

**secondary volume**
A volume that serves as a backup of a primary volume.

**Simple Network Monitor Protocol**
A protocol for monitoring network devices. See also MIB, SNMP agent, SNMP manager, SNMP trap.

**SMS gateway**
An external server that is used to send SMSs.
SMTP gateway
An external host that is used to relay email messages through the SMTP protocol.

snapshot set
The resulting set of synchronized snapshots of a volume set in a consistency group. See also consistency group, Volume set.

snapshot
A point-in-time snapshot or copy of a volume. See also image snapshot.

SNMP agent
A device that reports information through the SNMP protocol to SNMP managers.

SNMP manager
A host that collects information from SNMP agents through the SNMP protocol.

SNMP trap
An SNMP message sent from the SNMP agent to the SNMP manager, where the sending is initiated by the SNMP agent and not as a response to a message sent from the SNMP manager.

SNMP

snooze
The process of sending recurring event notifications until the events are cleared.

storage pool
A reserved area of virtual disk space serving the storage requirements of the volumes.

Sync Job
A synchronization procedure run by the master at specified user-defined intervals, entailing synchronization between the master and the subordinate.

synchronization
The process of making the primary volume and secondary volume identical after a communication downtime or upon the initialization of the mirroring.

T

target
See remote target.

TCP/IP

thin provisioning
The ability to define logical volume sizes that are much larger than the physical capacity installed on the system.

Transmission Control Protocol
Transmission Control Protocol (TCP) on top of the Internet Protocol (IP) establishes a virtual connection between a destination and a source over which streams of data can be exchanged. See also IP.

trap
See SNMP trap.

U

unassociated volume
A volume that is not associated with a consistency group. See Consistency group.

uninterruptible power supply
Provides battery backup power for a determined time, so that the system can power down in a controlled manner, on the occurrence of a lengthy power outage.

V

volume cloning
Creating a snapshot from a volume.

volume set
A cluster of specific volumes in a consistency group, for which snapshots are taken simultaneously, thus, creating a synchronized snapshot of all of them. Snapshots of the volume set can be taken into multiple snapshot sets of the specific consistency group. See also Snapshot set, Volume set.

volume
A logical address space, having its data content stored on the systems disk drives. A volume can be virtually any size as long as the total allocated storage space of all volumes does not exceed the net capacity of the system. A volume can be exported to an attached host through a LUN. A volume can be exported to multiple hosts simultaneously. See also Storage pool, Unassociated volume.

W

WWPN
Worldwide port name
X

XCLI IBM XIV command-line interface (XCLI) command set. See command-line interface.

XCLI client
The system on which the XCLI command is entered.

XCLI identification parameters
Parameters that identify the user issuing the command and the XIV system (if any) on which the command is to run. XCLI identification parameters can be specified:
• By entering them at the beginning of an interactive mode session
• In a configuration file
• When running a command in basic mode
• When running a list of commands as a batch

XCLI system command
An XCLI command that is sent to the XIV system for processing.

XCLI utility command
An XCLI command that is issued on the XCLI client. An XCLI utility command is not sent to an XIV system for processing. XCLI utility commands are used for setting up configurations on the XCLI client and for queries (for example, of software version) that can be processed on the client.

XDRP The disaster recovery program for the XIV system – The remote mirror feature of the XIV system.

XIV mapping
An association of data on the LDAP server (a specific LDAP attribute) and data on the XIV system. The mapping is used to determine which access rights to grant to an authenticated LDAP user.
Index

A
about this document  xii
sending comments  xii
AssociatorNames 16
Associators 15

B
Block Services 26
object model 26

C
checking connectivity 6
CIM agent
components 2
concepts 3
connectivity 6
disabling 5
enabling 5
limitations 5
overview 1
port number 5
restarting 5
security 4
XIV system 5
CIM classes
CIM_AuthorizedPrivilege 41
CIM_FCPort 41
CIM_SCSIProtocolController 41
CIM_SCSIProtocolEndPoint 41
CIM_StorageHardwareID 41
CIM_StorageVolume 41
CIM_SystemSpecificCollection 41
CIM_AuthorizedPrivilege 41
CIM_ERR_ACCESS_DENIED 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20
CIM_ERR_ALREADY_EXISTS 11
CIM_ERR_FAILED 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20
CIM_ERR_INVALID_CLASS 10, 11, 12, 13, 14, 15, 17, 18, 19
CIM_ERR_INVALID_NAMESPACE 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20
CIM_ERR_INVALID_PARAMETER 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20
CIM_ERR_INVALID_QUERY 14
CIM_ERR_NO_SUCH_PROPERTY 17, 18, 19
CIM_ERR_NOT_FOUND 10, 11, 17, 18, 19, 20
CIM_ERR_NOT_SUPPORTED 10, 11, 14
CIM_ERR_QUERY_FEATURE_NOT_SUPPORTED 14
CIM_ERR_QUERY_LANGUAGE_NOT_SUPPORTED 14
CIM_ERR_TYPE_MISMATCH 19
CIM_FCPort 41
CIM_SCSIProtocolController 41
CIM_SCSIProtocolEndPoint 41
CIM_StorageHardwareID 41
CIM_StorageVolume 41
CIM_SystemSpecificCollection 41
CIMOM
return codes 20

D
DeleteClass 10
DeleteInstance 10
DeleteQualifier 20
disabling
CIM agent 5
documentation
improvement  xii

E
enabling
CIM agent 5
EnumerateClasses 12
EnumerateInstanceNames 14
EnumerateInstances 13
EnumerateQualifiers 20
error codes 20
ExecuteQuery 14

F
forums ix

G
GetClass 9
GetInstance 10
GetProperty 18
GetQualifier 19
group management
Replication Services 49

H
hosts 39
HTTP
status messages 20

I
IBM Tivoli Storage Productivity Center 6
IBMTSDS_FCPort 41
L
legal notices 59
LUN mapping 39
LUN masking 39

M
Masking and Mapping 39
object model 41
methods
  AssociatorNames 16
  Associators 15
  CreateClass 11
  CreateInstance 11
  DeleteClass 10
  DeleteInstance 10
  DeleteQualifier 20
  EnumerateClasses 12
  EnumeratedInstanceNames 14
  EnumeratedInstances 13
  EnumerateQualifiers 20
  ExecuteQuery 14
  GetClass 9
  GetInstance 10
  GetProperty 18
  GetQualifier 19
  ModifyClass 11
  ModifyInstance 11
  ReferenceNames 17
  References 17
  SetProperty 19
  SetQualifier 20
models
  Block Services 26
  Masking and Mapping 41
  Replication Services 48
  ModifyClass 11
  ModifyInstance 11

N
notices
  legal 57

P
packages
  Block Services 26
  Masking and Mapping 41
  Replication Services 48
parameters
  AssocClass 15, 16
  ClassName 9, 12, 13, 14
  DeepInheritance 12, 13
  IncludeClassOrigin 9, 10, 12, 13, 15, 17
  IncludeQualifiers 9, 12
  Instance 11
  InstanceName 10, 18, 19
parameters (continued)
  LocalOnly 9, 12
  ObjectName 15, 16, 17
  Property 18, 19
  QualifierName 19
  Query 14
  QueryLanguage 14
  ResultClass 15, 16, 17
  ResultRole 15, 16
  Role 15, 16, 17
  SetName 20
PDFs ix
  port number 5
ports 39
profiles
  Block Services 26
  Masking and Mapping 39
publications ix
R
reader feedback, sending xii
ReferenceNames 17
References 17
related information ix
Replication management 51
Replication Services
  group management 49
  object model 48
restarting
  CIM agent 5
return code 20
return values
  CIM_ERR_ACCESS_DENIED 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20
  CIM_ERR_ALREADY_EXISTS 11
  CIM_ERR_FAILED 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20
  CIM_ERR_INVALID_CLASS 10, 11, 12, 13, 14, 15, 17, 18, 19
  CIM_ERR_INVALID_NAMESPACE 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20
  CIM_ERR_INVALID_PARAMETER 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20
  CIM_ERR_INVALID_QUERY 14
  CIM_ERR_NO_SUCH_PROPERTY 17, 18, 19
  CIM_ERR_NOT_FOUND 10, 11, 17, 18, 19, 20
  CIM_ERR_NOT_SUPPORTED 10, 11, 14, 20
  CIM_ERR_QUERY_FEATURE_NOT_SUPPORTED 14
  CIM_ERR_QUERY_LANGUAGE_NOT_SUPPORTED 14
  CIM_ERR_TYPE_MISMATCH 19
descriptions 20
S
SCVMM testing 55
secure connection
  port number 5
sending
  comments xii
  SetProperty 19
  SetQualifier 20
  SLP discovery 6
  slptool findsrvs 6
  SNAI testing 55
starting
  CIM agent  5
storage pools  26, 39

T
trademarks  59

V
volumes  26, 39

X
XIV CIM classes
  IBMTSDS_FCPort  41
  IBMTSDS_Privilege  41
  IBMTSDS_SCSIProtocolController  41
  IBMTSDS_SEVolume  41
  IBMTSDS_StorageHardwareID  41
  IBMTSDS_SystemSpecificCollection  41
XIV Open API
  components  7
  overview  1