

IBM XIV Storage System
Version 11.6

*Application Programming Interface
(API) Reference Guide*



Note

Before using this document and the product it supports, read the information in Notices.

Publication number: GC27-3916-07. This publication applies to version 11.6 of the IBM XIV Storage System Open Application Programming Interface (API) and to all subsequent releases and modifications until otherwise indicated in a newer publication.

© **Copyright IBM Corporation 2010, 2015.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

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	ix
Ordering publications	x
Sending or posting your comments	x

Chapter 1. XIV Open API overview 1

CIM agent overview	1
CIM agent components	3
CIM concepts	3
CIM agent security	4

Chapter 2. CIM agent for the XIV system 7

CIM agent installation overview	7
Configuring the CIM agent	7
Enabling the CIM agent	8
Verifying the CIM agent connection	8

Chapter 3. XIV Open API component definitions 9

CIM agent communication concepts	10
CIM agent communication methods	10
GetClass	11
GetInstance	12
DeleteClass	12
DeleteInstance	12
CreateClass	13
CreateInstance	13
ModifyClass	14
ModifyInstance	14
EnumerateClasses	14
EnumerateClassNames	15
EnumerateInstances	16
EnumerateInstanceNames	16
ExecuteQuery	17
Associators	18
AssociatorNames	18
References	19
ReferenceNames	20
GetProperty	21
SetProperty	22
GetQualifier	22
SetQualifier	23
DeleteQualifier	23
EnumerateQualifiers	23
Return error codes	24

Chapter 4. Functional diagrams of the CIM agent 27

Block Server performance profile	27
Block Server Performance object model	28
IBMTSDS_PerformanceStatisticsService. GetStatisticsCollection	30
Getting volume or host statistics with GetStatisticsCollection	31
Block Services profile	31
Block Services object model	32
IBMTSDS_StorageConfigurationService. CreateOrModifyStoragePool	36
IBMTSDS_StorageConfigurationService. DeleteStoragePool	37
IBMTSDS_StorageConfigurationService. CreateOrModifyElementsFromStoragePool	37
IBMTSDS_StorageConfigurationService. CreateOrModifyElementFromStoragePool	38
IBMTSDS_StorageConfigurationService. ReturnToStoragePool	39
IBMTSDS_StorageConfigurationService. ReturnElementsToStoragePool	39
IBMTSDS_PrimitiveStoragePool. GetSupportedSizeRange	39
IBMTSDS_VirtualPool.GetSupportedSizeRange	40
Creating a ConcreteStoragePool	40
Creating a StorageVolume	40
iSCSI Target Ports profile	41
iSCSI Target Ports Object Model	41
IBMTSDS_iSCSIConfigurationService. CreateIPProtocolEndpoint	44
IBMTSDS_iSCSIConfigurationService. ModifyIPProtocolEndpoint	45
IBMTSDS_iSCSIConfigurationService. DeleteIPProtocolEndpoint	45
Creating an IP endpoint	45
Modifying an IP endpoint	46
Deleting an IP endpoint	46
Masking and Mapping profile	47
Masking and mapping object model	49
IBMTSDS_StorageHardwareIDManagement. CreateStorageHardwareID	51
IBMTSDS_StorageHardwareIDManagement. DeleteStorageHardwareID	51
IBMTSDS_StorageHardwareIDManagement. CreateHardwareIDCollection	52
IBMTSDS_StorageHardwareIDManagement. AddHardwareIDsToCollection	52
IBMTSDS_ControllerConfigurationService. ExposePaths	52
IBMTSDS_ControllerConfigurationService. HidePaths	53
IBMTSDS_ControllerConfigurationService. DeleteProtocolController	54
Sample recipe	54

Indication profile	55
Replication Services profile	56
Replication Services object model	57
Group management	60
IBMTSDS_ReplicationService. CreateGroup	61
IBMTSDS_ReplicationService. DeleteGroup	61
IBMTSDS_ReplicationService. AddMembers	61
IBMTSDS_ReplicationService. RemoveMembers	62
Replication management	62
IBMTSDS_ReplicationService. CreateElementReplica	64
IBMTSDS_ReplicationService. CreateGroupReplica	65
IBMTSDS_ReplicationService. ModifyReplicaSynchronization	65
IBMTSDS_ReplicationService. ModifyListSynchronization	66
IBMTSDS_ReplicationService. GetAvailableTargetElements	66
IBMTSDS_ReplicationService. GetReplicationRelationships	67
IBMTSDS_ReplicationService. GetReplicationRelationshipInstances	67
IBMTSDS_ReplicationService. GetPeerSystems	67

IBMTSDS_ReplicationService. CreateGroupReplicaFromElements	67
Job Control profile	69
Job Control object model	69
Thin Provisioning profile	70
Thin Provisioning object model	70
IBMTSDS_PrimordialStoragePoolCapabilities. CreateSetting	71
IBMTSDS_VirtualPoolSetting. modify	72
IBMTSDS_VirtualPoolSetting. delete	72
IBMTSDS_StorageConfigurationService. CreateOrModifyStoragePool	72
Creating a thin provisioning pool	73
Thin provisioning indications	74

Chapter 5. Conformance tests. 75

Notices	77
Trademarks	79

Glossary 81

Index 87

Figures

1. How a CIM agent works	2	8. Masking and mapping object model in SMI-S	49
2. The MOF compiler stores the model in the CIMOM data store.	4	9. Replication Services (Local) SMI-S model for XIV systems	58
3. Block Services Performance SMI-S model for XIV systems	29	10. Replication Services (Remote) SMI-S model for XIV systems	59
4. Block Services SMI-S model for XIV systems	33	11. Sample local group information retrieval	61
5. Block Services Package with Settings and Capabilities model	34	12. Sample remote group information retrieval	63
6. iSCSI Target Ports SMI-S model for XIV systems	42	13. Association classes for mirrored volumes and consistency groups	63
7. Masking and mapping physical model in the XIV system	48	14. Job Control SMI-S model for XIV systems	70
		15. Thin Provisioning SMI-S model for XIV systems	71

Tables

1. Functional groups for the CIM agent	11	21. Block Server Performance metrics	27
2. GetClass method parameters	11	22. Synchronous actions	35
3. GetInstance method parameters	12	23. iSCSI terminology and CIM class names	43
4. DeleteInstance method parameters	13	24. Masking and mapping classes	50
5. CreateInstance method parameters	13	25. ExposePaths parameters and use cases	53
6. ModifyInstance method parameters	14	26. HidePaths parameters and use cases	53
7. EnumerateClasses method parameters	14	27. Indication types and object classes	55
8. EnumerateClassNames method parameters	15	28. Mapping XIV terminology to SMI terminology	59
9. EnumerateInstances method parameters	16	29. Replication Service methods	60
10. EnumerateInstanceNames method parameters	17	30. Group management classes	60
11. ExecuteQuery method parameters	17	31. Extrinsic methods for group management	61
12. Associators method parameters	18	32. Replication management classes.	62
13. Associators method parameters	19	33. Extrinsic methods for replication management	64
14. References method parameters	20	34. Operations, operation descriptions, and corresponding WaitForCopyState states	66
15. ReferenceNames method parameters	20	35. SMI Indications	68
16. GetProperty method parameters	21	36. Replication Service methods	71
17. SetProperty method parameters	22	37. Thin provisioning profile Indications	74
18. GetQualifier method parameters	22	38. SNIA CTP results	75
19. SetQualifier method parameters.	23		
20. Return error codes for the CIMOM.	24		

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 Open Application Programming Interface (API) microcode version 11.6.

Who should use this guide

This publication is for system administrators and system and application programmers, or whomever 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

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

- IBM XIV Storage System on the IBM Knowledge Center (ibm.com/support/knowledgecenter/STJTAG)
- Storage Networking Industry Association (SNIA) website (www.snia.org)
- Distributed Management Task Force (DMTF) website (www.dmtf.org)

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.

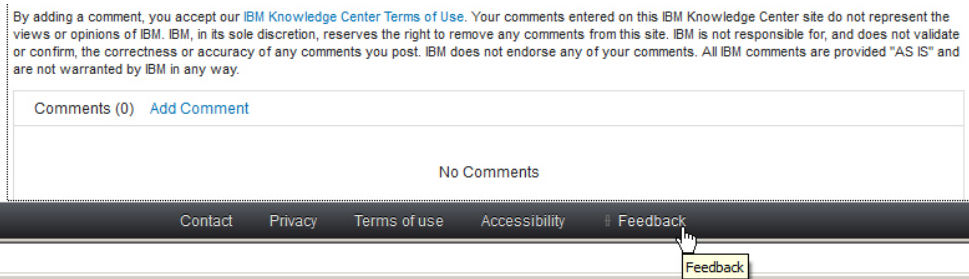
Sending or posting your comments

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

Procedure

To submit any comments about this guide:

- Go to IBM Knowledge Center (ibm.com/support/knowledgecenter), drill down to the relevant page, and then click the **Feedback** link that is located at the bottom of the page.



The feedback form is displayed and you can use it to enter and submit your comments privately.

- You can post a public comment on the Knowledge Center page that you are viewing, by clicking **Add Comment**. For this option, you must first log in to IBM Knowledge Center with your IBM ID.
- You can send your comments by email to starpubs@us.ibm.com. Be sure to include the following information:
 - Exact publication title and product version
 - Publication form number (for example: SC01-0001-01)
 - Page, table, or illustration numbers that you are commenting on
 - A detailed description of any information that should be changed

Note: When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

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.

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 that are 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.

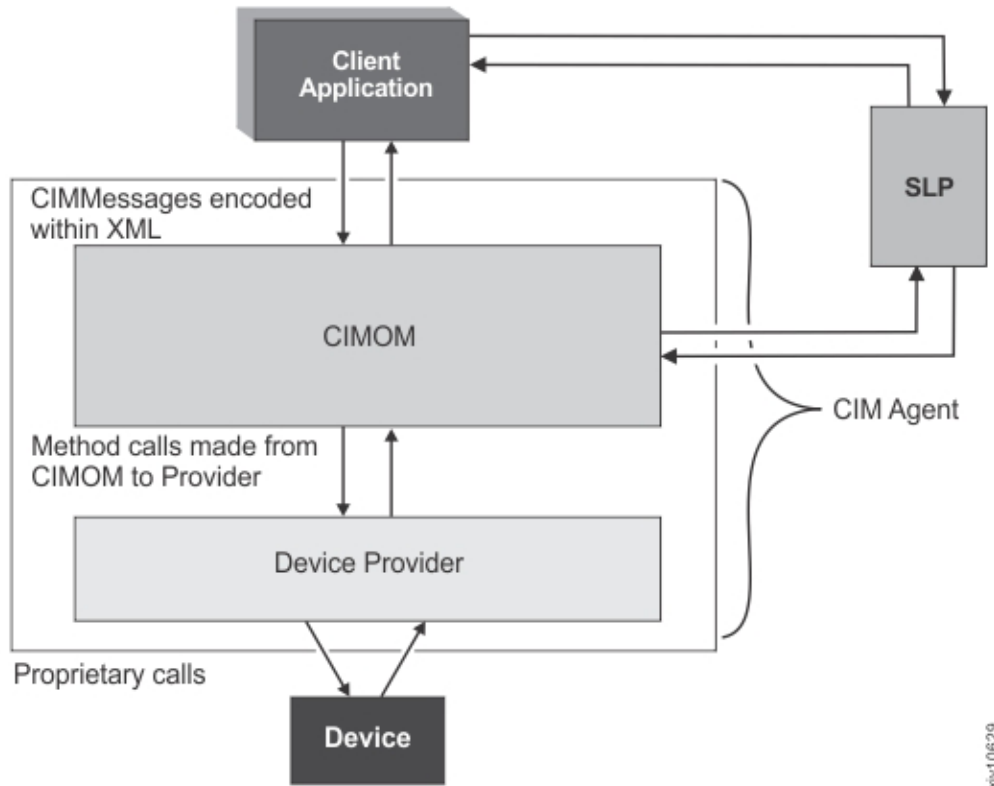


Figure 1. How a CIM agent works

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 2.40.0
- Storage Networking Industry Association (SNIA) Storage Management Initiative Specification (SMI-S) and the Shared Storage Model, a framework for describing storage architectures, Version 1.6

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

When you install, configure, and enable 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. CIM-compliant applications like the CIM agent can manage the data on your XIV system.

The following sites provide more information about the CIM standards:

- DMTF Common Information Model (CIM) Standards
<http://www.dmtf.org/standards/cim/>
- Storage Networking Industry Association Standards
http://www.snia.org/tech_activities/standards/curr_standards/smi/
- DMTF Common Information Model (CIM) Standards
- 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 that enables 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, which includes systems, applications, databases, networks, and devices. The CIM specification provides the language and the methodology that is 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. It 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 this information is obtained, 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 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.

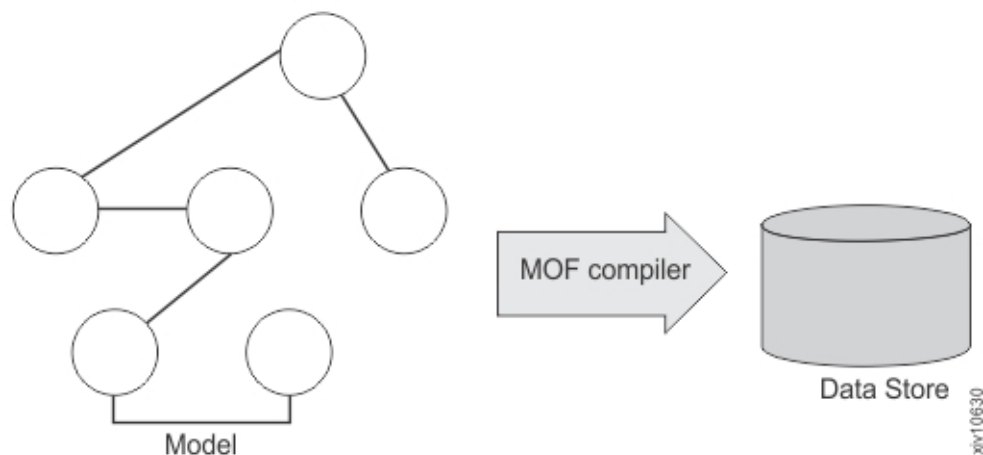


Figure 2. The MOF compiler stores the model in the CIMOM data store.

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. Then, the provider might make one or many requests of a device by 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. They are 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 by 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

The CIM agent is preinstalled on the administrative module. The embedded CIM agent is enabled automatically 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 authenticate. To manage accounts, you must use the IBM XIV Storage Management GUI or the IBM XIV command-line interface (XCLI).

CIM agent installation overview

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

The embedded CIM agent does not require configuration changes to manage the XIV system.

About this task

For the IBM XIV Storage System version 10.2.4.x, 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 complete any functions. The XIV system administrator can use the XCLI or the IBM XIV Storage Management GUI to create user names and passwords.

For the IBM XIV Storage System Gen3 version 11.5 or later, the Multitenancy feature is supported. To support this feature, XIV implements the concept of a domain, which will link users to their dedicated pools, hosts, mirror targets, and other resources. A user can manage only the storage resources to which they are associated, without the ability to modify or monitor other system resources. The domain restricts the set of objects a user can manage to those associated with the domain. If a user is associated with one or more domains, they are a domain user, otherwise, they are a global user. This means that users in the same category, but different domain will manage different resources.

For the IBM XIV Storage System Gen3 version 11.5 or later, the SMI-S Agent does *not* support the Multitenancy feature. CIM only supports the global user with the domain policy access value of OPEN. CIM requires a global Storage administrator/Application administrator/Read-only user who can manage all the resources in the system.

See the IBM(r) XIV(r) Storage System: Architecture, Implementation, and Usage (www.redbooks.ibm.com/abstracts/sg247659.html) and the IBM XIV Storage System Product Overview for more information about the architecture of the XIV system.

Enabling the CIM agent

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?uid=ssg1S4000879>.

The `slptool` is included in the DSCIMCLI package. You can download the DSCIMCLI package from the IBM CIM Agent for DS Open API site.

Procedure

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

Verify that the CIM agent registered into the SLP DA 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 whether the CIM agent is running on the device. If the discovery process fails, you can initiate 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 that are defined to a single namespace. Within the CIM agent, the schemas that are supported are the ones that are 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 are 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 XIV Open API.

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 XIV Open API. 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

The CIM agent uses client communication and intrinsic and extrinsic methods of communication.

Client communication

A client application communicates with the CIM agent through operation request messages that are encoded within XML. The CIM agent returns responses with operation response messages. Requests and responses are sub-elements 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 supports only simple operation requests and responses by examining the output of the OPTIONS method.

Intrinsic and Extrinsic Methods

All operations on the CIM agent are completed 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 by 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 can result in CIM agent calls to the device provider. This result happens when 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 on page 11 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

Functional group	Parameters	Supported or not supported
Basic read	GetInstance EnumerateInstances EnumerateInstanceNames GetProperty	Supported
Basic write	SetProperty	Not Supported
Schema manipulation	CreateClass ModifyClass DeleteClass	Not Supported
Instance manipulation	CreateInstance ModifyInstance DeleteInstance	Not Supported
Association traversal	Associators AssociatorNames References RefernceNames	Supported
Qualifier read	GetQualifier EnumerateQualifiers	Supported
Qualifier manipulation	SetQualifier DeleteQualifier	Not Supported
Query execution	ExecQuery	Supported

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

Parameter	Type	Description
ClassName	String	Defines the name of the class you want to retrieve.
LocalOnly	Boolean	TRUE returns all properties, methods, and qualifiers that are overridden within the definition of the class.
IncludeQualifiers	Boolean	TRUE returns all qualifiers for the class, its properties, methods, or method parameters. FALSE returns no qualifiers.

Table 2. *GetClass* method parameters (continued)

Parameter	Type	Description
IncludeClassOrigin	Boolean	TRUE returns the CLASSORIGIN attribute of the class.

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 3. *GetInstance* method parameters

Parameter	Type	Description
InstanceName	String	Defines the name of the instance to retrieve.
IncludeClassOrigin	Boolean	TRUE returns the CLASSORIGIN attribute of the class.

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.

Parameters

Table 4 describes the parameters of the **DeleteInstance** method.

Table 4. DeleteInstance method parameters

Parameter	Type	Description
InstanceName	String	Defines the name of the instance you want to delete.

Return values

The named instance is deleted or 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**

CreateClass

The **CreateClass** method creates 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 5. CreateInstance method parameters

Parameter	Type	Description
Instance	Object	The instance to be created. The instance must be based on a class that is already defined in the target namespace.

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_INVALID_CLASS**
- **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.

Parameters

Table 6 describes the parameters of the **ModifyInstance** method.

Table 6. ModifyInstance method parameters

Parameter	Type	Description
Instance	Object	Defines the modified instance.

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 7. EnumerateClasses method parameters

Parameter	Type	Description
ClassName	String	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.

Table 7. EnumerateClasses method parameters (continued)

Parameter	Type	Description
DeepInheritance	Boolean	TRUE returns all subclasses of the specified class. FALSE returns only immediate child subclasses.
LocalOnly	Boolean	TRUE returns all properties, methods, and qualifiers, that are overridden within the definition of the class.
IncludeQualifiers	Boolean	TRUE returns all qualifiers for the class, its properties, methods, or method parameters. FALSE returns no qualifiers.
IncludeClassOrigin	Boolean	TRUE returns the CLASSORIGIN of the class.

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**
- **CIM_ERR_INVALID_PARAMETER**
- **CIM_ERR_INVALID_CLASS**
- **CIM_ERR_FAILED**

EnumerateClassNames

The **EnumerateClassNames** method enumerates the names of subclasses of a class that is defined within the target namespace.

Parameters

Table 8 describes the parameters of the **EnumerateClassNames** method.

Table 8. EnumerateClassNames method parameters

Parameter	Type	Description
ClassName	String	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.
DeepInheritance	Boolean	TRUE returns all subclass names of the specified class. FALSE returns only immediate child subclass names.

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 9. EnumerateInstances method parameters

Parameter	Type	Description
ClassName	String	Defines the name of the class for which instances are to be returned.
DeepInheritance	Boolean	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.
IncludeClassOrigin	Boolean	TRUE returns the CLASSORIGIN attribute of the class within the instance.

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 10. *EnumerateInstanceNames* method parameters

Header	Header	Description
ClassName	String	Defines the name of the class for which instance names are returned.

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 11. *ExecuteQuery* method parameters

Parameter	Type	Description
QueryLanguage	String	Defines the query language in which the query parameter is expressed.
Query	String	Defines the query to be initiated.

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 12. *Associators method parameters*

Parameter	Type	Description
ObjectName	String	Defines the class name or instance name that is the source of the association.
AssocClass	String	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.
ResultClass	String	If not NULL, indicates that all returned objects must be instances of this class or one of its subclasses or be this class.
Role	String	If not NULL, indicates that each return object must be associated with the source object that 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.
ResultRole	String	If not NULL, indicates that each returned object must be associated with the source object that 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.
IncludeClassOrigin	Boolean	TRUE returns the CLASSORIGIN attribute of the class.

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 13. *Associators method parameters*

Parameter	Type	Description
ObjectName	String	Defines the class name or instance name that is the source of the association.
AssocClass	String	If not NULL, indicates that returned objects are associated with the source object through an instance of this class or one of its subclasses.
ResultClass	String	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.
Role	String	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.
ResultRole	String	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.

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_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 14. *References method parameters*

Parameter	Type	Description
ObjectName	String	Defines the class name or instance name whose referring objects are to be returned.
ResultClass	String	If not NULL, indicates that all returned objects must be instances of this class or one of its subclasses or must be this class.
Role	String	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.
IncludeClassOrigin	Boolean	TRUE returns the CLASSORIGIN attribute of the class.

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 15. *ReferenceNames method parameters*

Parameter	Type	Description
ObjectName	String	Defines the class name or instance name whose referring objects are to be returned.

Table 15. ReferenceNames method parameters (continued)

Parameter	Type	Description
ResultClass	String	If not NULL, indicates that all returned objects must be instances of this class or one of its subclasses or must be this class.
Role	String	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.

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

Parameter	Type	Description
InstanceName	String	Defines the name of the instance.
Property	String	The name of the property whose value is to be returned from the instance.

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 17. SetProperty method parameters

Parameter	Type	Description
InstanceName	String	Defines the name of the instance.
Property	String	The name of the property whose value is to be returned from the instance.

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 18. GetQualifier method parameters

Parameter	Type	Description
QualifierName	String	Defines the qualifier whose declaration is to be returned.

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 19. SetQualifier method parameters

Parameter	Type	Description
QualifierDeclaration	Void	Defines the qualifier declaration to be added to the target namespace.

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 one of the following ways:

- Through HTTP status messages
- Through error codes that are 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

Code	Symbolic Name	Definition
1	CIM_ERR_FAILED	A general error occurred that is not covered by a more specific error code.
2	CIM_ERR_ACCESS_DENIED	Access to a CIM resource was not available to the client.
3	CIM_ERR_INVALID_NAMESPACE	The target namespace does not exist.
4	CIM_ERR_INVALID_PARAMETER	One or more parameter values that are passed to the method were not valid.
5	CIM_ERR_INVALID_CLASS	The specified class does not exist.
6	CIM_ERR_NOT_FOUND	The requested object was not found.
7	CIM_ERR_NOT_SUPPORTED	The requested operation is not supported.
8	CIM_ERR_CLASS_HAS_CHILDREN	The operation cannot be carried out on this class because it has subclasses.
9	CIM_ERR_CLASS_HAS_INSTANCES	The operation cannot be carried out on this class because it has instances.
10	CIM_ERR_INVALID_SUPERCLASS	The operation cannot be carried out because the specified superclass does not exist.
11	CIM_ERR_ALREADY_EXISTS	The operation cannot be carried out because the object exists.
12	CIM_ERR_NO_SUCH_PROPERTY	The specified property does not exist.
13	CIM_ERR_TYPE_MISMATCH	The value that is supplied is not compatible with the type that is specified.
14	CIM_ERR_QUERY_LANGUAGE_NOT_SUPPORTED	The query language is not recognized or supported.
15	CIM_ERR_INVALID_QUERY	The query is not valid for the specified query language.
16	CIM_ERR_METHOD_NOT_AVAILABLE	The extrinsic method cannot be run.
17	CIM_ERR_METHOD_NOT_FOUND	The specified extrinsic method does not exist.
20	CIM_ERR_LOW_ON_MEMORY	There is not enough memory.
21	XMLERROR	An XML error occurred.

Table 20. Return error codes for the CIMOM (continued)

Code	Symbolic Name	Definition
22	CIM_ERR_LISTNER_ALREADY_DEFINED	The listener is already defined.
23	CIM_ERR_INDICATION_NOT_COLLECTED	The indications are not collected.
24	CIM_ERR_NO_METHOD_NAME	The method name is null.
25	CIM_ERR_INVALID_QUALIFIER_DATATYPE	The data type qualifier is not valid.
26	CIM_ERR_NAMESPACE_NOT_IN_MANAGER	The namespace value is not found.
27	CIM_ERR_INSTANTIATE_FAILED	The instantiation failed.
28	CIM_ERR_FAILED_TO_LOCATE_INDICATION_HANDLER	The indication handler is not found.
29	CIM_ERR_IO_EXCEPTION	An I/O exception occurred.
30	CIM_ERR_COULD_NOT_DELETE_FILE	The file cannot be deleted.
31	INVALID_QUALIFIER_NAME	The qualifier name is null.
32	NO_QUALIFIER_VALUE	The qualifier value is null.
33	NO_SUCH_QUALIFIER1	There is no such qualifier.
34	NO_SUCH_QUALIFIER2	There is no such qualifier.
35	QUALIFIER_UNOVERRIDABLE	The qualifier cannot be overwritten.
36	SCOPE_ERROR	A scope error occurred.
37	TYPE_ERROR	A type error occurred.
38	CIM_ERR_MISSING_KEY	The key is missing.
39	CIM_ERR_KEY_CANNOT_MODIFY	The key cannot be modified.
40	CIM_ERR_NO_KEYS	There are no keys found.
41	CIM_ERR_KEYS_NOT_UNIQUE	The keys are not unique.
100	CIM_ERR_SET_CLASS_NOT_SUPPORTED	The set class operation is not supported.
101	CIM_ERR_SET_INSTANCE_NOT_SUPPORTED	The set instance operation is not supported.
102	CIM_ERR_QUALIFIER_NOT_FOUND	The qualifier value is not found.
103	CIM_ERR_QUALIFIERTYPE_NOT_FOUND	The qualifier type is not found.
104	CIM_ERR_CONNECTION_FAILURE	The connection failed.
105	CIM_ERR_FAIL_TO_WRITE_TO_SERVER	There is a fail to write to the server.
106	CIM_ERR_SERVER_NOT_SPECIFIED	The server is not specified.
107	CIM_ERR_INDICATION_ERROR	There is an indication processing error.
108	CIM_ERR_FAIL_TO_WRITE_TO_CIMOM	A write operation to the CIMOM failed.
109	CIM_ERR_SUBSCRIPTION_EXISTS	A subscription exists.
110	CIM_ERR_INVALID_SUBSCRIPTION_DEST	The subscription destination is not valid.
111	CIM_ERR_INVALID_FILTER_PATH	The filter path is not valid.
112	CIM_ERR_INVALID_HANDLER_PATH	The handler path is not valid.
113	CIM_ERR_NO_FILTER_INSTANCE	The filter instance is not found.
114	CIM_ERR_NO_HANDLER_INSTANCE	The handler instance is not found.
115	CIM_ERR_UNSUPPPORTED_FILTER	The filter that is referenced in the subscription is not supported.
116	CIM_ERR_INVALID_TRUSTSTORE	The CIMOM cannot be connected to because there is a bad or missing truststore or an incorrect truststore password.

Table 20. Return error codes for the CIMOM (continued)

Code	Symbolic Name	Definition
117	CIM_ERR_ALREADY_CONNECTED	The CIMOM cannot be connected to because it is already connected.
118	CIM_ERR_UNKNOWN_SERVER	The server is unknown. The CIMOM cannot accept connections.
119	CIM_ERR_INVALID_CERTIFICATE	The correct certificate cannot be found in the truststore. The CIMOM cannot accept connections.

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. It uses a number of metrics with definitions that are 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 and hosts.

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 Performance metrics

	Equivalent CIM class	Corresponding ElementType value	Properties that are supplied in associated CIM_BlockStorageStatisticalData instances
Volume	IBMTSDS_SEVolumeStatistics	8	TotalIOs KBytesTransferred KBytesRead KBytesWritten ReadIOs ReadHitIOs WriteIOs WriteHitIOs IOTimeCounter ReadIOTimeCounter WriteIOTimeCounter ReadHitIOTimeCounter WriteHitIOTimeCounter
Host	IBMTSDS_HostStatistics	4	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 is a set of classes that are required to use an extrinsic method to retrieve a string that contains a batch of performance statistics.

See Figure 3 on page 29 for the Block Services Performance SMI-S model for XIV systems.

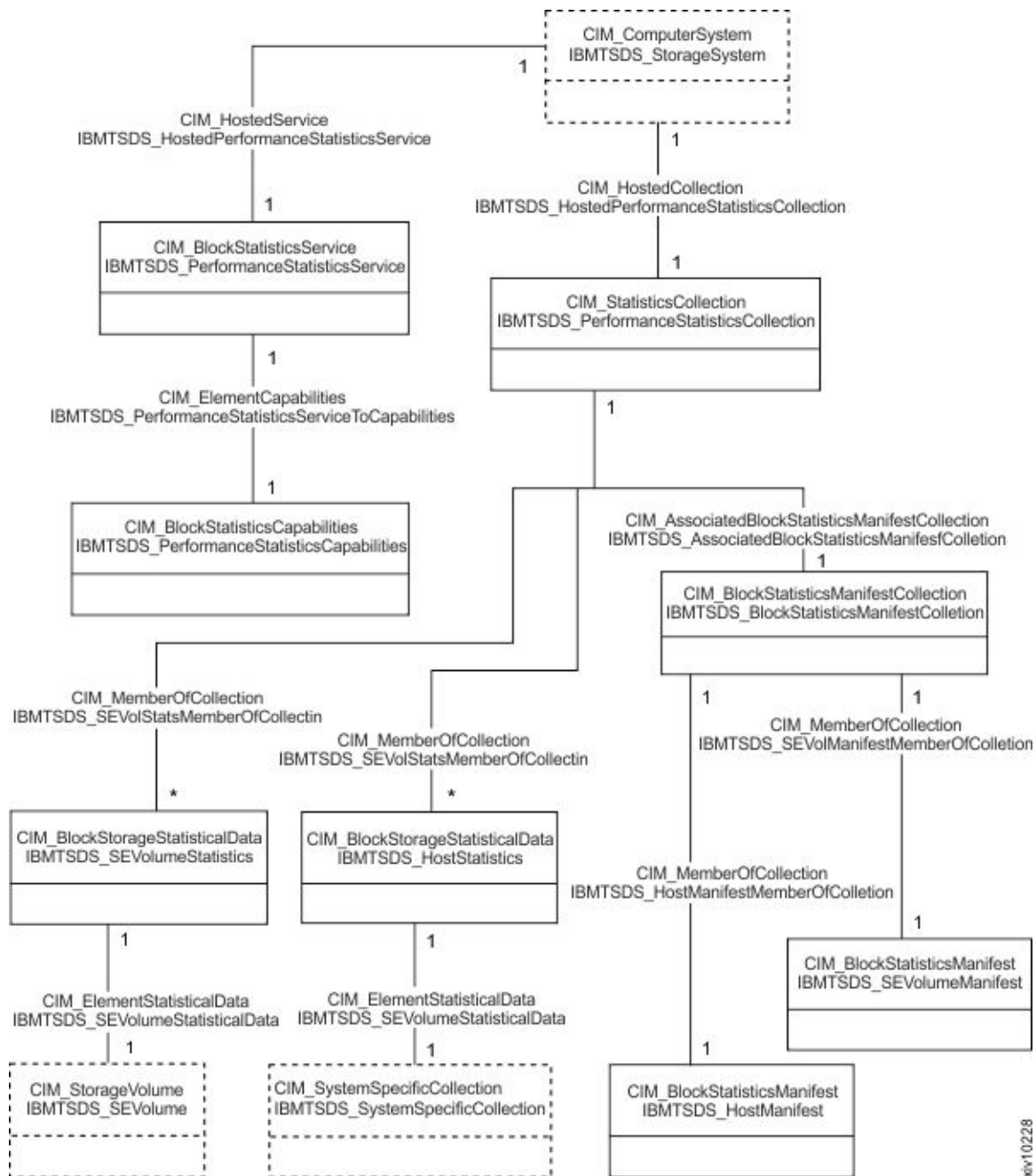


Figure 3. Block Services Performance SMI-S model for XIV systems

The classes that inherit from **CIM_BlockStorageStatisticalData** **IBMTSDS_SEVolumeStatistics** and **IBMTSDS_HostStatistics** are of the first category. Each instance contains properties that describe the performance statistics of a single instance of the **IBMTSDS_SEVolume** or **IBMTSDS_SystemSpecificCollection**. **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 that are 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 about 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 to calculate I/O rates, it must know when the counters roll back to zero.

IBMTSDS_PerformanceStatisticsService. GetStatisticsCollection

The **IBMTSDS_PerformanceStatisticsService.GetStatisticsCollection** method returns a string representation of a set of performance statistics data.

ElementTypes

An array of values that indicates the type of element that returns statistics. Volumes and hosts use the standard value in the CIM schema. If this parameter is left null, the default value 4 and 8 are used. See the **BlockServerPerformance.mof** file 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**. It is a **ManifestCollection** because the method can return statistics for multiple **ElementTypes** at the same time, and each **ElementType** has its own **BlockStatisticsManifest**. The CIM agent currently 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;1078871;67382;1685;1673;19744570;19692176;99538;IBM.2810-6000095-100930;8;20110118225313.891038+480;8386;504537;76662400;1426;1422;1182723;1133669;67382;6960;6940;75479677;75360317;437155;
```


where each row is an array element. The **ElementType** of each row is 8, which refers to volumes. So there are two instances of **VolumeStatistics** with attributes of each in 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 or host 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 that you can use the XIV Open API to configure XIV devices.

Procedure

1. Use `enumerateInstanceNames` on **IBMTSDS_PerformanceStatisticsService** and then save the reference.
2. Use `enumerateInstancesNames` on **IBMTSDS_BlockStatisticsManifestCollection** and then save the reference.
3. Use `enumerateInstances` on **IBMTSDS_VolumeManifest** (or **CIM_BlockStatisticsManifest** and look for **ElementType** 8) or **IBMTSDS_SystemSpecificCollection** (or **CIM_BlockStatisticsManifest** and look for **ElementType** 4), and then save **BulkFormat**.
4. Use `invokeMethod` on `GetStatisticsCollection` with **ElementType** 8 for volume or **ElementType** 4 for host (if **ElementType** is null, 4 and 8 is assigned to **ElementType**), **StatisticsFormat** 2, and the **ManifestCollection** obtained in the second step, on the **PerformanceStatisticsCollection** instance that is obtained in the first step.
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 provides a means for 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 contains 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. It 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. A **VirtualPool** without its own **SnapshotPool** cannot contain snapshots.

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* on the IBM Redbooks® website (www.redbooks.ibm.com) and the *IBM XIV Storage System Product Overview* 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 10.2.4.x, SMI-S specification 1.4 is fully supported. .

For the 11.6 release, SMI-S specification 1.6 is fully supported. You can create, delete, or modify services

See Figure 4 on page 33 for the Block Services SMI-S model for XIV systems.

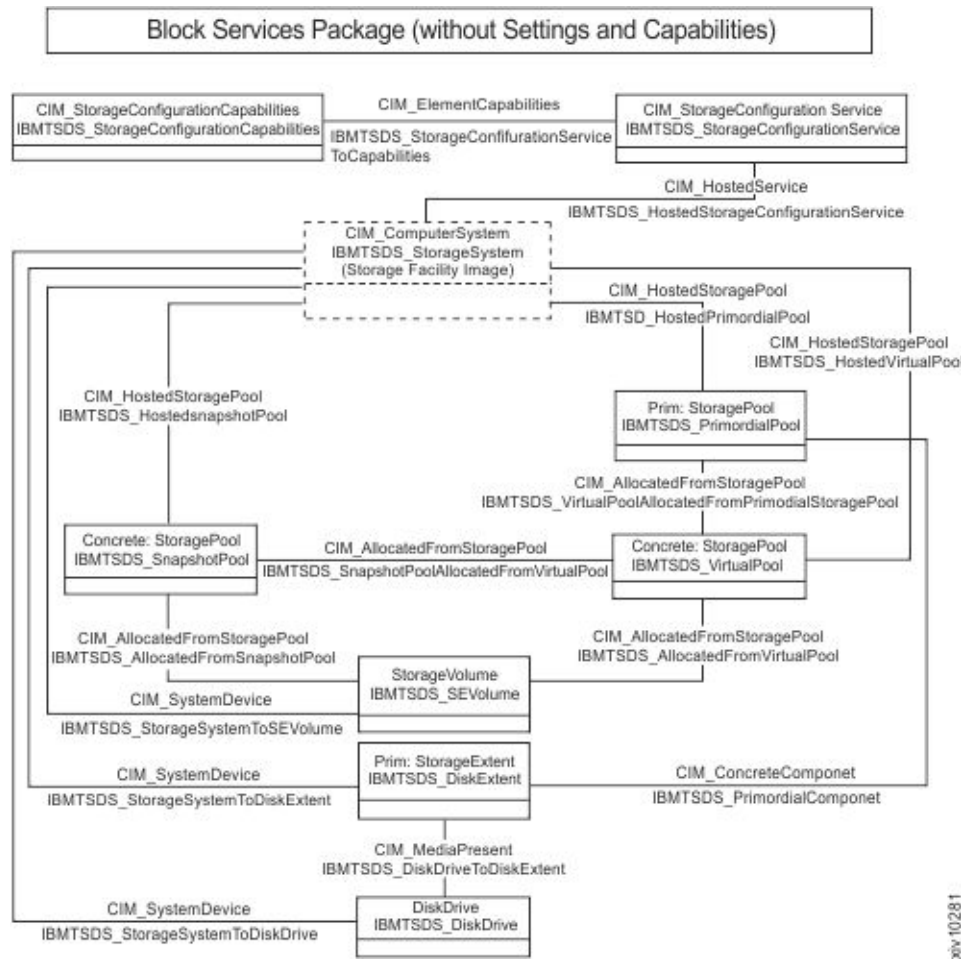
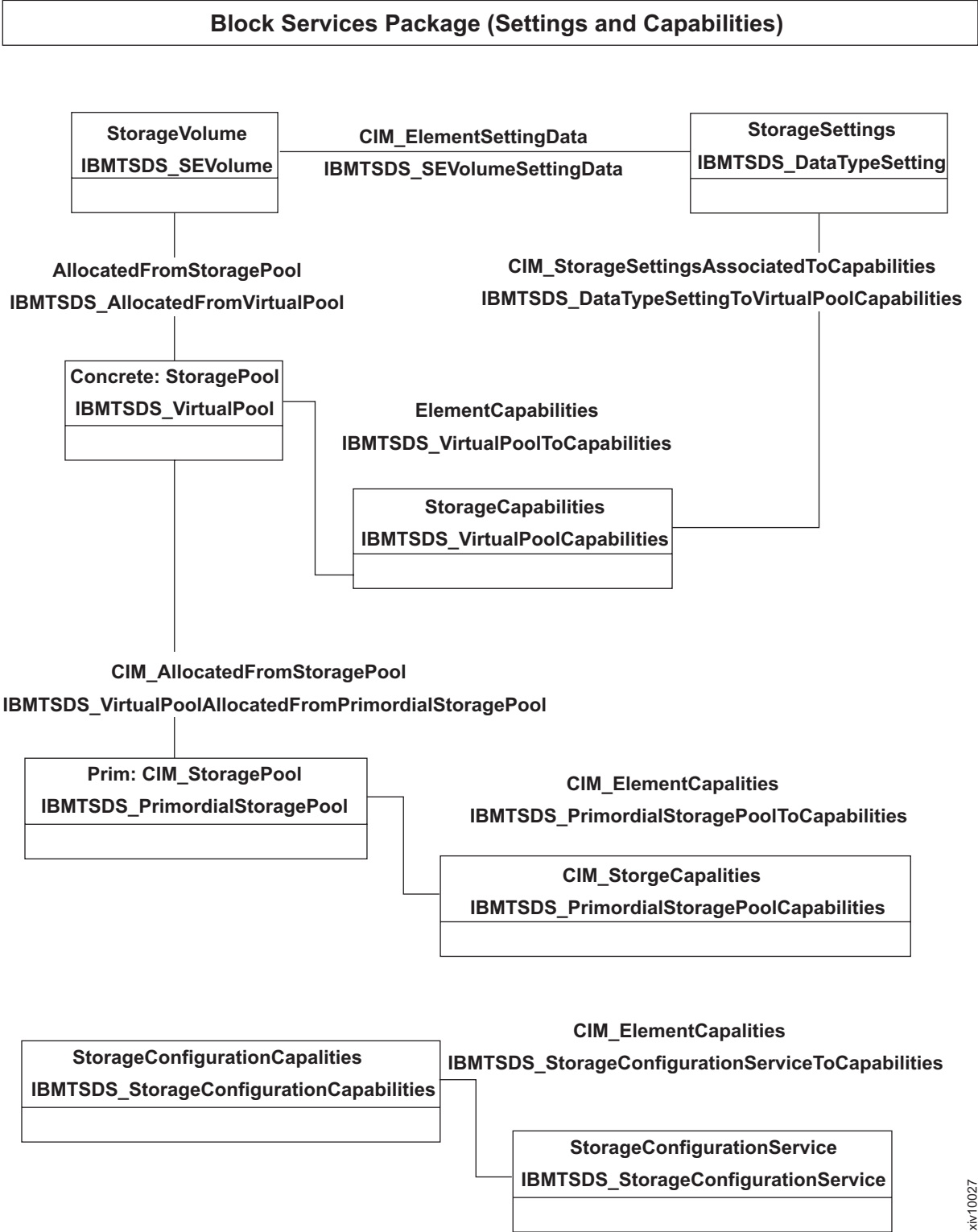


Figure 4. Block Services SMI-S model for XIV systems

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. See Figure 5 on page 34 for the Block Services package with settings and capabilities model.



xiv10027

Figure 5. Block Services Package with Settings and Capabilities model

A **StorageSetting** is a class that contains properties to specify the quality of service, such as **DataRedundancy** or parity layout. When you create 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. The system allocates 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. The volume's capacity is indicated as a discrete number of 512-byte blocks. The system allocates the soft volume size that is used within the **Storage Pool** as the minimum number of discrete 17 GB increments to meet the requested size. However, the size that is reported to hosts is equivalent to the precise number of blocks defined.

For more information about the block services object model, see the `BlockServices.mof` file.

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 initiated synchronously or asynchronously. XIV CIM Agent supports the following synchronous actions in **IBMTSDS_StorageConfigurationService** class:

Table 22. Synchronous actions

Synchronous actions	XIV CIM functions
StoragePool Creation	CreateOrModifyStoragePool
StoragePool Modification	CreateOrModifyStoragePool
StoragePool Deletion	DeleteStoragePool
StorageVolume Creation	CreateOrModifyElementFromStoragePool CreateOrModifyElementsFromStoragePool
StorageVolume Modification	CreateOrModifyElementFromStoragePool CreateOrModifyElementsFromStoragePool
StorageVolume Expansion	CreateOrModifyElementFromStoragePool CreateOrModifyElementsFromStoragePool
StorageVolume Shrinking	CreateOrModifyElementFromStoragePool CreateOrModifyElementsFromStoragePool
StorageVolume Deletion	Return to StoragePool
StorageVolume Deletion	Return Elements to StoragePool

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, specifies whether you want to create or modify a pool. If you specify a reference to a pool, it indicates that you want to modify the pool. If the parameter is null, it indicates that you want to create pool.

InPools

InPools specifies from which pool to create a pool. To create a **VirtualPool**, only the object reference of **IBMTSDS_PrimordialStoragePool** is allowed. All **VirtualPools** are created in the **Primordial StoragePool**. To create a **SnapshotPool**, only the object reference of **IBMTSDS_VirtualPool** is allowed 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 **IBMTSDS_VirtualPoolSetting** is allowed; to create a **SnapshotPool**, only the object reference **IBMTSDS_SnapshotPoolSetting** is allowed. The **Goal** parameter also specifies the requested hard size of the pool. See the BlockServer.mof file for details.

ElementName

The **ElementName** property provides a means for you to set a meaningful name for the pool to be created or modified. If specified, limit it to 63 characters which can include letters, digits, blanks, -, _ . and ~ characters. And blanks cannot be the beginning and ending characters. If not specified during pool creation, a random pool name is generated in the form of pool<random integer>.

Note: The name of the Pool must be unique in the system. 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, **Size** specifies the requested soft size of the pool. Null or 0 is not allowed for pool creation. 0 for pool modification means that the pool's size is not changed. As an output parameter, Size specifies the soft size achieved. If it is not possible to create a pool of the wanted size, a size is returned with size value set to the nearest supported size.

IBMTSDS_StorageConfigurationService. DeleteStoragePool

DeleteStoragePool 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 you delete 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 can 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 null when you modify **StorageVolumes**. As output, represents the number of volumes that are created if successful. Or creation was not successful because 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. The length must be equal to **Quantity** when you create volumes. If not NULL, the length must be the same as the length of **TheElements** when you modify volumes. For example, to create two volumes 'testVolume_3' and 'testVolume_4', 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 have the same format [LeadingString]_[IncrementingNumber]. [LeadingString] is the value of this parameter, and the first [IncrementingNumber] is the value of the **FirstSuffix** parameter. The incremental number is 1. 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 you use the second way described to create volumes or modify volume names. Users can supply the starting [IncrementingNumber] with this parameter. If it is null, the default value is 1. When you use the first way, it must be null.

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

ReturnCodes

Each volume that is created or modified can have a different error that is 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 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 handles 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 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

An enumeration that indicates what type of element is being created or modified. Only volume values are supported. So the value of **ElementType** is **2 (StorageVolume)** or **5 (ThinlyProvisionedStorageVolume)**. See the `BlockServer.mof` file for details.

InPool (*required for creation and moving*)

A reference to the **StoragePool** which the volume is to be created in or moved to. It must be an instance of **IBMTSDS_VirtualPool**.

Goal

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

Size

In creating or modifying a volume, this parameter specifies the wanted 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, set the **ForceShrink** parameter as true.

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 null or set to false, shrinking a volume fails. To shrink a volume, set this parameter to true. The default value is false.

Attention: Shrinking a volume can cause data loss.

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

Locked

In modifying a volume/snapshot, this parameter specifies the volume/snapshot locking status to be locked or unlocked. If this parameter is null, the volume/snapshot locking status is not modified. If this parameter is set to 0, the volume/snapshot is unlocked. If this parameter is set to 1, the volume/snapshot is locked. Other values are invalid. When a volume/snapshot is locked, and other properties are modified in the same method, the volume/snapshot is locked in the last step. When a volume/snapshot is unlocked and other properties are modified in the same method, the volume/snapshot is 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 descriptive name to assign to the volume.

Note: The volume name must be unique in the system. 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 that is 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 errors that can be generated for each individual volume that is 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. The **Goal** input parameter represents the pool data type to use to calculate the possible valid sizes. 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 is 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. The **Goal** input parameter represents the pool data type used to calculate the possible valid sizes. 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 is 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 that you can use the XIV Open API to configure XIV devices.

Procedure

1. Call **IBMTSDS_PrimitiveStoragePool.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. Use `invokeMethod` on **IBMTSDS_StorageConfigurationService.CreateOrModifyStoragePool** and specify the **ElementName**, **Size**, and **Goal** input parameters.
3. Save the pool output parameter. It is the reference to the **VirtualPool** that was 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 that you can use the XIV Open API to configure XIV devices.

Procedure

1. Use **associatorNames** from the **Concrete StoragePool** that was created to **CIM_StorageCapabilities** to get matching instance of **IBMTSDS_VirtualPoolCapabilities**.
2. Use **associatorNames** from **IBMTSDS_VirtualPoolCapabilities** to **CIM_StorageSetting** to get valid instances of **IBMTSDS_DataTypeSetting** and allow user to pick the data type to use to create the volume.

3. Call **IBMTSDS_VirtualPool.GetSupportedSizeRange** on the virtual pool that is 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. Use **InvokeMethod** on **IBMTSDS_StorageConfigurationService.CreateOrModifyElementFromStoragePool**. Use **IBMTSDS_VirtualPool** created in Step 1, **IBMTSDS_DataTypeSetting**, **Size**, and **ElementName** specified as the **InPools**, **Size**, **Goal**, and **ElementName** input parameters.
5. Save the **TheElement** output parameter and display it to the user, informing the user know that 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.

A client can obtain information about iSCSI capabilities and settings of an XIV storage array. A client can also obtain information that is provided by iSCSI-capable arrays and the storage volumes that are exposed by those arrays on an iSCSI network:

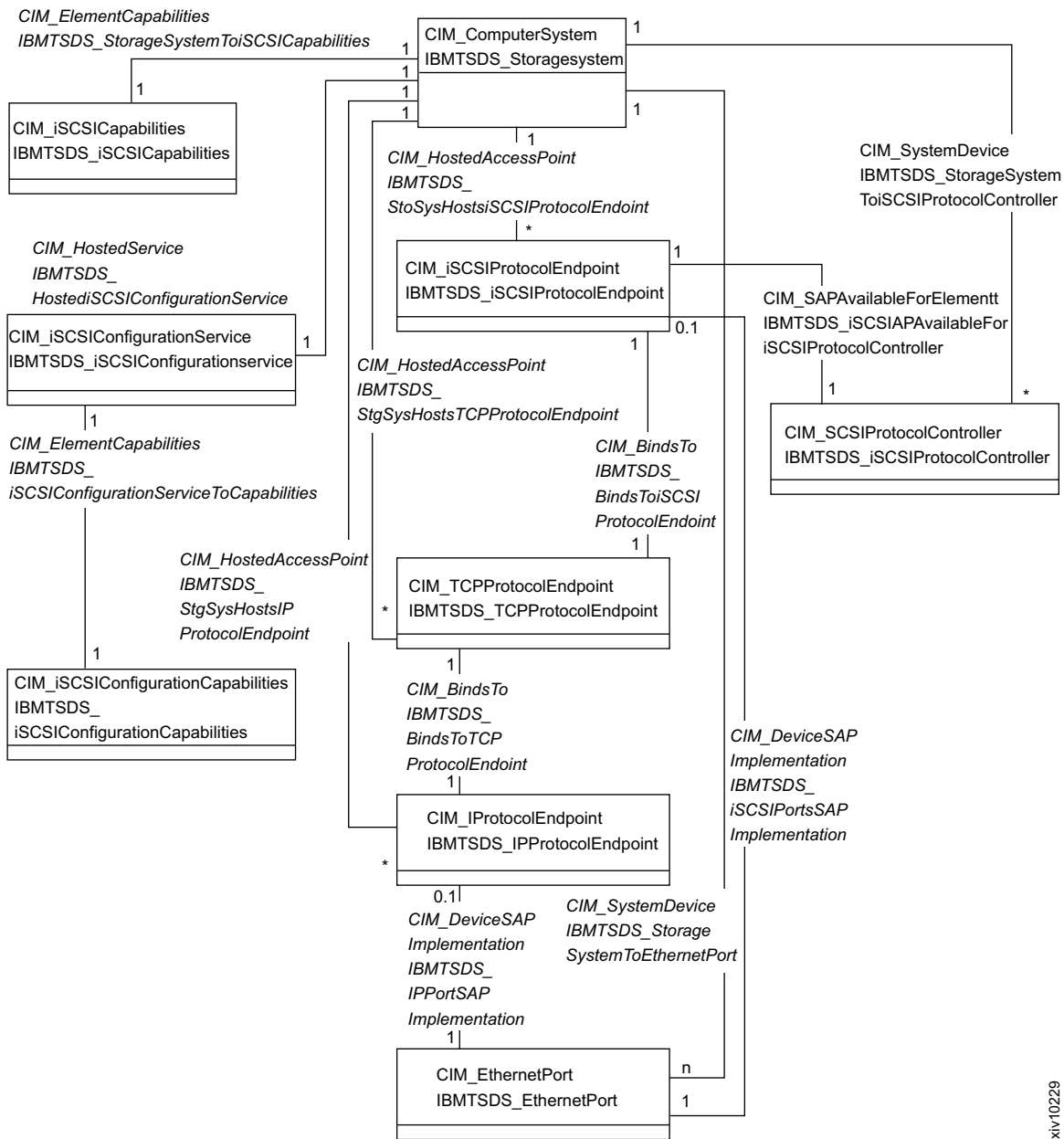
- Nodes
- Network Portals
- SCSI Ports

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.

See Figure 6 on page 42 for the iSCSI Target Ports SMI-S model for XIV systems.



xiv10229

Figure 6. iSCSI Target Ports SMI-S model for XIV systems

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

Table 23. iSCSI terminology and CIM class names

iSCSI term	CIM class name	Description
Network entity	ComputerSystem	The Network Entity represents a device that is accessible from the IP network. A Network Entity has one or more Network Portals. Each Network Portal can be used to gain access to the IP network by some iSCSI Nodes that are contained in that Network Entity.
Node	SCSIProtocolController	The iSCSI Node represents a single iSCSI Target. There are one or more iSCSI Nodes within a Network Entity. The iSCSI Node is accessible through one or more Network Portals. An iSCSI Node is identified by its iSCSI Name. Separating the iSCSI Name from the addresses allows multiple iSCSI nodes to use the same address, and the same iSCSI node to use multiple addresses.
SCSI port	iSCSIProtocolEndpoint	A SCSI Port with an iSCSI service delivery subsystem. A collection of Network Portals that together acts as a SCSI Target or target.
Network Portal	TCPProtocolEndpoint, IPProtocolEndpoint, EthernetPort	A component of a Network Entity that has a TCP/IP network address. It might be used by an iSCSI Node within that Network Entity for the connections within one of its iSCSI sessions. A Network Portal in a Target is identified by its IP address.

Network entities

Each **IBMTSDS_StorageSystem** is considered a separate Network Entity, which instance associated to an instance of **CIM_iSCSICapabilities** through 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 that are 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. The **Name** and **NameFormat** properties can be used to determine whether a **CIM_SCSIProtocolController** is an **iSCSI Node**. All

CIM_SCSIProtocolController instances that represent **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 **IBMTSDS_iSCSIProtocolController**. It is a subclass of **CIM_SCSIProtocolController**, which is associated to the first-level system (**IBMTSDS_StorageSystem**) through **CIM_SystemDevice**. For XIV arrays, each **IBMTSDS_StorageSystem** 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 **IBMTSDS_iSCSIProtocolController** instances are shown bolded in the following example.

```
instance of IBMTSDS_iSCSIProtocolController { SystemCreationClassName =
"IBMTSDS_StorageSystem"; SystemName = "IBM.2810-6000095";
CreationClassName = "IBMTSDS_iSCSIProtocolController"; Name =
"iqn.2005-10.com.xivstorage:000095.M_7_1"; NameFormat = 3; ElementName =
"M_7_1"; DeviceID = "IBM.2810-6000095-100955"; Caption =
"iSCSI Node of IP Interface M_7_1"; HealthState = 0; };
```

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 iSCSI nodes, then following the **CIM_SAPAvailableForElement** association from iSCSI Node (**IBMTSDS_iSCSIProtocolController**) to **CIM_iSCSIProtocolEndpoint**. 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 by using the **CIM_TCPProtocolEndpoint**, **CIM_IPProtocolEndpoint** and **CIM_EthernetPort** classes, which are associated through the **CIM_BindsTo** association:

- **CIM_IPProtocolEndpoint** contains the IP address of the Network Portal in the **IPv4Address** property. 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 an IP endpoint.

ElementName (*required*)

A meaningful name for the IP endpoint. It cannot be empty or null.

ModuleNumber (*required*)

An identifier that indicates the module that contains the IP endpoint. It cannot be 0 or 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 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 null, default value 4500 is 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 meaningful name for the IP endpoint.

ModuleNumber
An identifier that indicates the new module that contains 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 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.

IBMTSDS_iSCSIConfigurationService. DeleteIPProtocolEndpoint

This method is used to delete an IP **PortocolEndPoint**. The only input parameter that is 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 that you can use the XIV Open API to configure XIV devices.

Procedure

1. Obtain the reference (**CIMObjectPath**) of an **IBMTSDS_iSCSIConfigurationService** instance. The instance is associated with the **IBMTSDS_StorageSystem** in which you receive the performance statistics by traversing the **IBMTSDS_HostedIscsiConfigurationService** association.
2. Invoke the **IBMTSDS_iSCSIConfigurationService.CreateIPProtocolEndpoint** method to create an IP endpoint with specified values of parameters **ElementName**, **ModuleNumber**, **PortNumber**, **IPv4Address**, **IPv4Gateway**, **IPv4SubnetMask**, and **MTU**.
3. After it is successfully completed, check the 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 that 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 receive the performance statistics by traversing the **IBMTSDS_HostedIscsiConfigurationService** 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**, and **IPEndpoint**.
4. After it is successfully completed, 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 that you can use the XIV Open API to configure XIV devices.

Procedure

1. Obtain the reference (**CIMObjectPath**) of an **IBMTSDS_iSCSIConfigurationService** instance. Ensure that the reference is associated with the **IBMTSDS_StorageSystem** in which you receive the performance statistics by traversing the **IBMTSDS_HostedIscsiConfigurationService** 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 completed, 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 function.

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 by 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 on page 48 provides an example of how the XIV system uses masking and mapping.

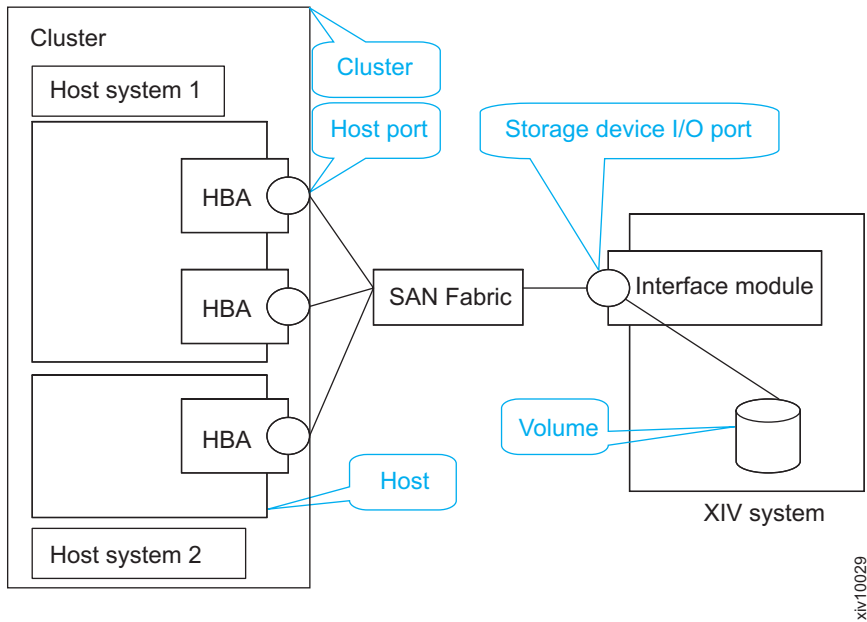


Figure 7. Masking and mapping physical model in the XIV system

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

Storage device I/O ports

Ports on the XIV system that represent a Fibre Channel port or an iSCSI port that is used for host I/O operations. Storage device I/O ports are sometimes referred to 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 Fibre 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 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.

Map hosts to a volume

You can map a host to volumes. A host can contain several initiator ports that include Fibre 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.

Map clusters to a volume

You can map a cluster to a volume when several hosts must 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 is 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. Target ports cannot be selected when you configure host mappings, but can be configured by zoning configurations on the switch. The target ports that are involved in all views can be displayed by enumerating the instances of `CIM_ProtocolControllerForPort` class. See Figure 8.

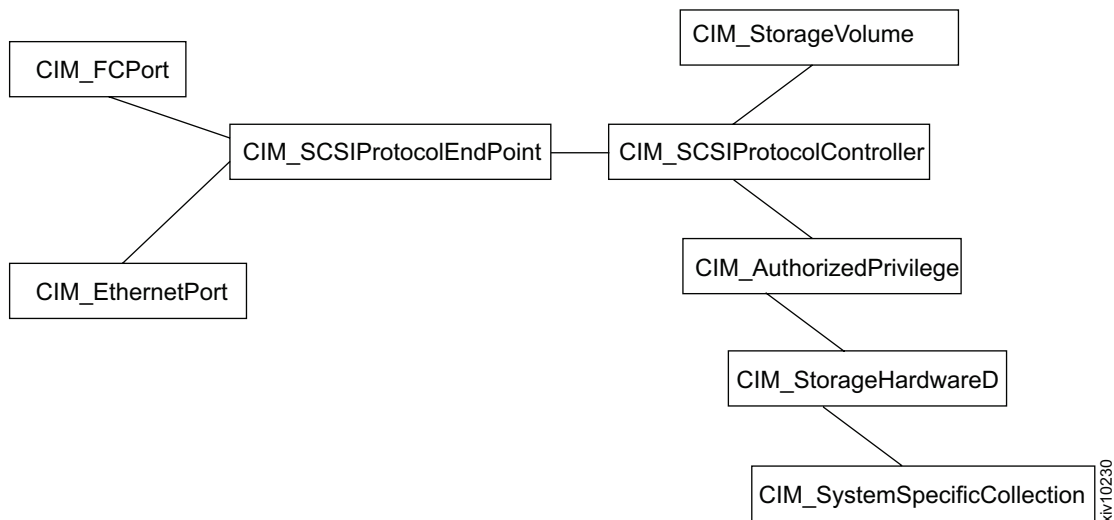


Figure 8. Masking and mapping object model in SMI-S

Table 24 on page 50 describes the masking and mapping classes that are available. For more information about the masking and mapping object model, see the `MaskingMapping.mof` file. The MOF documentation is in the `mof` folder in the CIM agent installation directory.

Table 24. Masking and mapping classes

CIM class name	Description	XIV CIM class name
CIM_FCPort	A Fibre Channel port on the XIV system.	IBMTSDS_FCPort
CIM_EthernetPort	An Ethernet port on the XIV system.	IBMTSDS_EthernetPort
CIM_SCSIProtocolEndPoint	The Fibre Channel or iSCSI port that is used in the mapping	IBMTSDS_SCSIProtocolEndpoint IBMTSDS_iSCSIProtocolEndpoint
CIM_SCSIProtocolController	A logical entity that represents a host and the relationship to the volumes	IBMTSDS_SCSIProtocolController
CIM_StorageVolume	A storage volume on the XIV system	IBMTSDS_SEVolume
CIM_AuthorizedPrivilege	A logical entity that represents the access permissions for a set of volumes	IBMTSDS_Privilege
CIM_StorageHardwareID	An initiator port on the host system	IBMTSDS_StorageHardwareID
CIM_SystemSpecificCollection	A logical entity which represents a collection of StorageHardwareID (host)	IBMTSDS_SystemSpecificCollection
CIM_Cluster	A logical entity which represents a collection of SystemSpecificCollection (hosts)	IBMTSDS_Cluster

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

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 in to 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, and create, modify, and delete **StorageHardwareIDs**.

- **CreateStorageHardwareID()** method creates **StorageHardwareID**. On XIV, a host is created which contains the specified initiator port.
- **DeleteStorageHardwareID()** deletes a **StorageHardwareID**. On XIV, the initiator port is removed from the host that contains it.

HardwareIDCollection manipulation

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

- **CreateHardwareIDCollection** method creates a **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 supports read/write access. A privilege can be located by using the **PrivilegeManagementService**.

IBMTSDS_StorageHardwareIDManagement. CreateStorageHardwareID

You can use the **CreateStorageHardwareID** method to create a **StorageHardwareID**. This method creates a host in the XIV device that contains the initiator port that is specified by **StorageID**.

Setting

An input parameter that represents the operating system of the port. It must be an instance of **IBMTSDS_StorageClientSettingData**. Each instance represents a different operating system. Only **Standard**, **HPUX**, and **z/VM** are supported by XIV CIM Agent. If it is null, **Standard** is used by default.

IDType

An input parameter that represents the type of port. To create a **HardwareID** representing a Fibre 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 that represents a reference to the **StorageHardwareID** to delete.

IBMTSDS_StorageHardwareIDManagement. CreateHardwareIDCollection

You can use the **CreateHardwareIDCollection** method to create **HardwareIDCollections**.

ElementName *(required)*

An input parameter that represents a meaningful name for the **HardwareIDCollection** that is being created. The **ElementName** must be unique in the system. It cannot be a name that is already assigned to one of the other **HardwareIDCollections** in the system. If not specified, CIM generates a random name.

HardwareIDs

An input parameter that represents the ID of the ports that are contained by the **HardwareIDCollection**. If specified, each ID must be the WWN of a Fibre Channel port or the IQN of an iSCSI port. If not specified, an empty **HardwareIDCollection** is created.

HardwareIDCollection

An output parameter that represents 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 fails if the port already belongs to another host. The operation succeeds if the port already belongs to the specified host, but it has no effect.

HardwareIDs *(required)*

An input parameter that represents an array of strings that contain **StorageIDs** of **StorageHardwareID** instances that become members of the **HardwareIDCollection**.

HardwareIDCollection *(required)*

An input parameter that represents the reference to the **SystemSpecificCollection** into which the **HardwareIDs** are added.

IBMTSDS_ControllerConfigurationService. ExposePaths

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

ExposePaths initiates the mapping and masking operation in one method call. It produces 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

Parameters/ Use cases	LUNames	Initiator PortIDs	TargetPortIDs	DeviceNumbers	DeviceAccesses	ProtocolControllers (on input)
Create a view	Optional	Optional	NULL	Optional	Optional	NULL
Add LUNs to a view	Mandatory		NULL	Optional	Optional	contains a single SPC reference
Add initiator IDs to a view	NULL	Mandatory	NULL	NULL	NULL	contains a single SPC reference

LUNames *(required)*

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

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

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

DeviceNumbers *(required)*

A string array input parameter that represents the LUN IDs. Each item in this array must be a number 0 - 511. The number of items is the same as the number of items that are specified by the **LUNames**. If not specified, the CIM Agent assigns unused LUN IDs for the operation.

DeviceAccesses

A unit16 array input parameter that represents the access rights to give to the **StorageHardwareID** to the volumes specified in **LUNames**. Read/write access is supported. It 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, it represents the **SCSIProtocolController** to be modified. As an output parameter, it represents the **SCSIProtocolController** that is being modified or created.

IBMTSDS_ControllerConfigurationService. 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

Parameters/ Use cases	LUNames	InitiatorPortIDs	TargetPortIDs	ProtocolControllers (on input)
Remove LUs from a view	Mandatory	NULL	NULL	contains a single SPC reference
Remove initiator IDs from a view	NULL	Mandatory	NULL	contains a single SPC reference
Hide full paths from a view	Mandatory	Mandatory	NULL	contains a single SPC reference

LUNames *(required for removing access to volumes)*

A string array input parameter that represents the volumes to unmap from the

StorageHardwareID. They are not references to volume instances, but are strings that match the name property of the **IBMTSDS_SEVolume (CIM_StorageVolume)**.

InitiatorPortIDs (*required for removing access to initiator ports*)

A string array input parameter that represents the initiator port that you want to remove from the **SCSIProtocolController** view.

ProtocolControllers (*required*)

As an input parameter, it represents the **SCSIProtocolController** to be modified. As an output parameter, it represents the **SCSIProtocolController** that was modified or deleted.

IBMTSDS_ControllerConfigurationService. DeleteProtocolController

The **DeleteProtocolController** method deletes an existing **ProtocolController**. In XIV, the corresponding host and all the host ports that are contained by the host are deleted.

ProtocolController (*required*)

An input parameter that represents the **ProtocolController** to be deleted.

DeleteUnits

An input parameter that represents the requirement to delete the volumes that are mapped to the **ProtocolController**. If true is specified, the mapped volumes are deleted.

Sample recipe

The following task is a sample recipe for configuring an XIV device. It 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. Use **InvokeMethod** on **IBMTSDS_ControllerConfigurationService.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 that matches the selected host.
 - c. Use **InvokeMethod** on **IBMTSDS_ControllerConfigurationService.ExposePaths** with **LUNames** set to volume names, **DeviceAccesses** set to 2, and **ProtocolControllers** set to the **SCSIProtocolController**.
3. Add initiator ports to an existing SPC:
 - a. Attain IDs of initiator ports.
 - b. Find the **IBMTSDS_SCSIProtocolController** instance.
 - c. Use **InvokeMethod** on **IBMTSDS_ControllerConfigurationService.ExposePaths** with **InitiatorPortIDs** set to the initiator 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 that matches the selected host.
 - c. Use **InvokeMethod** on **IBMTSDS_ControllerConfigurationService.HidePaths** with **LUNames** set to volume names, and **ProtocolControllers** set to the **SCSIProtocolController**.
5. Remove initiator ports from an SPC:
 - a. Attain the SPC reference and IDs of the initiator port.
 - b. Find the **IBMTSDS_SCSIProtocolController** instance.
 - c. Use **InvokeMethod** on **IBMTSDS_ControllerConfigurationService.HidePaths** with **InitiatorPortIDs** set to the IDs of the initiator ports, and **ProtocolControllers** set to the **SCSIProtocolController**.

Indication profile

SMI-S supports two types of indications: Lifecycle Indications and Alert Indications. Lifecycle 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. Lifecycle Indications are implemented in all XIV CIM Agent releases. Beginning with XIV CIM Agent release 11.0, **AlertIndications** are implemented.

Supported lifecycle indications by class

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

Table 27. Indication types and object classes

Class name	Supported lifecycle indications
IBMTSDS_SEVolume	IBMTSDS_InstCreation IBMTSDS_InstDeletion IBMTSDS_InstModification
IBMTSDS_VirtualPool	IBMTSDS_InstCreation IBMTSDS_InstDeletion IBMTSDS_InstModification
IBMTSDS_ProtocolControllerForSEUnit	IBMTSDS_InstCreation IBMTSDS_InstDeletion
IBMTSDS_SCSIProtocolController	IBMTSDS_InstCreation IBMTSDS_InstDeletion
IBMTSDS_SystemSpecificCollection	IBMTSDS_InstCreation IBMTSDS_InstDeletion
IBMTSDS_StorageHardwareID	IBMTSDS_InstCreation IBMTSDS_InstDeletion
IBMTSDS_iSCSIProtocolEndpoint	IBMTSDS_InstCreation IBMTSDS_InstDeletion

The lifecycle indications are sent following an XIV CIM Agent configuration function is successfully completed. If SMI-S client starts the XIV CIM Agent **ExposePaths** method to map a volume to a host, a related **InstCreation** indication with the new **IBMTSDS_ProtocolControllerForSEUnit** instance is sent from the provider.

In addition, XIV CIM Agent periodically fetches XIV events and sends related lifecycle indications. If a **StorageVolume** is created on an XIV array through the XIV GUI, an SMI-S client receives a related **InstCreation** indication with the new **IBMTSDS_SEVolume** instance from the provider. By default, the XIV event fetch 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 `yyyymmddhhmmss.mmmmmmsutc`, where:

- `yyyy` is the four-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 has values other than 000 (such as 540). Values other than 000 are 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 can be initiated on elements from the same storage system or across a connection to a different storage system. Elements that are used in a copy operation can be grouped to facilitate the copy operation on many elements at the same time. Furthermore, the elements of a group can be declared as Consistent.

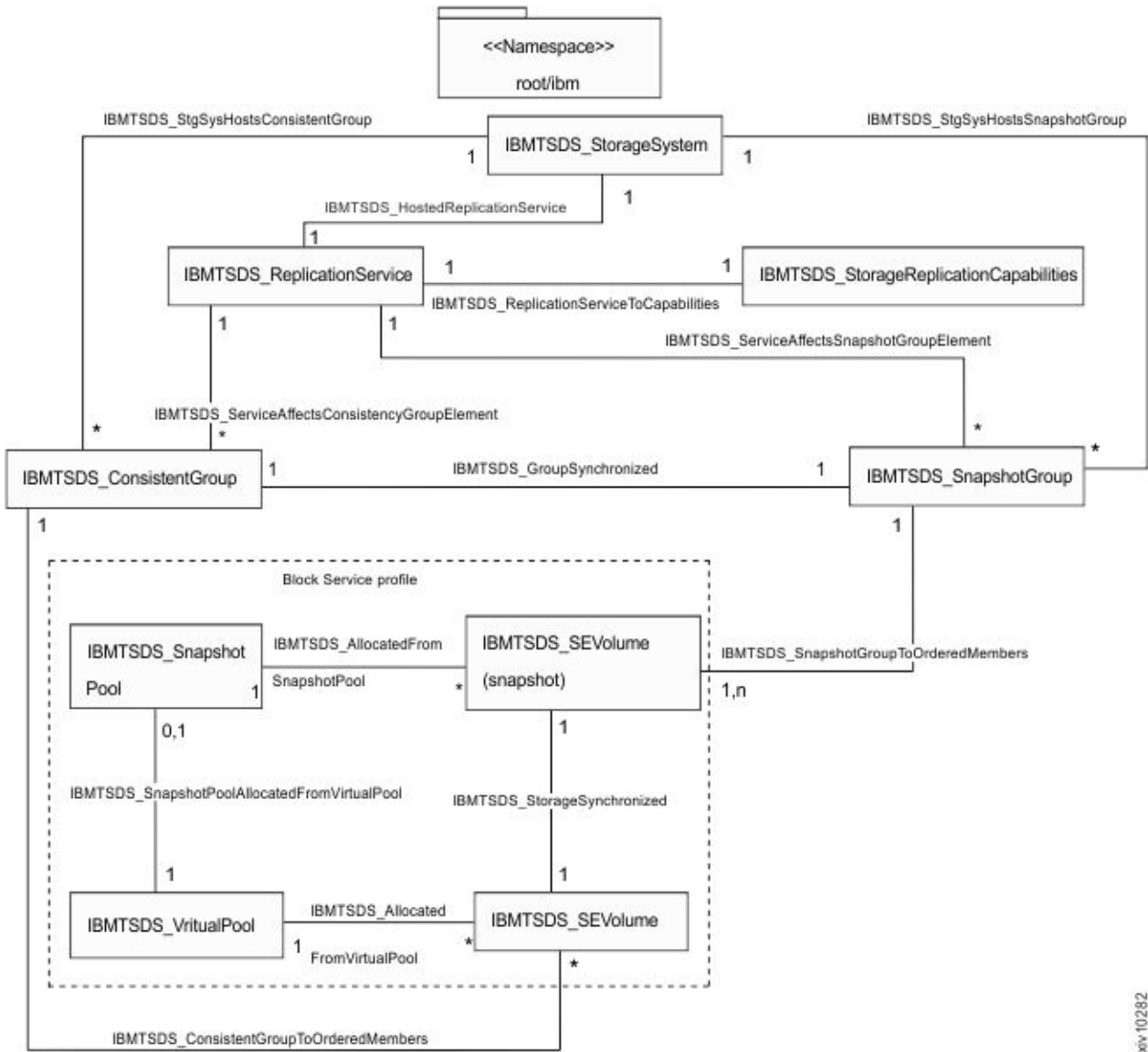
Two types of synchronization views are supported. A target element can be synchronized to the current view of the source element, or it can be synchronized to a point-in-time view. Snapshots and clones represent a point-in-time view, while a mirror represents a current view. Two copy modes are supported: synchronous and asynchronous. In synchronous mode, the writer waits for acknowledgement that a write to the source element was processed by the target element. Until it receives this acknowledgement, it does not accept 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 later.

The Replication Services profile supports local and remote replication. Local replication specifies that both the source elements and the target elements are contained in a single managed system, such as an array platform. Remote replication specifies that the source elements and the target elements are contained in separate systems. For remote replication, the client may interact with both the source system and the target system, but the client only invokes the replication methods to a single replication service.

Replication Services object model

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

See Figure 9 on page 58 and Figure 10 on page 59.



xiv10282

Figure 9. Replication Services (Local) SMI-S model for XIV systems

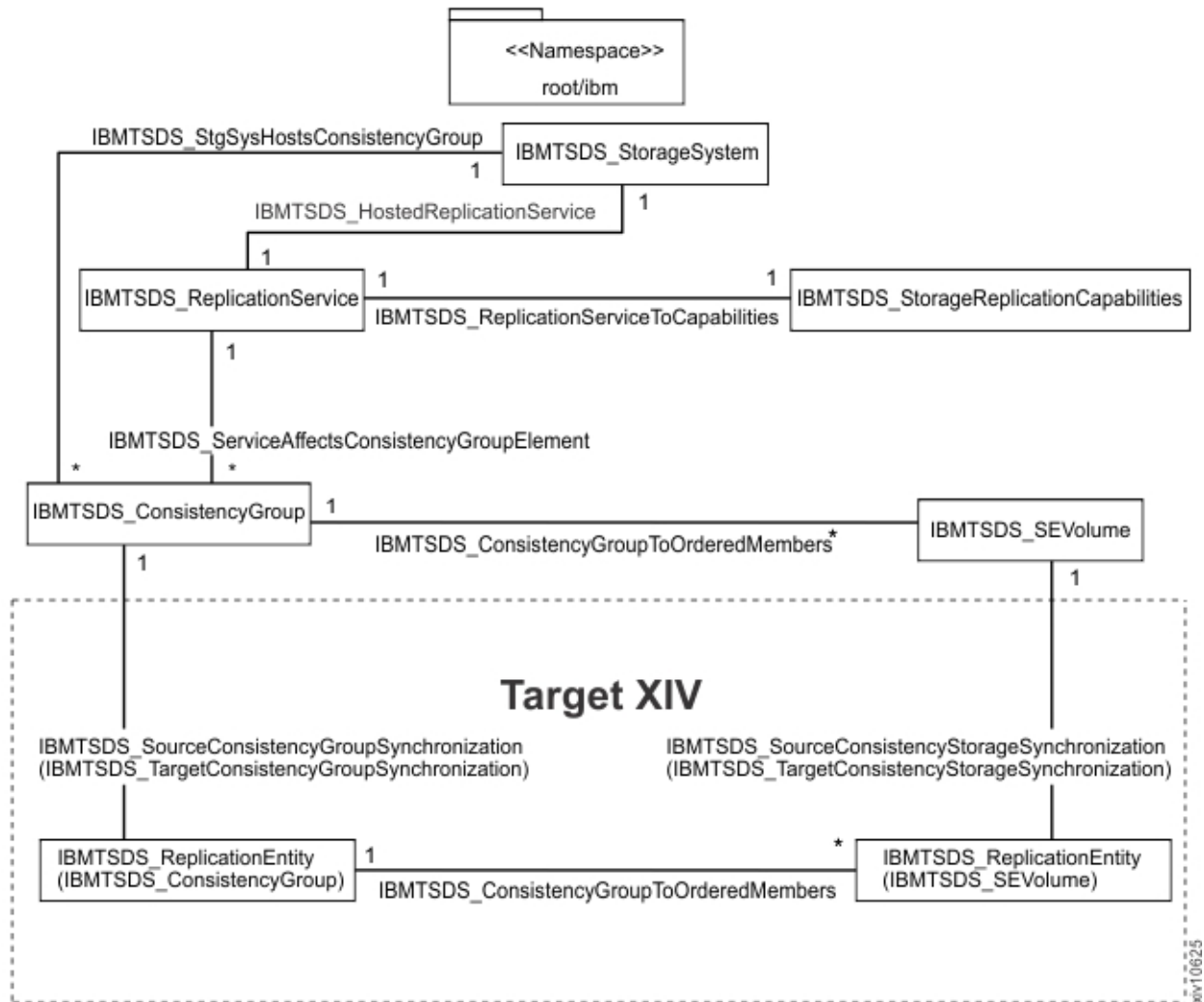


Figure 10. Replication Services (Remote) SMI-S model for XIV systems

Replication Service capabilities

Use XIV terminology to map to SMI-specific terminology:

Table 28. Mapping XIV terminology to SMI terminology

XIV Terminology	SMI Terminology	Description	XIV CIM Agent support
Snapshot	Asynchronous Snapshot Local	A "Point-in-Time", associated virtual copy of the source element. The target element enables visibility into a session.	version 11.1 or later
Volume copy	Synchronous Clone Local	A full, "Point-In-Time", unassociated local copy of the source element.	version 11.1 or later
Synchronous Remote Mirroring	Synchronous Mirror Remote	A synchronized remote copy of the source element.	version 11.6 or later
Asynchronous Remote Mirroring	Asynchronous Mirror Remote	An asynchronous remote copy of the source element.	version 11.6 or later

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

Method	Description
ConvertSyncTypeToReplicationType	Translates CopyType, Mode, and Local/Remote to the corresponding ReplicationType
ConvertReplicationTypeToSyncType	Translates ReplicationType to the corresponding CopyType, Mode, and Local/Remote
GetSupportedFeatures	Determines supported features
GetSupportedGroupFeatures	Determines supported group features
GetSupportedCopyStates	Determines the supported copy states
GetSupportedGroupCopyStates	Determines supported group copy states
GetSupportedWaitForCopyStates	Determines supported wait for copy states
GetSupportedConsistency	Determines supported consistency
GetSupportedOperations	Determines supported operations
GetSupportedGroupOperations	Determines supported group operations
GetSupportedListOperations	Determines supported list operations
GetSupportedMaximum	Determines supported maximum
GetDefaultConsistency	Determines default consistency
GetDefaultGroupPersistency	Determines default group persistency
GetSynchronizationSupported	For the supplied element, this method returns the supported synchronization operations in a series of parallel output arrays.
GetSupportedOperationsForSynchronization	For the supplied synchronized association, this method returns the supported operations in a series of parallel output arrays.

Group management

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

Table 30. Group management classes

XIV	SMI class	XIV CIM class
Consistency group	CIM_ReplicationGroup	IBMTSDS_ConsistencyGroup
Snapshot group	CIM_ReplicationGroup	IBMTSDS_SnapshotGroup
Association between Consistency group and volumes in it	CIM_OrderedMemberOfCollection	IBMTSDS_ConsistencyGroupToOrderedMembers
Association between Snapshot group and snapshots in it	CIM_OrderedMemberOfCollection	IBMTSDS_SnapshotGroupToOrderedMembers

See Figure 11 on page 61.

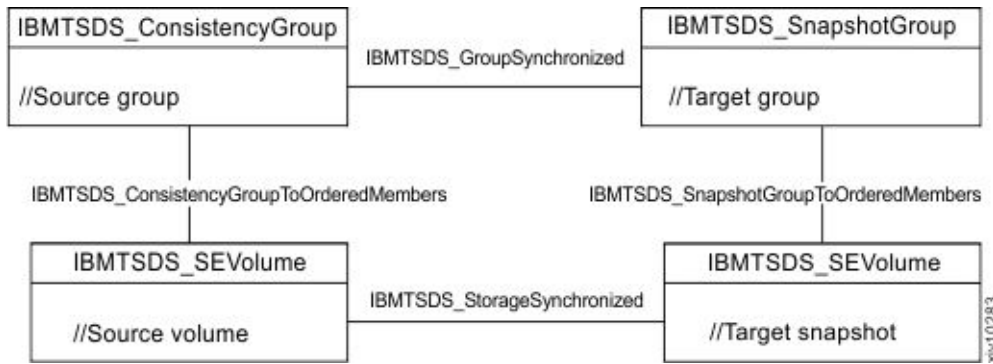


Figure 11. Sample local group information retrieval

Table 31. Extrinsic methods for group management

Method	Description
CreateGroup	Creates a consistency group
DeleteGroup	Deletes an existing consistency group
AddMembers	Adds volumes to a consistency group
RemoveMembers	Removes volumes from a consistency group

IBMTSDS_ReplicationService. CreateGroup

The **CreateGroup** method creates a consistency group.

GroupName

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

Members

As an input, refers the members to be added to the group that is created. If not specified, an empty group is 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 are removed before you delete 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 adds volumes to a consistency group.

ReplicationGroup

As an input, refers to the consistency group to which members are 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 are 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 is deleted if all members are removed.

Replication management

The CIM agent version 11.1 and later supports local replication manipulation, including local snapshot and clone. The CIM agent version 11.6 and later also supports remote replication manipulation, including volume and consistency group mirroring.

See the following tables and figures for information about replication management:

Table 32. Replication management classes

XIV	SMI class	XIV CIM class
Volume, snapshot	CIM_StorageVolume	IBMTSDS_SEVolume
Remote volume / consistency group	CIM_ReplicationEntity	IBMTSDS_ReplicationEntity
Association between volume and its snapshot	CIM_StorageSynchronized	IBMTSDS_StorageSynchronized
Association between consistency group and its snapshot group	CIM_GroupSynchronized	IBMTSDS_GroupSynchronized
Association between mirrored volumes	CIM_StorageSynchronized	IBMTSDS_SourceConsistencyStorageSynchronized / IBMTSDS_TargetConsistencyStorageSynchronized
Association between mirrored consistency groups	CIM_GroupSynchronized	IBMTSDS_SourceConsistencyGroupSynchronized / IBMTSDS_TargetConsistencyGroupSynchronized

See Figure 12 on page 63.

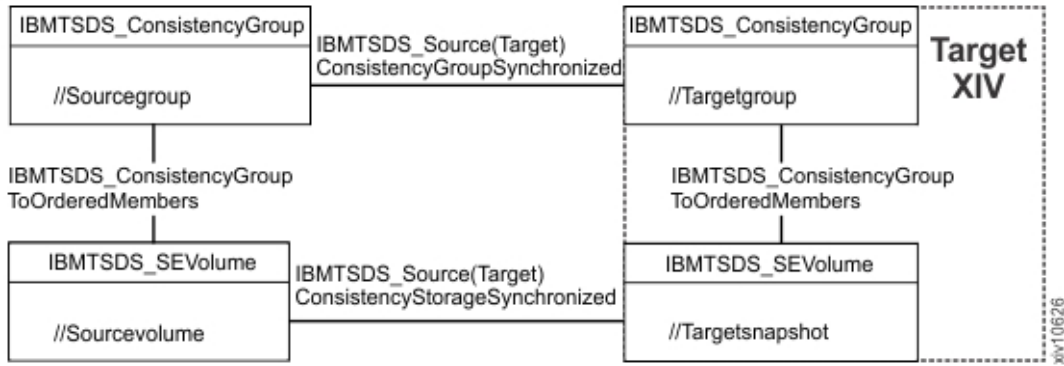


Figure 12. Sample remote group information retrieval

Note: The **IBMTSDS_SourceConsistencyStorageSynchronized** and **IBMTSDS_TargetConsistencyStorageSynchronized** classes are association classes which associate the source volume with the target volume. They are essentially the same association class, but differ in how they get data from Primary XIV and Secondary XIV respectively; see Figure 13. At the same time, the **IBMTSDS_SourceConsistencyGroupSynchronized** and **IBMTSDS_TargetConsistencyGroupSynchronized** associate the source consistency group with the target consistency group.

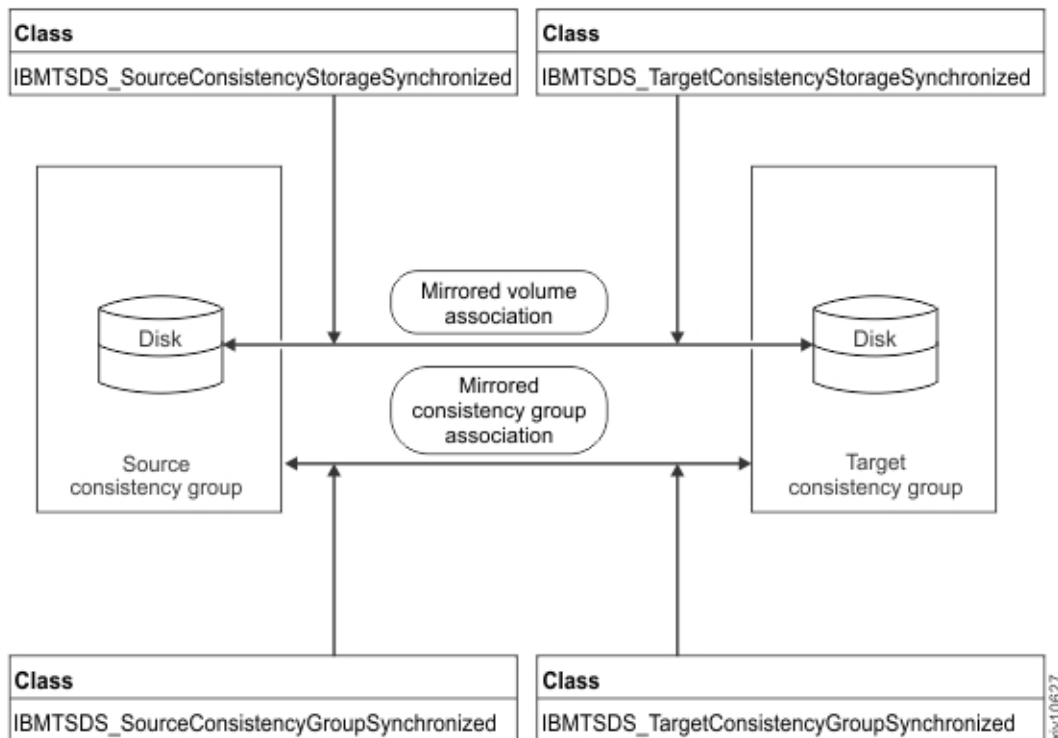


Figure 13. Association classes for mirrored volumes and consistency groups

Table 33. Extrinsic methods for replication management

Method	Description
CreateElementReplica	Create a snapshot of a volume, or a duplication snapshot of a snapshot, or a clone of a volume.
CreateGroupReplica	Create a snapshot group of a consistency group.
ModifyReplicaSynchronization	For local replication service, restore a snapshot or a snapshot group; or delete a snapshot or a snapshot group.
ModifyListSynchronization	Restore a list of snapshots or snapshot groups; or delete snapshots or snapshot groups.
GetAvailableTargetElements	Return snapshots of a specified volume or snapshot.
GetPeerSystems	Get all of the peer systems.
CreateGroupReplicaFromElements	Create consistency group and volumes mirror.
GetReplicationRelationships	Get all of available synchronization relationships.
GetReplicationRelationshipInstances	Get all of available synchronization relationship instances as strings of references.

IBMTSDS_ReplicationService. CreateElementReplica

The **CreateElementReplica** method creates a snapshot of a volume, duplicate snapshot of a snapshot, clone of a volume, and 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 is created if specified.

SourceElement

As an input, refers the source volume or snapshot for which the replica is created.

TargetElement

As an input, refers to a target element to use if specified. This parameter cannot be specified when you are 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 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. If not specified, CIM selects the first available pool on the device to create the clone.

WaitForCopyState

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

IBMTSDS_ReplicationService. CreateGroupReplica

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

RelationshipName

As an input, refers to the name of the snapshot group that is 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 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 is 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 that is 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 to the pool in which the replica is being created. The pool must be the same as the pool of the SourceGroup if specified.

WaitForCopyState

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

IBMTSDS_ReplicationService. ModifyReplicaSynchronization

For local replication service, 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. For remote replication service, use the **ModifyReplicaSynchronization** method to manipulate the mirror consistency group (for example, to split the mirrored consistency group or set up failover for the mirrored consistency group).

Operation

As an input, refers to the type of modification to be made to the replica. The following values/modifications are supported: 4 (Activate), 5 (AddSyncPair), 7 (Deactivate), 8 (Detach), 10 (Failover), 12 (Fracture), 13 (RemoveSyncPair), 14 (ReSync Replica), 15 (Restore from Replica), 19 (Return To ResourcePool) and 20 (Reverse Roles), and 21 (Split).

Synchronization

As an input, refers to the replica synchronization to be modified. For local, it must be an instance of **IBMTSDS_GroupSynchronized** or **IBMTSDS_StorageSynchronized** class. For remote, it must be an instance of **IBMTSDS_SourceConsistencyGroupSynchronized** or **IBMTSDS_SourceConsistencyStorageSynchronized**.

WaitForCopyState

As an input, refers to the copy state the replica must reach before the method

returns. The values 3, 4, 8, and 10 are supported, if specified. Different operations correspond with different **WaitForCopyState** values, as shown in Table 34.

Table 34. Operations, operation descriptions, and corresponding WaitForCopyState states

Operation	Operation Description (from profile)	Corresponding WaitForCopyState Value
Activate	Activate an Inactive or Prepared StorageSynchronized association.	Unsynchronized
Detach	Remove the association between the source and target elements. Does not delete the target element.	N/A
Failover	Enable the read and write operations from the host to the target element. This operation is useful for situations when the source element is unavailable.	Failedover
Fracture/Split	Separate the target element from the source element.	Inactive
Resync Replica	Resynchronize a fractured target element. Or, from a Broken or Aborted relationship. To continue from the Broken state, the problem should be corrected first before resyncing the replica. Also, to continue from the Aborted state.	Synchronized
Reverse Roles	Switch the source and the target element roles. The source element may need to be Read Only. See GetSupportedFeatures in capabilities.	Unsynchronized
Add/Remove SyncPair	Add/Remove mirrored volumes into/from mirrored consistency group.	Synchronized

Note: The **ReverseRoles** and **ResyncReplica** operations may take some time to execute, so a Job will be returned (4096).

IBMTSDS_ReplicationService. ModifyListSynchronization

This method is a batch operation of the **IBMTSDS_ReplicationService.ModifyReplicaSynchronization** method, which modifies a list of synchronizations.

IBMTSDS_ReplicationService. GetAvailableTargetElements

The **GetAvailableTargetElements** method retrieves 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 **IBMTSDS_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.

IBMTSDS_ReplicationService. GetReplicationRelationships

The **GetReplicationRelationships** method retrieves all of the snapshot or mirror relationships on a XIV device.

Type

As an input, refers to the replication type. To get snapshots and mirrored volumes, specify 2; to get snapshot groups, specify 3; if not specified, all snapshots and snapshot groups are returned.

SyncType

As an input, refers to type of target elements. To get snapshots and snapshot groups, specify 7. To get mirror relationships, specify 7; if no value is specified, all snapshots, snapshot groups, and mirror relationships are returned.

Mode

As an input, refers to mode of target elements. 2 and 3 are supported if specified.

Locality

As an input, refers to locality of target elements. To get mirrored volumes and consistency groups, specify 3. To get snapshots and snapshot groups, specify 2. If no value is specified, all snapshots, snapshot groups, and mirror relationships are returned.

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, snapshot groups, or mirror relationships.

IBMTSDS_ReplicationService. GetReplicationRelationshipInstances

This method is similar to **GetReplicationRelationships**, except that it returns strings of references of snapshots, snapshot groups, or mirror relationships.

IBMTSDS_ReplicationService. GetPeerSystems

Use this method to get (or start a job to get) all of the peer systems. A peer system is a system that is known and visible to the replication service. This method will return all the connected XIV Storage Systems.

IBMTSDS_ReplicationService. CreateGroupReplicaFromElements

Use this method to create (or start a job to create) new storage objects that are replicas of the specified source storage objects (**SourceElements**).

If 0 is returned, the function completed successfully, and no **ConcreteJob** instance is created. If 4096/0x1000 is returned, a **ConcreteJob** is started, and a reference to it is returned in the **Job** output parameter. This method combines the functionality of **CreateGroup** and **CreateGroupReplica**, in one call. This method creates mirrors for volumes and consistency groups and then adds the mirrored volumes into the mirrored consistency groups.

SyncType

As an input, refers to the type of target elements. Only 6 is supported.

Mode

As an input, refers to the mode of target elements. 2 and 3 are supported, if specified.

SourceGroup

As an input, refers to a source group. It must be an instance of **IBMTSDS_ConsistencyGroup** class.

SourceElements

As an input, refers to the source volumes. It must be an instance of **IBMTSDS_SEVolume** class.

SourceGroupName

As an input, refers to the name of the source group to be created. As an output, refers to the name of the source group that is being created.

TargetGroup

As an input, refers to a target group. It must be an instance of **IBMTSDS_ConsistencyGroup** class. If not specified, a target group will be created in the target pool, which is specified by the **TargetPool** parameter. **TargetGroup** and **TargetPool** can't be NULL at the same time.

TargetGroupName

As an input, refers to the name of the target group to be created.

Consistency

As an input, only 3 is supported, if specified.

TargetPool

As an input, refers to the target pool that puts the target elements (the replicas). It must be an instance of **IBMTSDS_VirtualPool**. If not specified, it will be gotten from the **TargetGroup**. If **TargetGroup** and **TargetPool** are specified at the same time, and **TargetGroup** is not in the **TargetPool**, **TargetGroup** will be moved to **TargetPool**.

WaitForCopyState

As an input, refers to the copy state the replica must reach before the method returns. Only 3 is supported, if specified.

Job

As an output, refers to the job.

Table 35. SMI Indications

SMI Indication	Description
IBMTSDS_StorageSynchronized InstCreation	After a snapshot is created or snapshot group is created
IBMTSDS_GroupSynchronized InstCreation	After a snapshot group is created
IBMTSDS_GroupSynchronized InstDeletion	After a snapshot group is deleted
IBMTSDS_StorageSynchronized InstDeletion	After a snapshot or snapshot group is deleted

Table 35. SMI Indications (continued)

SMI Indication	Description
IBMTSDS_ConsistencyGroupInstDeletion	After a consistency group is deleted
IBMTSDS_ConsistencyGroupInstCreation	After a consistency group is created
IBMTSDS_ConsistencyGroupInstModification	After a consistency group is renamed
IBMTSDS_ConsistencyGroupToOrderedMembersInstCreation	After volume is added/removed into/from consistency group
IBMTSDS_ConsistencyGroupToOrderedMembersInstDeletion	After all volumes are removed from the consistency group
IBMTSDS_SourceConsistencyGroupSynchronizedInstDeletion	After the mirror relationship is deleted
IBMTSDS_SourceConsistencyGroupSynchronizedInstCreation	After the mirror relationship is created
IBMTSDS_TargetConsistencyGroupSynchronizedInstCreation	After the mirror relationship is created
IBMTSDS_SourceConsistencyGroupSynchronizedInstModification	After the mirror relationship designation is changed
IBMTSDS_TargetConsistencyGroupSynchronizedInstModification	After the mirror relationship designation is changed

Job Control profile

In some profiles, like the replication services profile, some of the methods described may take some time to execute. In these cases, a mechanism is needed to handle asynchronous execution of the method as a 'Job.' The Job Control profile defines the constructs and behavior for job control.

When the Job Control profile is implemented, and a client executes a method that executes asynchronously, a reference to an instance of **ConcreteJob** is returned and the return value for the method is set to "Method parameters checked - job started" (4096). The **ConcreteJob** instance allows the progress of the method to be checked, and instance Indications can be used to subscribe for **Job** completion.

Although there are many methods, such as format volume and copy volume that are asynchronous jobs, in this work item, we only implement the job control for the volume/consistency group mirror.

In XIV CIM provider, the methods **CreateGroupReplicaFromElements** and **ModifyReplicaSynchronization** will output a job that will be associated to the elements whose references are created or modified as a side-effect of the job's execution via the **AffectJobElement** association.

The lifetime of a completed job instance, and thus the **AffectedJobElement** association to the appropriate **Element**, is currently implementation dependent. But, the set of **AffectedJobElement** associations to the **Input** and **Output** elements present when the job finishes execution will remain until the job is deleted.

Job Control object model

The Job Control profile provides a mechanism to handle asynchronous execution of methods that take some time to execute.

For a diagram of how this mechanism works, see Figure 14 on page 70.



Figure 14. Job Control SMI-S model for XIV systems

For **IBMTSDS_MirrorJob**, if all the operations completed successfully, the **JobState** will be 7 (Completed), and the **OperationStatus** will be 2 (Ok), 17 (Completed). If not, the **JobState** will be 10 (Exception), and the **OperationStatus** will be 6 (Error), 17 (Completed). If the job finished, and the duration time was over 60 minutes, the job will be deleted automatically.

Thin Provisioning profile

Thin provisioning is a capability of some Block Services implementations. It defers provisioning of backing store for regions of a volume until the regions are accessed (written) by the consumer (for example, host file system).

The alternatives (fully provisioned volumes) allocate all the requested capacity from the backing store at the time the volume is created. For thin provisioned volumes, the block server implementation tracks information about which regions are accessed. After a region is accessed, the backing storage is allocated. The Thin Provisioning profile allows SMI-S clients to determine whether a storage system (and children such as pools and volumes) supports thin provisioning. Clients can also determine the difference between the exposed “virtual capacity” and actual committed physical storage, and create thinly provisioned volumes and pools.

For the IBM XIV Storage System, thin provisioning is a pool definition (not a volume definition). It defines the pool as a thin provisioned pool and then assigns different soft and hard sizes to it. From the volume perspective, it is the same. You define the volume size (no soft size or hard size distinction), and it takes the size from the soft size of the pool. Upon creation of a volume in a regular or thin provisioned pool, no pre-allocation of hard capacity is done. Only soft capacity is taken from the soft capacity of the pool.

Thin Provisioning object model

Compared with the Block Services profile, the Thin Provisioning profile has more classes and some existing classes with modified properties.

See Figure 15 on page 71 for the Thin Provisioning SMI-S model for XIV systems.

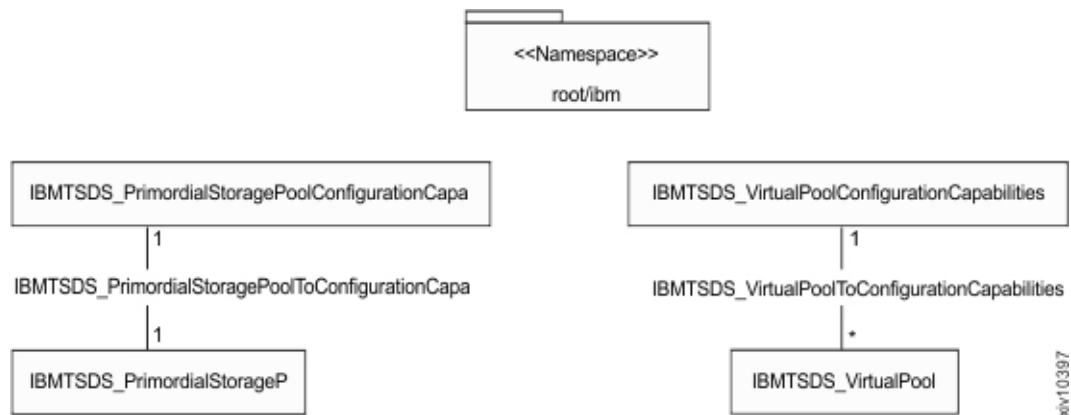


Figure 15. Thin Provisioning SMI-S model for XIV systems

As with the Block Services profile, the CIM agent has three `CIM_StorageSetting` instances, but `IBMTSDS_VirtualPoolSetting` adds an instance type in the Thin Provisioning profile. One is the default, and the other is a newly created one.

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 a regular `VirtualPool`.

IBMTSDS_VirtualPoolSetting.InstanceID="IBMTSDS:IBM.2810-1310129-112714143137626"

This `StorageSetting` is supplied as the `Goal` parameter for the `CreateOrModifyStoragePool` method to create or modify a thin provisioning `VirtualPool`.

For more information about the thin provisioning object model, see the `BlockServices.mof` file.

Extrinsic Methods

Table 36. Replication Service methods

Method	Description
CreateSetting	Creates an <code>IBMTSDS_VirtualPoolSetting</code> instance
modify	Modifies the properties of the newly created <code>IBMTSDS_VirtualPoolSetting</code> instance
delete	Deletes the newly created <code>IBMTSDS_VirtualPoolSetting</code> instance
CreateOrModifyStoragePool	Creates or modifies a regular <code>VirtualPool</code> or thin provisioning <code>VirtualPool</code>

IBMTSDS_PrimordialStoragePoolCapabilities. CreateSetting

You can use the `CreateSetting` method to create an “`IBMTSDS_VirtualPoolSetting`” instance.

SettingType

As an input, the value must be 2 (default), 3 (goal), or null.

NewSetting

As an output, refers to the newly created **IBMTSDS_VirtualPoolSetting** instance.

IBMTSDS_VirtualPoolSetting. modify

You can use the **modify** method to modify the properties of a newly created **IBMTSDS_VirtualPoolSetting** instance.

ThinProvisionedInitialReserve (*required if ThinProvisionedPoolType is null*)

As an input, represents the hard size of the newly created pool to be used as the **Goal** parameter for the **CreateOrModifyStoragePool** method.

ThinProvisionedPoolType (*required if ThinProvisionedInitialReserve is null*)

As an input, represents the pool type that you want to create to use as the goal parameter for the **CreateOrModifyStoragePool** method. 7 indicates that the newly created pool is a regular pool, and 8 indicates that the newly created pool is a thin provisioning pool.

Cop

As an input, represents the CIM Object Paths (COPs) of the newly created **IBMTSDS_VirtualPoolSetting** instance to modify.

IBMTSDS_VirtualPoolSetting. delete

You can use the **delete** method to delete the newly created **IBMTSDS_VirtualPoolSetting** instance. The default instance of **IBMTSDS_VirtualPoolSetting** cannot be deleted.

Cop

As an input, represents the CIM Object Paths (COPs) of the newly created **IBMTSDS_VirtualPoolSetting** instance to delete.

IBMTSDS_StorageConfigurationService. CreateOrModifyStoragePool

You can use the **CreateOrModifyStoragePool** method to create or modify a regular **VirtualPool** or a thin provisioning **VirtualPool**.

Pool (*required for modification*)

As an input parameter, specifies whether you want to create or modify a pool. If you specify a reference to a pool, indicates that you want to modify the pool. If the parameter is left null, indicates that you want to create a pool.

InPools

InPools specifies from which pool to create the pool. To create a **VirtualPool**, only the object reference of **IBMTSDS_PrimordialStoragePool** is allowed for this parameter since all **VirtualPools** are created in the **Primordial StoragePool**. To create a **SnapshotPool**, only the object reference of **IBMTSDS_VirtualPool** is allowed for this parameter if specified since **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 allowed. To create a **SnapshotPool**, only the object reference of

IBMTSDS_SnapshotPoolSetting is allowed. The **Goal** parameter also specifies the requested hard size of the **Pool**. See the `BlockServer.mof` file for details.

ElementName

The **ElementName** property provides a means for you to set a meaningful name for the pool that is being created or modified. If specified, it is limited to 63 characters and can contain letters, digits, blank spaces, `-`, `_`, `.` and `~` characters. Blank spaces cannot be the beginning and ending characters. If not specified during pool creation, a random pool name is generated in the format `pool<random integer>`.

Note: The name of the **Pool** must be unique in the system. 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, **Size** specifies the requested soft size of the Pool. Null or 0 is not allowed for pool creation. 0 for pool modification indicates that the pool's size is not changed. As an output parameter, **Size** specifies the soft size achieved. If the value entered is not supported, the size value is set to the nearest supported size.

Creating a thin provisioning pool

You can use the **CreateOrModifyStoragePool** method to create a thin provisioning pool.

About this task

This procedure provides an example for configuring an XIV device. There are other ways that 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), specify a size.
2. Call **invokeMethod** on **IBMTSDS_PrimordialStoragePoolCapabilities.CreateSetting** and specify the **SettingType**.
3. Call **enumerateInstances** on **IBMTSDS_VirtualPoolSetting**, and then save the **CIMObjectPath** of this newly created instance.
4. Call **invokeMethod** on **IBMTSDS_VirtualPoolSetting.modify** and specify the **IBMTSDS_VirtualPoolSetting** obtained in step 3 to **Cop**, pool hard size to **ThinProvisionedInitialReserve**, and pool type to **ThinProvisionedPoolType**.

Note: For the thin provisioning pool, the hard size must be less than the soft size.

5. Call **invokeMethod** on **IBMTSDS_StorageConfigurationService.CreateOrModifyStoragePool** and specify the **IBMTSDS_VirtualPoolSetting** obtained in step 3 to **Goal** and **ElementName** to **Size**.
6. Save the pool output parameter. It is the reference to the thin provisioning **VirtualPool** that was created.

Thin provisioning indications

The XIV CIM Agent supports three thin provisioning profile indications.

Table 37. Thin provisioning profile Indications

CIM Indication	XIV Event	Description
CAPACITY_WARNING	STORAGE_POOL_VOLUME_USAGE_TOO_HIGH	Thin provisioned <i>Volume or Pool</i> with the identifier <i>Volume or Pool ID</i> capacity in use near available limit.
CAPACITY_CRITICAL	STORAGE_POOL_EXHAUSTED	Thin provisioned <i>Volume or Pool</i> with the identifier <i>Volume or Pool ID</i> capacity in use exceeded available limit.
CAPACITY_OKAY	STORAGE_POOL_VOLUME_USAGE_BACK_TO_NORMAL	Thin provisioned <i>Volume or Pool</i> with the identifier <i>Volume or Pool ID</i> capacity condition cleared.

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 38. SNIA CTP results

XIV CIM Agent release	Official CTP Test Suite	Official CTP Test Results
10.2.4	<ul style="list-style-type: none">• SMI-S version 1.4• Test version 1.4.0.24	http://www.snia.org/ctp/conformingproviders/ibm.html#sftw3
11.0	<ul style="list-style-type: none">• SMI-S version 1.4• Test version 1.4.0.24	http://www.snia.org/ctp/conformingproviders/ibm.html#sftw3
11.1	<ul style="list-style-type: none">• SMI-S version 1.4• Test version 1.4.0.24	http://www.snia.org/ctp/conformingproviders/ibm.html#sftw3
11.2.x	<ul style="list-style-type: none">• SMI-S version 1.4• Test version 1.4.0.24	http://www.snia.org/ctp/conformingproviders/ibm.html#sftw3
11.3.x	<ul style="list-style-type: none">• SMI-S version 1.4• Test version 1.4.0.24	http://www.snia.org/ctp/conformingproviders/ibm.html#sftw3
11.4.x	<ul style="list-style-type: none">• SMI-S version 1.4• Test version 1.4.0.24	http://www.snia.org/ctp/conformingproviders/ibm.html#sftw3
11.5.x	<ul style="list-style-type: none">• SMI-S version 1.4• Test version 1.4.0.24	XIV CIM 11.5.0 has passed CTP1.4 test. Because CTP1.4 test was deprecated from 2013.11.8 by SNIA, the result is not posted on the SNIA website.
11.6.x	<ul style="list-style-type: none">• SMI-S version 1.6• Test version 1.6.0.816	http://www.snia.org/tech_activities/standards/curr_standards/smi

SCVMM Storage Automation tests

All Microsoft System Center Virtual Machine Manager (SCVMM) 2012 Storage Automation tests were completed on the IBM XIV Storage System Gen3 with the XIV CIM Agent version 11.6.0. For detailed information, refer to <http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/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.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, 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.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

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.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

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 at www.ibm.com/legal/copytrade.shtml.

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

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:

- The *American National Standard Dictionary for Information Systems*, ANSI X3.172–1990, copyright 1990 by the American National Standards Institute (ANSI), 11 West 42nd Street, New York, New York 10036. Definitions derived from this book have the symbol (A) after the definition.
- IBM Terminology, which is available online at the IBM Terminology website (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 Lightweight Directory Access Protocol.

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.

M**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

See *Simple Network Monitor Protocol*.

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

See *Transmission Control Protocol, Internet Protocol*.

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

AssociatorNames 19
Associators 18

B

Block Services 31
 object model 32

C

checking connectivity 8
CIM agent
 components 3
 concepts 3
 connectivity 8
 disabling 8
 enabling 8
 limitations 7
 overview 1
 port number 7
 restarting 8
 security 4
 XIV system 7
CIM classes
 CIM_AuthorizedPrivilege 49
 CIM_FCPort 49
 CIM_SCSIProtocolController 49
 CIM_SCSIProtocolEndPoint 49
 CIM_StorageHardwareID 49
 CIM_StorageVolume 49
 CIM_SystemSpecificCollection 49
CIM_AuthorizedPrivilege 49
CIM_ERR_ACCESS_DENIED 21
CIM_ERR_ACCESS_DENIED 11, 12, 13, 14, 16, 17, 18, 19, 20,
 22, 23
CIM_ERR_ALREADY_EXISTS 13
CIM_ERR_FAILED 11, 12, 13, 14, 16, 17, 18, 19, 20, 21, 22, 23
CIM_ERR_INVALID_CLASS 21
CIM_ERR_INVALID_NAMESPACE 11, 12, 13, 14, 16, 17, 18,
 19, 20, 21, 22, 23
CIM_ERR_INVALID_PARAMETER 11, 12, 13, 14, 16, 17, 18,
 19, 20, 21, 22, 23
CIM_ERR_INVALID_CLASS 12, 13, 14, 16, 17, 18, 20, 22
CIM_ERR_INVALID_QUERY 17
CIM_ERR_NO_SUCH_PROPERTY 20, 21, 22
CIM_ERR_NOT_FOUND 12, 13, 14, 20, 21, 22, 23
CIM_ERR_NOT_SUPPORTED 12, 13, 14, 17
CIM_ERR_QUERY_FEATURE_NOT_SUPPORTED 17
CIM_ERR_QUERY_LANGUAGE_NOT_SUPPORTED 17
CIM_ERR_TYPE_MISMATCH 22
CIM_FCPort 49
CIM_SCSIProtocolController 49
CIM_SCSIProtocolEndPoint 49
CIM_StorageHardwareID 49
CIM_StorageVolume 49
CIM_SystemSpecificCollection 49
CIMOM
 return codes 24

client application
 receiving CIMOM error codes 24
clusters 47
commands
 slptool findsrvs 8
component
 definitions 9
conformance testing
 SCVMM 75
 SNIA 75
CreateClass 13
CreateInstance 13

D

DeleteClass 12
DeleteInstance 13
DeleteQualifier 23
disabling
 CIM agent 8

E

enabling
 CIM agent 8
EnumerateClasses 14
EnumerateInstanceNames 17
EnumerateInstances 16
EnumerateQualifiers 23
error codes 24
ExecuteQuery 17

F

forums ix

G

GetClass 11
GetInstance 12
GetProperty 21
GetQualifier 22
group management
 Replication Services 60

H

hosts 47
HTTP
 status messages 24

I

IBM Tivoli Storage Productivity Center 8
IBMTSDS_FCPort 49
IBMTSDS_Privilege 49
IBMTSDS_SCSIProtocolController 49
IBMTSDS_SEVolume 49

IBMTSDS_StorageHardwareID 49
IBMTSDS_SystemSpecificCollection 49

J

Job Control
 object model 69

L

LUN mapping 47
LUN masking 47

M

Masking and Mapping 47
 object model 49
methods
 AssociatorNames 19
 Associators 18
 CreateClass 13
 CreateInstance 13
 DeleteClass 12
 DeleteInstance 13
 DeleteQualifier 23
 EnumerateClasses 14
 EnumerateInstanceNames 17
 EnumerateInstances 16
 EnumerateQualifiers 23
 ExecuteQuery 17
 GetClass 11
 GetInstance 12
 GetProperty 21
 GetQualifier 22
 ModifyClass 14
 ModifyInstance 14
 ReferenceNames 20
 References 20
 SetProperty 22
 SetQualifier 23
models
 Block Services 32
 Job Control 69
 Masking and Mapping 49
 Replication Services 57
 Thin Provisioning 70
ModifyClass 14
ModifyInstance 14

N

notices
 legal 77

P

packages
 Block Services 32
 Job Control 69
 Masking and Mapping 49
 Replication Services 57
 Thin Provisioning 70
parameters
 AssocClass 18, 19
 ClassName 11, 14, 16, 17

parameters (*continued*)
 DeepInheritance 14, 16
 IncludeClassOrigin 11, 12, 14, 16, 18, 20
 IncludeQualifiers 11, 14
 Instance 13, 14
 InstanceName 12, 13, 21, 22
 LocalOnly 11, 14
 ObjectName 18, 19, 20
 Property 21, 22
 QualifierName 22
 Query 17
 QueryLanguage 17
 ResultClass 18, 19, 20
 ResultRole 18, 19
 Role 18, 19, 20
 SetName 23
PDFs ix
port number 7
ports 47
profiles
 Block Services 31
 Masking and Mapping 47
publications ix

R

ReferenceNames 20
References 20
related information ix
Replication management 62
Replication Services
 group management 60
 object model 57
restarting
 CIM agent 8
return code 24
return values
 CIM_ERR_ACCESS_DENIED 11, 12, 13, 14, 16, 17, 18, 19,
 20, 21, 22, 23
 CIM_ERR_ALREADY_EXISTS 13
 CIM_ERR_FAILED 11, 12, 13, 14, 16, 17, 18, 19, 20, 21, 22,
 23
 CIM_ERR_INVALID_NAMESPACE 11, 12, 13, 14, 16, 17,
 18, 19, 20, 21, 22, 23
 CIM_ERR_INVALID_PARAMETER 11, 12, 13, 14, 16, 17,
 18, 19, 20, 21, 22, 23
 CIM_ERR_INVALID_CLASS 12, 13, 14, 16, 17, 18, 20, 21,
 22
 CIM_ERR_INVALID_QUERY 17
 CIM_ERR_NO_SUCH_PROPERTY 20, 21, 22
 CIM_ERR_NOT_FOUND 12, 13, 14, 20, 21, 22, 23
 CIM_ERR_NOT_SUPPORTED 12, 13, 14, 17, 23
 CIM_ERR_QUERY_FEATURE_NOT_SUPPORTED 17
 CIM_ERR_QUERY_LANGUAGE_NOT_SUPPORTED 17
 CIM_ERR_TYPE_MISMATCH 22
descriptions 24

S

SCVMM testing 75
secure connection
 port number 7
SetProperty 22
SetQualifier 23
SLP discovery 8
slptool findsrvs 8

SNAI testing 75
starting
 CIM agent 8
storage pools 31, 47

T

Thin Provisioning
 object model 70
trademarks 79

V

volumes 31, 47

X

XIV CIM classes
 IBMTSDS_SystemSpecificCollection 49
 IBMTSDS_FCPort 49
 IBMTSDS_Privilege 49
 IBMTSDS_SCSIProtocolController 49
 IBMTSDS_SEVolume 49
 IBMTSDS_StorageHardwareID 49
XIV Open API
 components 9
 overview 1



Printed in USA

GC27-3916-07

