IBM Storwize V7000

CIM Agent Developer's Guide

Version 6.1.0

GC27-2292-00

IBM Storwize V7000

CIM Agent Developer's Guide

Version 6.1.0

GC27-2292-00

Note

Before using this information and the product it supports, read the information in “Notices” on page 65.

This edition applies to the IBM Storwize V7000 , Version 6.1.0, and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Figures

Tables

About this guide

This publication introduces the Common Information Model (CIM) agent for the IBM® System Storage Storwize® V7000.

This section describes:
- Content and intended audience of this book
- Typefaces that are used to show emphasis
- Information that is related to this book
- Websites that provide information about the Storwize V7000 or related products or technologies
- How to order IBM publications
- How to send in your feedback on this book

Who should use this guide

This reference book is for application programmers who are developing with the Common Information Model.

This reference book is for CIM-based application programmers who want to perform the following tasks:
- Understand the CIM agent for the Storwize V7000
- Discover and connect to the CIM agent service
- Retrieve and extract the CIM agent object classes, attributes, and methods
- Create new object instances for basic storage configuration, logical unit number (LUN) masking, and Copy Services on the Storwize V7000

Emphasis

Different typefaces are used in this guide to show emphasis.

The following typefaces are used to show emphasis:

Boldface	Text in **boldface** represents menu items and command names.
Italics	Text in *italics* is used to emphasize a word. In command syntax, it is used for variables for which you supply actual values, such as a default directory or the name of a cluster.
`Monospace`	Text in `monospace` identifies the data or commands that you type, samples of command output, examples of program code or messages from the system, or names of command flags, parameters, arguments, and name-value pairs.

Storwize V7000 library and related publications

Product manuals, other publications, and websites contain information that relates to Storwize V7000.

Storwize V7000 Information Center

The IBM Storwize V7000 Information Center contains all of the information that is required to install, configure, and manage the Storwize V7000. The information center is updated between Storwize V7000 product releases to provide the most current documentation. The information center is available at the following website:

publib.boulder.ibm.com/infocenter/storwize/ic/index.jsp

Storwize V7000 library

Unless otherwise noted, the publications in the Storwize V7000 library are available in Adobe portable document format (PDF) from the following website:

Support for Storwize V7000 website at www.ibm.com/storage/support/storwize/v7000

IBM documentation and related websites

Table 1 lists websites that provide publications and other information about the Storwize V7000 or related products or technologies.

Table 1. IBM documentation and related websites

Website	Address
Support for Storwize V7000 (2076)	Support for Storwize V7000 website at www.ibm.com/storage/support/storwize/v7000
Support for IBM System Storage® and IBM TotalStorage products	www.ibm.com/storage/support/
IBM Publications Center	www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss
IBM Redbooks® publications	www.redbooks.ibm.com/

Related accessibility information

To view a PDF file, you need Adobe Acrobat Reader, which can be downloaded from the Adobe website:

www.adobe.com/support/downloads/main.html

How to order IBM publications

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

The IBM Publications Center 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. You can access the IBM Publications Center through the following website:

www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss

Sending your comments

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

To submit any comments about this book or any other Storwize V7000 documentation:

- Go to the feedback page on the website for the Storwize V7000 Information Center at publib.boulder.ibm.com/infocenter/storwize/ic/index.jsp?topic=/com.ibm.storage.Storwize V7000.console.doc/feedback.htm. There you can use the feedback page to enter and submit comments or browse to the topic and use the feedback link in the running footer of that page to identify the topic for which you have a comment.
- Send your comments by email to starpubs@us.ibm.com. Include the following information for this publication or use suitable replacements for the publication title and form number for the publication on which you are commenting:
  - Publication title: *IBM Storwize V7000 CIM Agent Developer's Guide*
  - Publication form number: GC27-2292-00
  - Page, table, or illustration numbers that you are commenting on
  - A detailed description of any information that should be changed

Chapter 1. Introduction to CIM agent

The Common Information Model (CIM) provides an open approach to the design and implementation of storage systems, applications, databases, networks, and devices. The CIM agent is a set of standards that is developed by the Distributed Management Task Force (DMTF).

The following information introduces the Storage Management Initiative Specification (SMI-S), the CIM agent, the IBM Storwize V7000, and the CIM agent for the Storwize V7000. Functional views of the CIM agent object models illustrate the architecture and specific functions of the CIM agent.

Storage Management Initiative Specification

The Storage Management Initiative Specification (SMI-S) is a design specification of the Storage Management Initiative (SMI) that is launched by the Storage Networking Industry Association (SNIA).

The SMI-S specifies a secure and reliable interface that allows storage management systems to identify, classify, monitor, and control physical and logical resources in a storage area network (SAN). The interface integrates the various devices to be managed in a SAN and the tools used to manage them.

SMI-S is based on a number of existing technologies or industry standards that include the following:

Common Information Model (CIM)
An object model for data storage and management that is developed by the Distributed Management Task Force (DMTF). CIM makes it possible to organize devices and components of devices in an object-oriented pattern.

Web-Based Enterprise Management (WBEM)
A tiered enterprise management architecture that is also developed by the DMTF. This architecture provides the management design framework that consists of devices, device providers, the object manager, and the messaging protocol for the communication between client applications and the object manager. In the case of the CIM, the object manager is the CIMOM and the messaging protocol is the CIM over HTTP technology. The CIM over HTTP approach specifies that the CIM data is encoded in XML and sent in specific messages between the client applications and the CIMOM over the TCP/IP network in a SAN.

Service Location Protocol (SLP)
A directory service that the client application uses to locate the CIMOM.

Intended to be an industry standard, SMI-S extends the generic capabilities of the CIM, the WBEM, and the SLP to implement storage networking interoperability. For example, the WBEM provides provisions for security, resource-locking management, event notification, and service discovery.

For more information about SMI-S conforming profiles that are supported by CIM agent for the Storwize V7000 and writing standards-based implementations, see the following website:

www.snia.org/forums/smi/tech_programs/smis_home/

CIM agent

The Common Information Model (CIM) agent is a set of standards that is developed by the Distributed Management Task Force (DMTF).

The CIM provides an open approach to the design and implementation of storage systems, applications, databases, networks, and devices.

The CIM specifications provide the language and the methodology for describing management data. For example, CIM schema 2.7 for Managing Storage Arrays specifies how to configure the management environment for data management in a common way. The CIM defines common object classes, associations, and methods. Member vendors can use those objects and extend them to specify how data can be processed and organized in a specific managed environment.

CIM agent concepts

This information describes the basic terminology and functions of the Common Information Model (CIM) agent object models.

The CIM agent specifications use the following concepts and terminology to describe the object models:

Association
A class with two references that define a relationship between two referenced objects.

Class
The definition of an object within a specific hierarchy. An object class can have properties and methods and serve as the target of an association.

Indication
An object representation of an event.

Instance
An individual object that is the member of a class. In object-oriented programming, an object that is created by instantiating a class.

Managed Object Format (MOF)
A language for defining Common Information Model (CIM) schemas.

Method
A way to implement a function on a class.

Namespace
The scope within which a CIM schema applies.

Object path
An object that consists of a namespace path and a model path. The namespace path provides access to the CIM implementation that the CIM agent manages, and the model path provides navigation within the implementation.

Property
An attribute that is used to characterize instances of a class.

Qualifier
A value that provides additional information about a class, association, indication, method, method parameter, instance, property, or reference.

Reference
A pointer to another instance that defines the role and scope of an object in an association.

Schema
A group of object classes defined for and applicable to a single namespace. Within the CIM agent, the supported schemas are the ones that are loaded through the Managed Object Format (MOF) compiler.

CIM agent components

With a Common Information Model (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.

Components

A CIM agent involves the following components:

agent code
An open-systems standard that interprets CIM requests and responses as they transfer between the client application and the device.

CIM object manager (CIMOM)
The common conceptual framework for data management that receives, validates, and authenticates the CIM requests from the client application. It then directs the requests to the appropriate component or device provider. Storwize V7000 is based on the Open Pegasus version 2.9.0 CIMOM.

client application
A storage management program that initiates CIM requests to the CIM agent for the device.

device The storage server that processes and hosts the client application requests.

device provider
A device-specific handler that serves as a plugin for the CIM. That is, the CIMOM uses the handler to interface with the device.

Service Location Protocol (SLP)
A directory service that the client application calls to locate the CIMOM.

CIM agent for Storwize V7000

The Common Information Model (CIM) agent for Storwize V7000 serves as a configuration interface for the Storwize V7000.

The CIM agent consists of the following main components:

- CIM object manager (CIMOM)
- Service Location Protocol (SLP) agent
- Storwize V7000 provider

Examples of how the CIM agent works

The Common Information Model (CIM) agent can be used to provide common building blocks to manage CIM-compliant devices.

Typical CIM agent configuration

Figure 1 on page 4 shows how a typical CIM agent works.

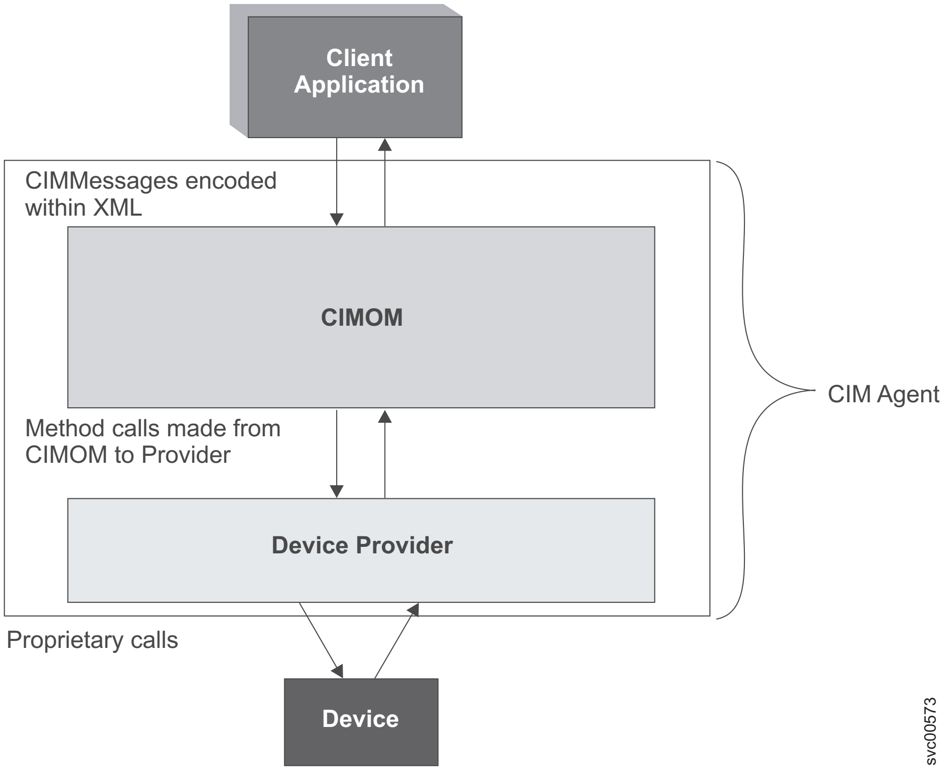


Figure 1. A typical CIM agent at work

A typical configuration for your CIM client is to use the CIMOM that is located at https://*cluster_ip* port 5989 with `namespace=/root/ibm`. (The standard secure port is 5989.)

The client application sends CIM requests to the CIMOM. As requests arrive, the CIMOM validates and authenticates each request. It then directs the requests to the appropriate functional component of the CIMOM or to a device provider. To satisfy client application requests, the provider makes calls to a device-unique programming interface on behalf of the CIMOM.

The management application can obtain an instance of the RemoteServiceAccessPoint from the CIMOM. This instance allows the management application to access the Web User Interface.

CIM agent configuration with Service Location Protocol

If you use Service Location Protocol (SLP) to discover the CIMOM, the client application locates the CIMOM by calling an SLP directory service. When the CIMOM is first invoked, it registers itself to the SLP Service agent and supplies its location, IP address, port number, and the type of service that it provides. A string that describes the CIM agent access point is registered.

The following output provides an example of the registered string:

```
service:wbem:https://<CIM Agent IP>:<port number>
```

The SLP provides the following attributes:

```
template-type=wbem
template-version=1.0
template-description=This template describes the attributes used for
advertising WBEM servers.
template-url-syntax=https://9.47.24.91:5989
service-location-tcp=https://9.47.24.91:5989
service-hi-name=IBM System Storage Storwize V7000 CIMOM
service-hi-description=IBM Storwize V7000 CIM Agent Version 1.0.0.xxx
```

```
service-id=IBMTSSVC:9.47.24.91
ProtocolVersion=1.2
CommunicationMechanism=cim-xml
FunctionalProfilesSupported=Basic Read, Basic Write, Instance Manipulation,
Association, Traversal, Query Execution, Qualifier Declaration, Indications
AuthenticationMechanismSupported=Basic
Namespace=/root/ibm
InteropSchemaNamespace=/root/ibm
MultipleOperationsSupported=false
RegisteredProfilesSupported=SNIA:Storage Virtualizer,SNIA:Storage Virtualizer:
Access Points,SNIA:Storage Virtualizer:Block Services,SNIA:Storage Virtualizer:
Cascading,SNIA:Storage Virtualizer:Copy Services,SNIA:Storage Virtualizer:
FC Initiator Ports,SNIA:Storage Virtualizer:FC Target Ports,SNIA:Storage
Virtualizer:Health,SNIA:Storage Virtualizer:Masking and Mapping,SNIA:Storage
Virtualizer:Multiple Computer System,SNIA:Storage Virtualizer:Physical Package,
SNIA:Storage Virtualizer:Software,SNIA:Server,SNIA:Server:Profile Registration,
SNIA:Server:Indication,SNIA:SMI-S
```

With this information, the client application starts to directly communicate with the CIMOM.

CIM agent configuration on the console and the cluster

In Storwize V7000 6.1.0, the CIMOM resides on the Storwize V7000 cluster. The management GUI uses the CIMOM that is on the cluster. Other IBM storage management tools continue to use the CIMOM that is on the cluster.

Figure 2 shows how this CIMOM configuration is defined.

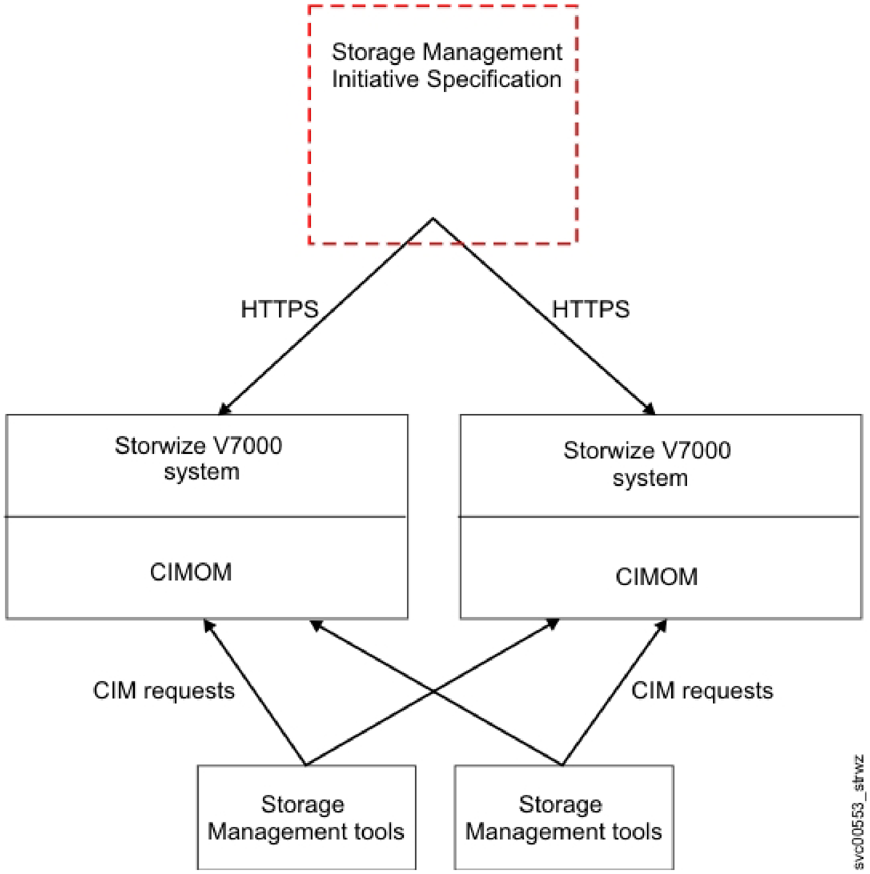


Figure 2. CIMOM on the cluster

Storwize V7000 CIMOM programming

CIM object manager (CIMOM) programming provides the opportunity to handle multiple connections from multiple sources while maintaining security. CIM clients connect to the CIMOM with a user name and password and then invoke methods to run commands.

The creation of a CIM client requires a suitable framework such as the Java WBEM Service project, the SBLIM CIM Client for Java, and the Aperi open source project. There are also implementations in other languages, including C++ and Python. For more information, see the following websites:

Java WBEM Service project
wbemservices.sourceforge.net/

SBLIM CIM Client for Java
www.sblim.wiki.sourceforge.net/CimClient

Aperi www.eclipse.org/aperi/faq/

Figure 3 shows a simple Java program that connects to a Storwize V7000 CIMOM.

```
import java.util.*;

import javax.wbem.cim.*;
import javax.wbem.client.*;

public class ITSOClient {
  public static void main(String[] args)
  {
    String username = args[0];
    String password = args[1];
    String masterConsoleIP = args[2];
    String masterConsoleSecurePort = args[3];
    UserPrincipal user = new UserPrincipal(username);
    PasswordCredential pwd = new PasswordCredential(password);
    CIMNameSpace ns = new CIMNameSpace("https://"+
         masterConsoleIP+":"+
         masterConsoleSecurePort+"/root/ibm");

    CIMClient client = null;
    try
    {
      System.out.println("Connecting to CIMOM");
      client = new CIMClient(ns,user,pwd);
   }
   catch (CIMException e)
   {
     // Handle the CIM Exception
     e.printStackTrace();
    }
}
```

Figure 3. Java program for connecting to a Storwize V7000 CIMOM

To view the CIM agent style pages that are complete Managed Object Format (MOF) documentation that is shipped with the management GUI, select the documentation information from the following website:

Support for Storwize V7000 website at www.ibm.com/storage/support/storwize/v7000

How Storwize V7000 concepts map to CIM concepts

To administer the Storwize V7000 through the CIM object manager (CIMOM), it is important to understand the relationship between Storwize V7000 and Common Information Model (CIM) concepts.

Table 2 shows how these concepts relate to one another.

Table 2. Relating Storwize V7000 concepts to CIM concepts

	CIM	
Storwize V7000 concept	**CIM name**	**CIM concept**
Cluster	IBMTSSVC_Cluster	Class
ClusterName	ElementName	Property
Cluster ID	Name	Property
VDisk	IBMTSSVC_StorageVolume	Class
VDisk ID	DeviceID	Property
FlashCopy® Consistency Group (regular)	IBMTSSVC_FlashCopyStorageSynchronizedSet	Class
FlashCopy Consistency Group (autodelete)	IBMTSSVC_CloneCopyStorageSynchronizedSet	Class
FlashCopy Mapping	IBMTSSVC_LocalStorageSynchronized	Association
FlashCopy Mapping Status	SyncState	Property
mkfcmap	AttachReplica	Method
preparefcmap	ModifySynchronization	Method
startfcmap	ModifySynchronization	Method
Remote Copy Consistency Group (Global Mirror)	IBMTSSVC_AsyncCopyStorageSynchronizedSet	Class
Remote Copy Consistency Group (Metro Mirror)	IBMTSSVC_SyncCopyStorageSynchronizedSet	Class
Remote Copy relationship	IBMTSSVC_RemoteStorageSynchronized	Association
Remote Copy relationship state	NativeState	Property
mkrcrelationship	AttachReplica	Method
startrcrelationship	ModifySynchronization	Method
mdisk	IBMTSSVC_BackendVolume	Class
mdiskgrp	IBMTSSVC_ConcreteStoragePool	Class
mkvdisk	CreateOrModifyElementFromStoragePool	Method
mkmdiskgrp	CreateOrModifyStoragePool	Method
rmvdisk	ReturnToStoragePool	Method
rmmdiskgrp	DeleteStoragePool	Method
Host (with regard to ports)	IBMTSSVC_StorageHardwareID	Class
Host (with regard to volume (VDisk) mapping)	IBMTSSVC_ProtocolController	Class
mkvdiskhostmap	ExposePaths	Method
rmvdiskhostmap	HidePaths	Method

Table 2. Relating Storwize V7000 concepts to CIM concepts (continued)

	CIM	
Storwize V7000 concept	**CIM name**	**CIM concept**
mkhost	CreateStorageHardwareID	Method
rmhost	DeleteStorageHardwareID	Method
DiskDrive	IBMTSSVC_DiskDrive	Class
Array	IBMTSSVC_Array	Class
mkarray	CreateOrModifyElementFromElements	Method
charray	CreateOrModifyElementFromElements	Method
rmarray	DeleteArray	Method
Enclosure	IBMTSSVC_Enclosure	Class

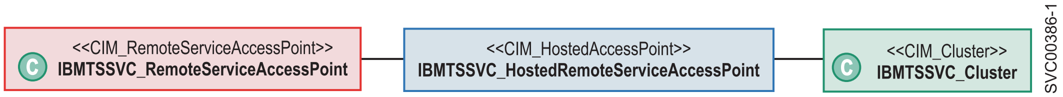
Functional diagrams of the Common Information Model agent

The functional diagrams of the Common Information Model (CIM) agent object model show specific functions that the CIM agent provides, including storage configuration service, Copy Services, LUN masking, and security. The diagrams also illustrate the architecture of the CIM Agent for the Storwize V7000.

Access point subprofile

The access point subprofile of the Common Information Model (CIM) agent for the Storwize V7000 consists of several basic classes.

Figure 4 shows the basic classes (building blocks) for the model.



<<CIM_RemoteServiceAccessPoint>>
C
IBMTSSVC_RemoteServiceAccessPoint
<<CIM_HostedAccessPoint>>
IBMTSSVC_HostedRemoteServiceAccessPoint
<<CIM_Cluster>>
C
IBMTSSVC_Cluster
SVC00386-1

Figure 4. High-level overview of the access point subprofile of the CIM agent

Block server performance subprofile

The block server performance subprofile describes performance statistics.

Figure 5 on page 9 describes the performance statistics of IBMTSSVC_Node, IBMTSSVC_StorageVolume, IBMTSSVC_BackendVolume, and IBMTSSVC_DiskDrive.

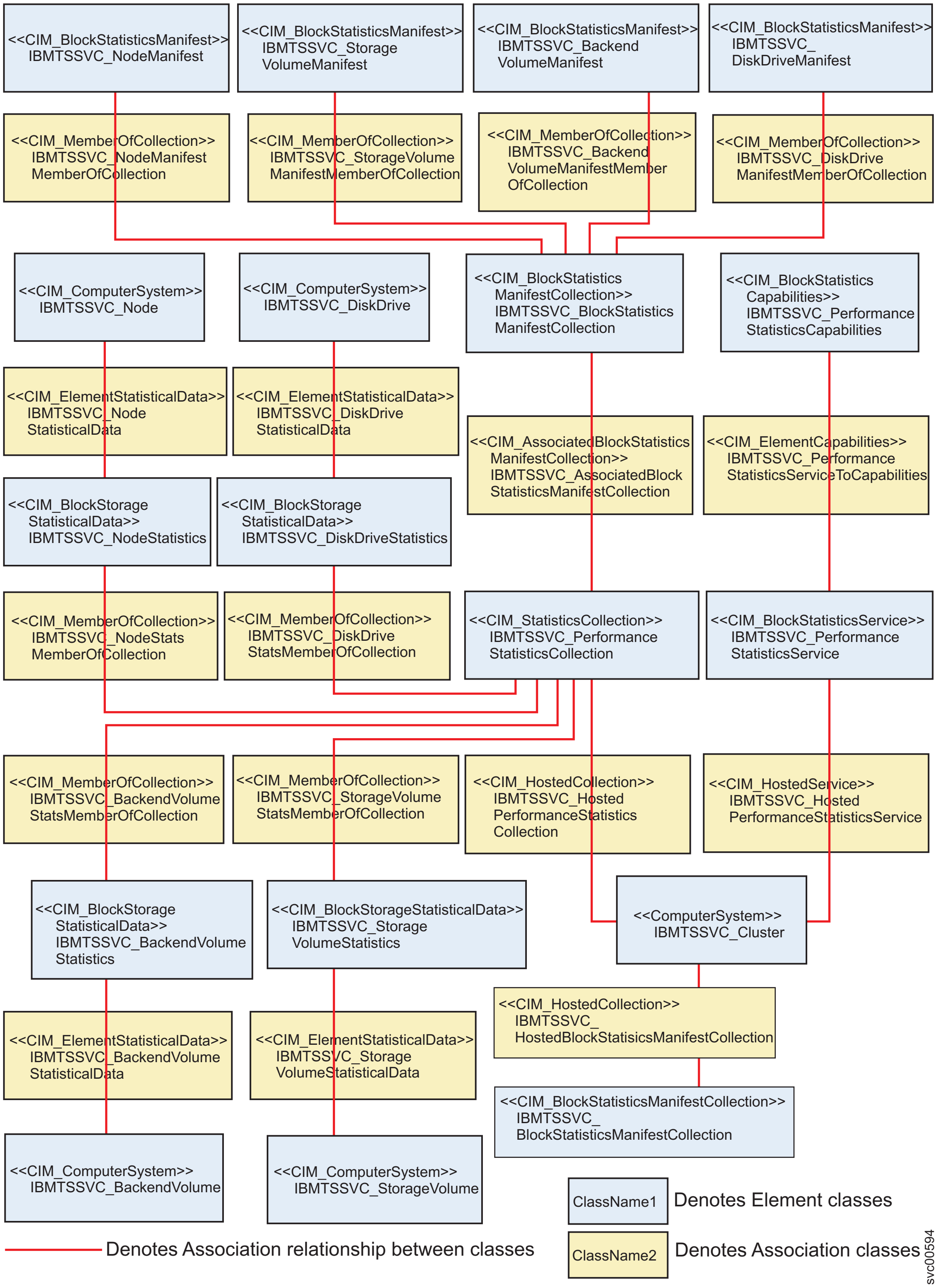

<<CIM_BlockStatisticsManifest>> IBMTSSVC_NodeManifest
<<CIM_BlockStatisticsManifest>> IBMTSSVC_Storage VolumeManifest
<<CIM_BlockStatisticsManifest>> IBMTSSVC_Backend VolumeManifest
<<CIM_BlockStatisticsManifest>> IBMTSSVC_ DiskDriveManifest
<<CIM_MemberOfCollection>> IBMTSSVC_NodeManifest MemberOfCollection
<<CIM_MemberOfCollection>> IBMTSSVC_StorageVolume ManifestMemberOfCollection
<<CIM_MemberOfCollection>> IBMTSSVC_Backend VolumeManifestMember OfCollection
<<CIM_MemberOfCollection>> IBMTSSVC_DiskDrive ManifestMemberOfCollection
<<CIM_ComputerSystem>> IBMTSSVC_Node
<<CIM_ComputerSystem>> IBMTSSVC_DiskDrive
<<CIM_BlockStatistics ManifestCollection>> IBMTSSVC_BlockStatistics ManifestCollection
<<CIM_BlockStatistics Capabilities>> IBMTSSVC_Performance StatisticsCapabilities
<<CIM_ElementStatisticalData>> IBMTSSVC_Node StatisticalData
<<CIM_ElementStatisticalData>> IBMTSSVC_DiskDrive StatisticalData
<<CIM_AssociatedBlockStatistics ManifestCollection>> IBMTSSVC_AssociatedBlock StatisticsManifestCollection
<<CIM_ElementCapabilities>> IBMTSSVC_Performance StatisticsServiceToCapabilities
<<CIM_BlockStorage StatisticalData>> IBMTSSVC_NodeStatistics
<<CIM_BlockStorage StatisticalData>> IBMTSSVC_DiskDriveStatistics
<<CIM_MemberOfCollection>> IBMTSSVC_NodeStats MemberOfCollection
<<CIM_MemberOfCollection>> IBMTSSVC_DiskDrive StatsMemberOfCollection
<<CIM_StatisticsCollection>> IBMTSSVC_Performance StatisticsCollection
<<CIM_BlockStatisticsService>> IBMTSSVC_Performance StatisticsService
<<CIM_MemberOfCollection>> IBMTSSVC_BackendVolume StatsMemberOfCollection
<<CIM_MemberOfCollection>> IBMTSSVC_StorageVolume StatsMemberOfCollection
<<CIM_HostedCollection>> IBMTSSVC_Hosted PerformanceStatistics Collection
<<CIM_HostedService>> IBMTSSVC_Hosted PerformanceStatisticsService
<<CIM_BlockStorage StatisticalData>> IBMTSSVC_BackendVolume Statistics
<<CIM_BlockStorageStatisticalData>> IBMTSSVC_Storage VolumeStatistics
<<ComputerSystem>> IBMTSSVC_Cluster
<<CIM_HostedCollection>> IBMTSSVC_ HostedBlockStatisicsManifestCollection
<<CIM_ElementStatisticalData>> IBMTSSVC_BackendVolume StatisticalData
<<CIM_ElementStatisticalData>> IBMTSSVC_Storage VolumeStatisticalData
<<CIM_BlockStatisticsManifestCollection>> IBMTSSVC_ BlockStatisticsManifestCollection
<<CIM_ComputerSystem>> IBMTSSVC_BackendVolume
<<CIM_ComputerSystem>> IBMTSSVC_StorageVolume
ClassName1 Denotes Element classes
Denotes Association relationship between classes
ClassName2 Denotes Association classes
svc00594

Figure 5. Block server performance subprofile

.

Block services profile

You can use several object classes to manipulate storage pools.

The block services profile allocates back-end storage volumes into storage pools and then creates storage volumes.

Figure 6 on page 11 provides a high-level overview of the object classes that you can use for pool manipulation.

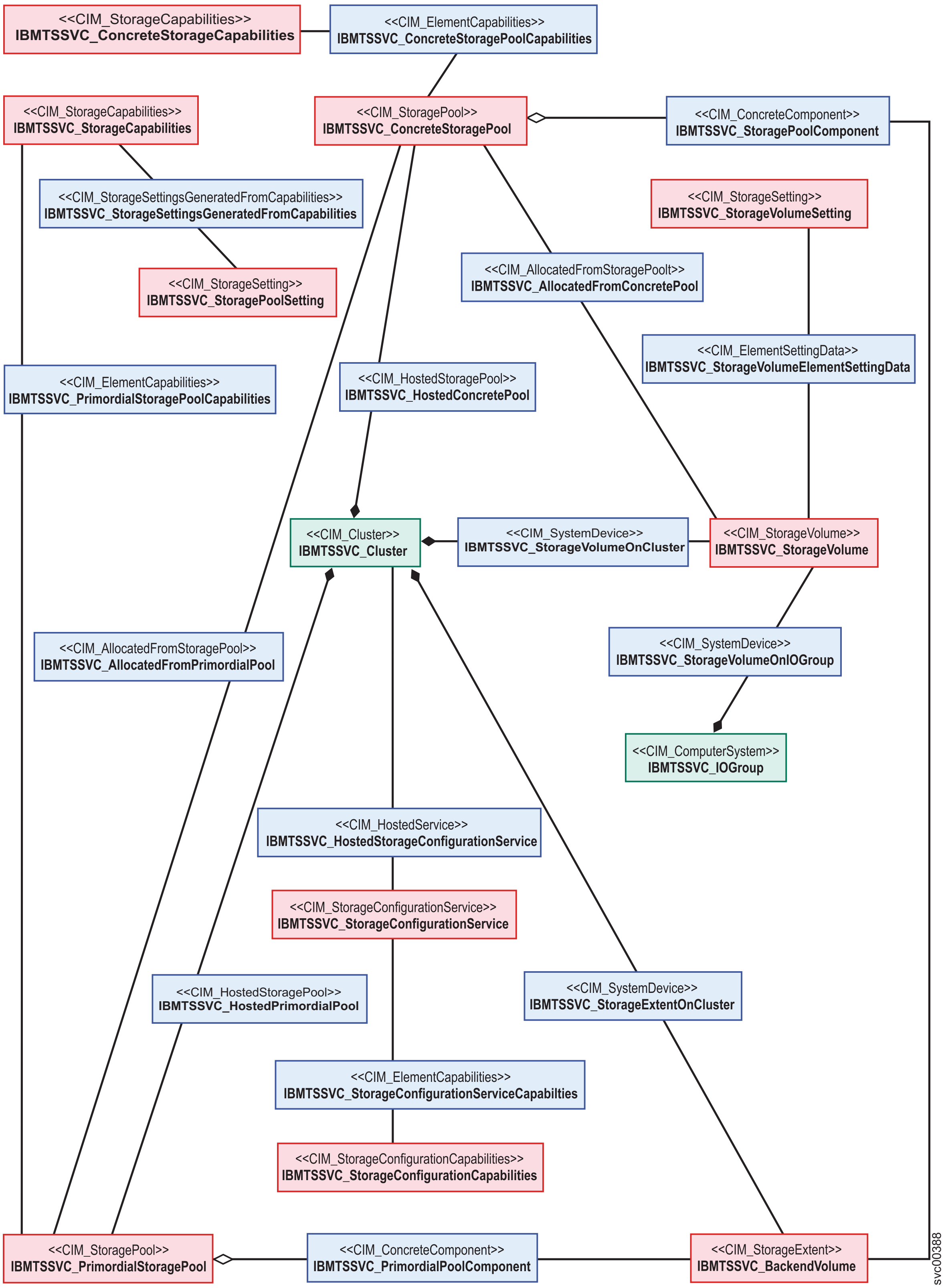

<<CIM_StorageCapabilities>>
IBMTSSVC_ConcreteStorageCapabilities
<<CIM_ElementCapabilities>>
IBMTSSVC_ConcreteStoragePoolCapabilities
<<CIM_StorageCapabilities>>
IBMTSSVC_StorageCapabilities
<<CIM_StoragePool>>
IBMTSSVC_ConcreteStoragePool
<<CIM_ConcreteComponent>>
IBMTSSVC_StoragePoolComponent
<<CIM_StorageSettingsGeneratedFromCapabilities>>
IBMTSSVC_StorageSettingsGeneratedFromCapabilities
<<CIM_StorageSetting>>
IBMTSSVC_StorageVolumeSetting
<<CIM_StorageSetting>>
IBMTSSVC_StoragePoolSetting
<<CIM_AllocatedFromStoragePoolt>>
IBMTSSVC_AllocatedFromConcretePool
<<CIM_ElementSettingData>>
IBMTSSVC_StorageVolumeElementSettingData
<<CIM_ElementCapabilities>>
IBMTSSVC_PrimordialStoragePoolCapabilities
<<CIM_HostedStoragePool>>
IBMTSSVC_HostedConcretePool
<<CIM_Cluster>>
IBMTSSVC_Cluster
<<CIM_SystemDevice>>
IBMTSSVC_StorageVolumeOnCluster
<<CIM_StorageVolume>>
IBMTSSVC_StorageVolume
<<CIM_AllocatedFromStoragePool>>
IBMTSSVC_AllocatedFromPrimordialPool
<<CIM_SystemDevice>>
IBMTSSVC_StorageVolumeOnIOGroup
<<CIM_ComputerSystem>>
IBMTSSVC_IOGroup
<<CIM_HostedService>>
IBMTSSVC_HostedStorageConfigurationService
<<CIM_StorageConfigurationService>>
IBMTSSVC_StorageConfigurationService
<<CIM_HostedStoragePool>>
IBMTSSVC_HostedPrimordialPool
<<CIM_SystemDevice>>
IBMTSSVC_StorageExtentOnCluster
<<CIM_ElementCapabilities>>
IBMTSSVC_StorageConfigurationServiceCapabilties
<<CIM_StorageConfigurationCapabilities>>
IBMTSSVC_StorageConfigurationCapabilities
<<CIM_StoragePool>>
IBMTSSVC_PrimordialStoragePool
<<CIM_ConcreteComponent>>
IBMTSSVC_PrimordialPoolComponent
<<CIM_StorageExtent>>
IBMTSSVC_BackendVolume
svc00388

Figure 6. High-level overview of block services.

RAID support for block services

RAID array classes provide support for the block services profile.

This diagram illustrates the classes and their relationships in regard to RAID (array) functions in the block services profile.

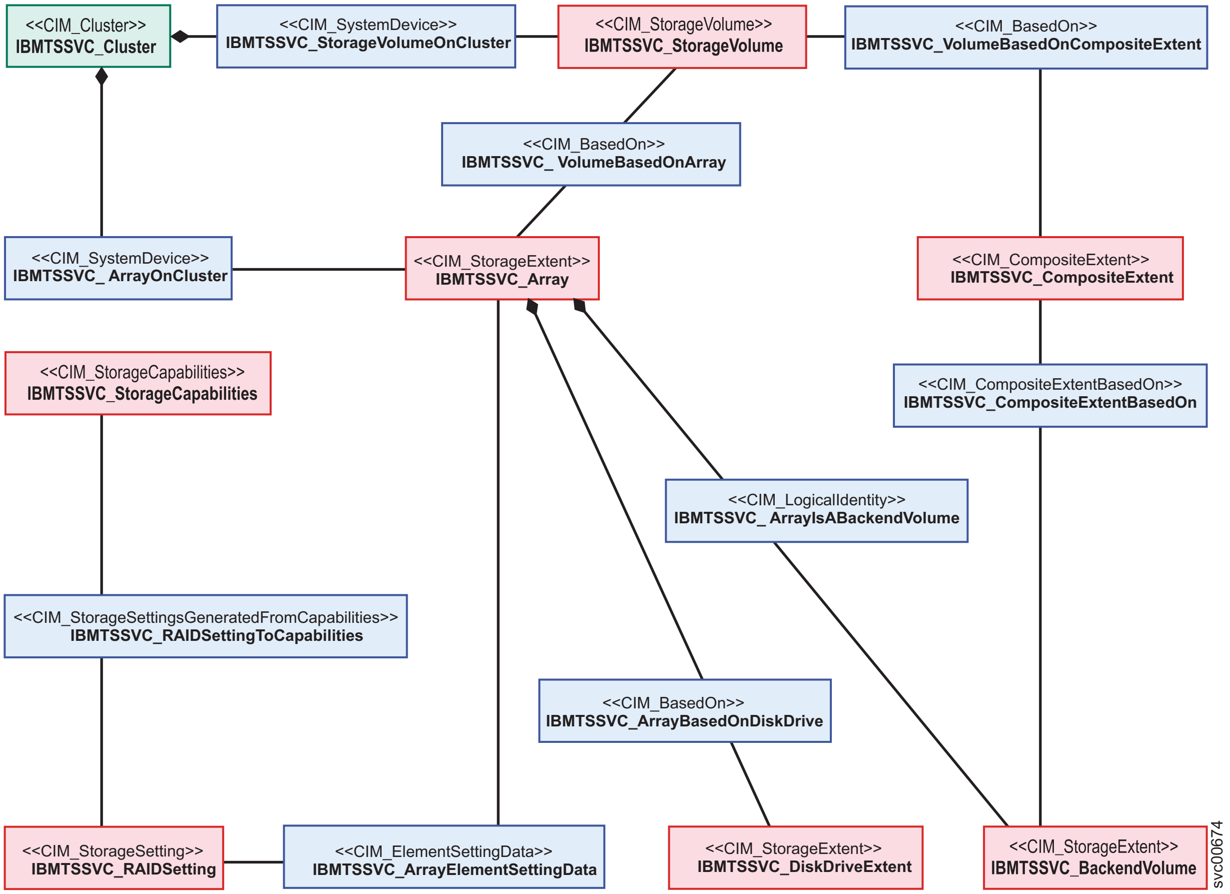


Figure 7. RAID support for block services

.

Cluster subprofile

There are several classes and associations that provide the function of a clustering service.

Figure 8 on page 13 shows the classes and associations that provide the function of a clustering service.

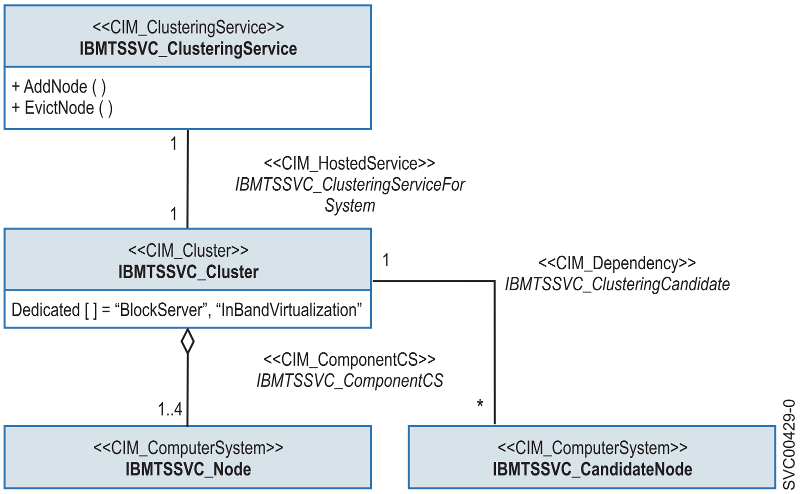


Figure 8. Class diagram of clustering instance

Copy Services

The IBMTSSVC_StorageConfigurationService class provides the methods to create copy relationships.

Figure 9 on page 14 shows the object classes that provide FlashCopy, Metro Mirror, and Global Mirror Copy Services.

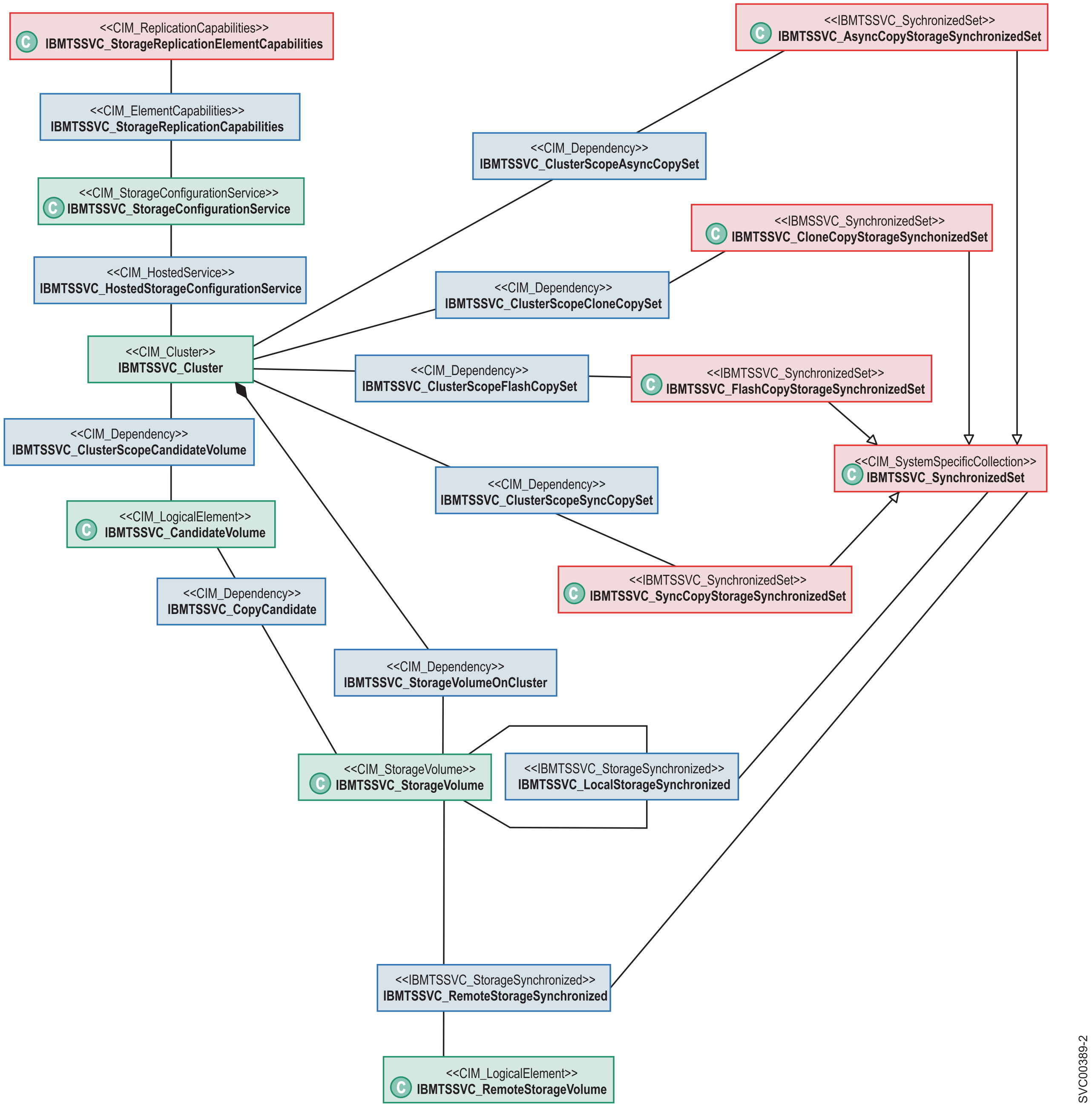

<<CIM_ReplicationCapabilities>>
IBMTSSVC_StorageReplicationElementCapabilities
<<CIM_ElementCapabilities>>
IBMTSSVC_StorageReplicationCapabilities
<<CIM_StorageConfigurationService>>
IBMTSSVC_StorageConfigurationService
<<CIM_HostedService>>
IBMTSSVC_HostedStorageConfigurationService
<<CIM_Cluster>>
IBMTSSVC_Cluster
<<CIM_Dependency>>
IBMTSSVC_ClusterScopeCandidateVolume
<<CIM_LogicalElement>>
IBMTSSVC_CandidateVolume
<<CIM_Dependency>>
IBMTSSVC_CopyCandidate
<<IBMTSSVC_SychronizedSet>>
IBMTSSVC_AsyncCopyStorageSynchronizedSet
<<CIM_Dependency>>
IBMTSSVC_ClusterScopeAsyncCopySet
<<IBMSSVC_SynchronizedSet>>
IBMTSSVC_CloneCopyStorageSynchonizedSet
<<CIM_Dependency>>
IBMTSSVC_ClusterScopeCloneCopySet
<<CIM_Dependency>>
IBMTSSVC_ClusterScopeFlashCopySet
<<IBMTSSVC_SynchronizedSet>>
IBMTSSVC_FlashCopyStorageSynchronizedSet
<<CIM_SystemSpecificCollection>>
IBMTSSVC_SynchronizedSet
<<CIM_Dependency>>
IBMTSSVC_ClusterScopeSyncCopySet
<<IBMTSSVC_SynchronizedSet>>
IBMTSSVC_SyncCopyStorageSynchronizedSet
<<CIM_Dependency>>
IBMTSSVC_StorageVolumeOnCluster
<<CIM_StorageVolume>>
IBMTSSVC_StorageVolume
<<IBMTSSVC_StorageSynchronized>>
IBMTSSVC_LocalStorageSynchronized
<<IBMTSSVC_StorageSynchronized>>
IBMTSSVC_RemoteStorageSynchronized
<<CIM_LogicalElement>>
IBMTSSVC_RemoteStorageVolume
SVC00389-2

Figure 9. Class diagram of copy services instances

Disk drive lite profile

The disk drive is the primary hardware element of the disk drive lite profile.

Figure 10 on page 15 shows the basic elements of the disk drive lite profile.

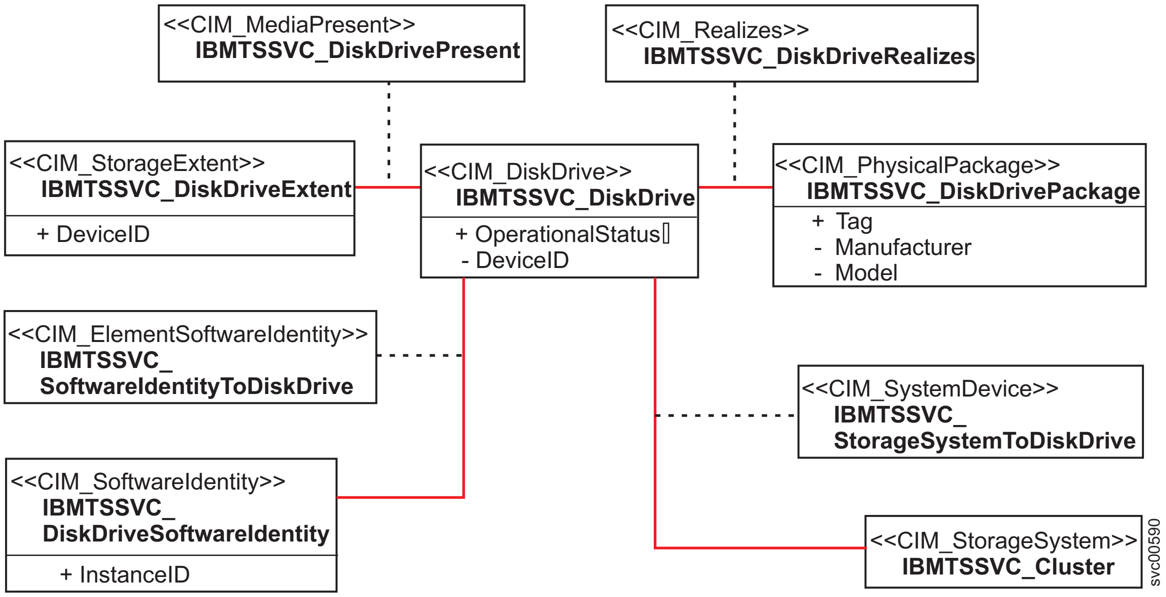


Figure 10. Overview of the Unified Modeling Language elements of the disk drive lite profile

.

Disk-drive sparing

Disk-drive sparing has a basic structure associated with RedundancySet.

The disk-drive sparing RedundancySet structure is composed of one instance class and three association classes:

Instance class:

- IBMTSSVC_StorageRedundancySet (Extended from CIM_StorageRedundancySet)

Association classes:

- IBMTSSVC_StoragePoolExtentDependency represents the association between IBMTSSVC_DiskDriveExtent and IBMTSSVC_ConcreteStoragePool.
- IBMTSSVC_HostedRedundancySet represents the association between IBMTSSVC_StorageRedundancySet and IBMTSSVC_Cluster.
- IBMTSSVC_MemberOfRedundancySet represents the association between IBMTSSVC_DiskDriveExtent and IBMTSSVC_StorageRedundancySet. The DiskDriveExtents form the redundancy set of a StoragePool.

Disk-drive sparing CIM structure

The disk-drive sparing CIM structure consists primarily of the DiskDriveExtent and the StorageRedundancySet.

The interrelationships of the various components are shown in Figure 11 on page 16.

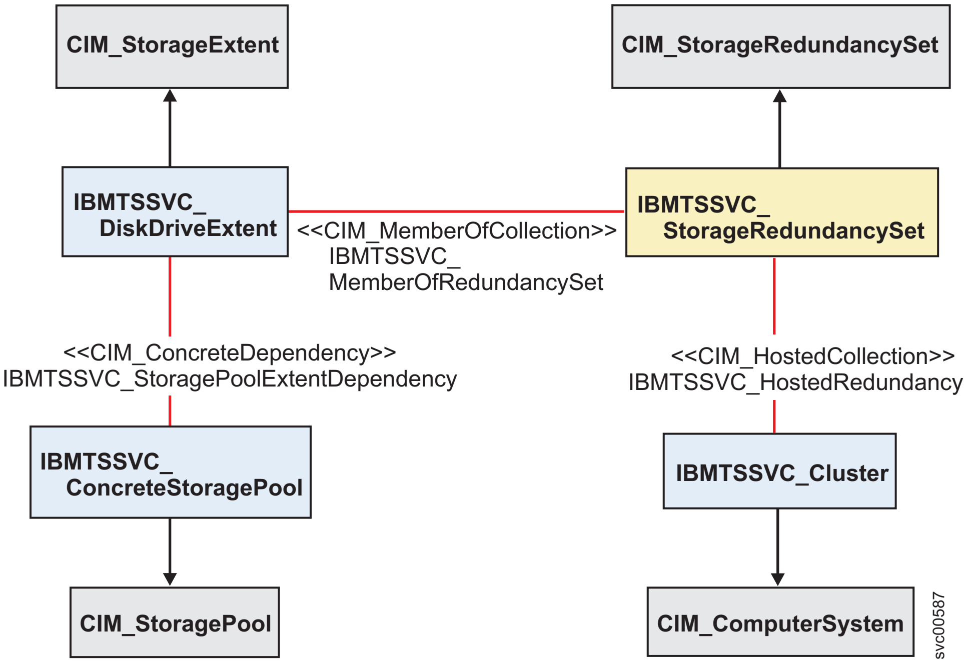

CIM_StorageExtent
CIM_StorageRedundancySet
IBMTSSVC_DiskDriveExtent
<<CIM_MemberOfCollection>> IBMTSSVC_MemberOfRedundancySet
IBMTSSVC_StorageRedundancySet
<<CIM_ConcreteDependency>> IBMTSSVC_StoragePoolExtentDependency
<<CIM_HostedCollection>> IBMTSSVC_HostedRedundancySet
IBMTSSVC_ConcreteStoragePool
IBMTSSVC_Cluster
CIM_StoragePool
CIM_ComputerSystem
svc00587

Figure 11. Disk-drive sparing CIM structure

Spare disk-drive extents

The DiskDriveExtents are used with a failed realization.

The DiskDriveExtents used to replace a DiskDriveExtent whose realization has failed are associated to the RedundancySet using an IsSpare association, as shown in Figure 12.

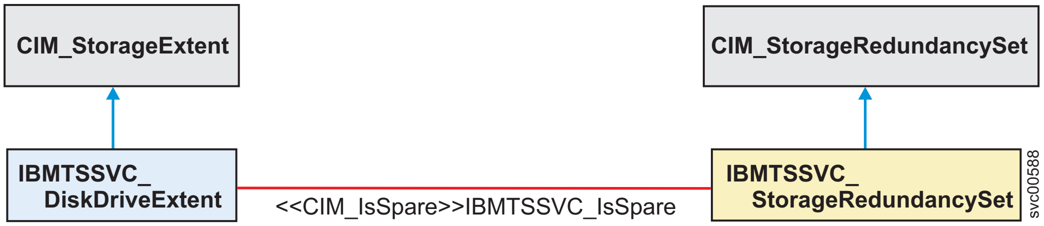


Figure 12. IsSpare association

Class IBMTSSVC_IsSpare represents the association between the IBMTSSVC_DiskDriveExtent to be used as spares and the IBMTSSVC_StorageRedundancySet.

Disk-drive failure association

A failed DiskDriveExtent can be spared.

The DiskDriveExtents associated can be used to spare or replace other DiskDriveExtents that might fail, as shown in Figure 13 on page 17.

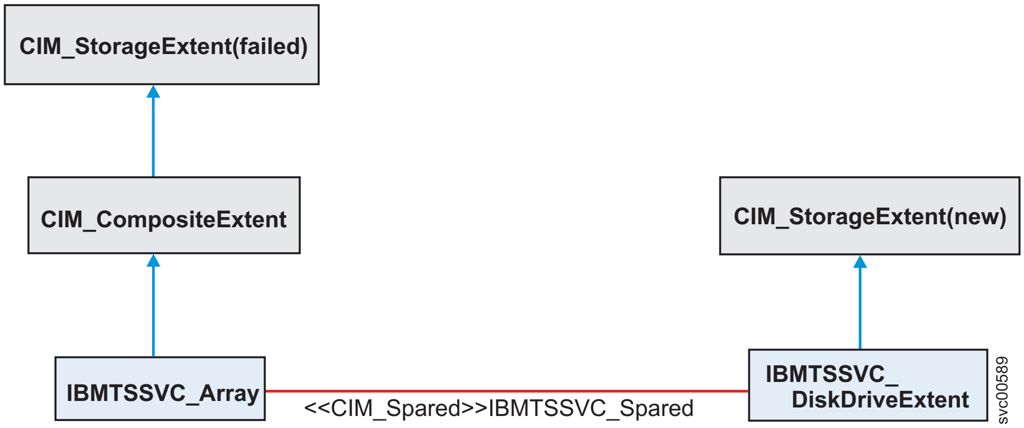


Figure 13. Association during failure

In this relationship between the new DiskDriveExtent and the array being rebuilt, a failed DiskDriveExtent is being spared by the new one. The IBMTSSVC_Spared associations remain until the sparing process is complete.

Enclosure profile

The enclosure is the primary hardware element of the enclosure profile.

Figure 14 shows the basic elements of the enclosure profile.

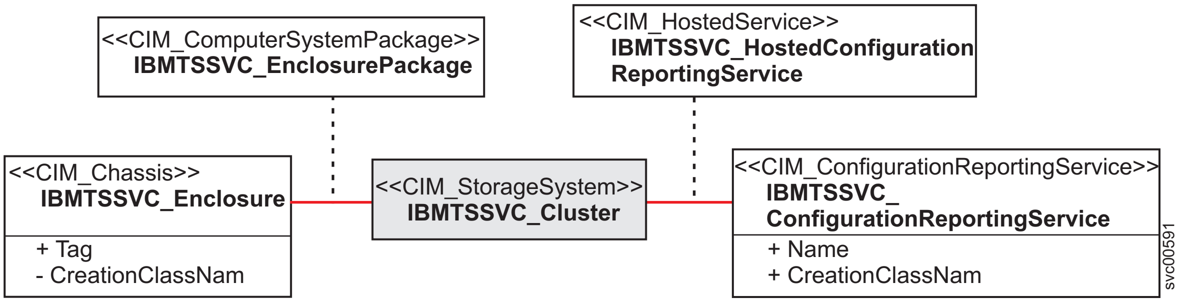


Figure 14. Elements of the enclosure profile

.

Extent composition subprofile

The extent composition subprofile links CompositeExtents and StorageExtents.

The extent composition subprofile allows the block services package to optionally virtualize exposable block storage elements from the underlying primordial storage pool. The abstraction is presented as a representative hierarchy of extents. Figure 15 on page 18 represents the relationships among the disk drive lite, extent composition and array subprofiles.

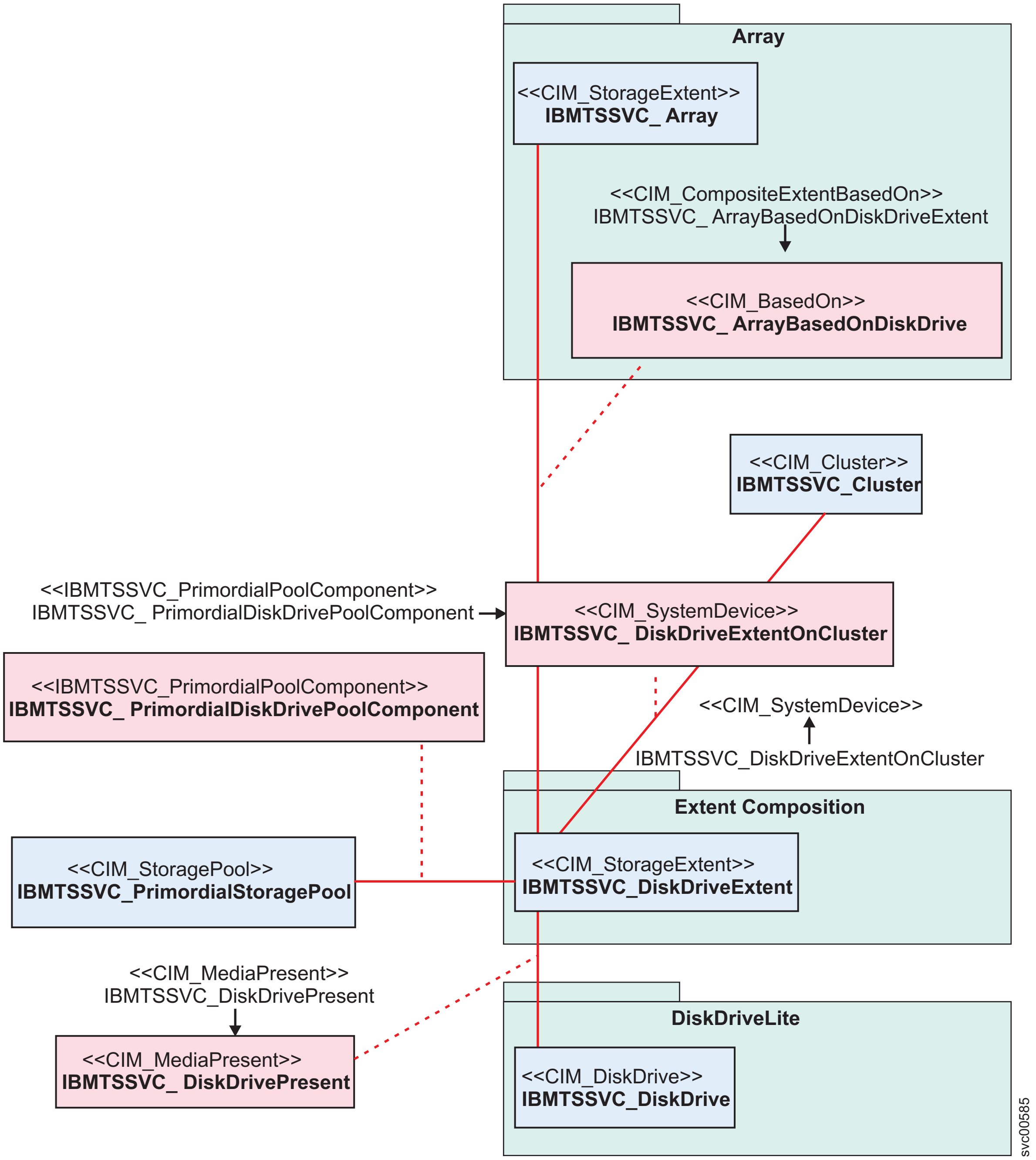

Array
<<CIM_StorageExtent>>
IBMTSSVC_ Array
<<CIM_CompositeExtentBasedOn>>
IBMTSSVC_ ArrayBasedOnDiskDriveExtent
<<CIM_BasedOn>>
IBMTSSVC_ ArrayBasedOnDiskDrive
<<CIM_Cluster>>
IBMTSSVC_Cluster
<<IBMTSSVC_PrimordialPoolComponent>>
IBMTSSVC_ PrimordialDiskDrivePoolComponent
<<CIM_SystemDevice>>
IBMTSSVC_ DiskDriveExtentOnCluster
<<IBMTSSVC_PrimordialPoolComponent>>
IBMTSSVC_ PrimordialDiskDrivePoolComponent
<<CIM_SystemDevice>>
IBMTSSVC_DiskDriveExtentOnCluster
Extent Composition
<<CIM_StoragePool>>
IBMTSSVC_PrimordialStoragePool
<<CIM_StorageExtent>>
IBMTSSVC_DiskDriveExtent
<<CIM_MediaPresent>>
IBMTSSVC_DiskDrivePresent
DiskDriveLite
<<CIM_MediaPresent>>
IBMTSSVC_ DiskDrivePresent
<<CIM_DiskDrive>>
IBMTSSVC_DiskDrive
svc00585

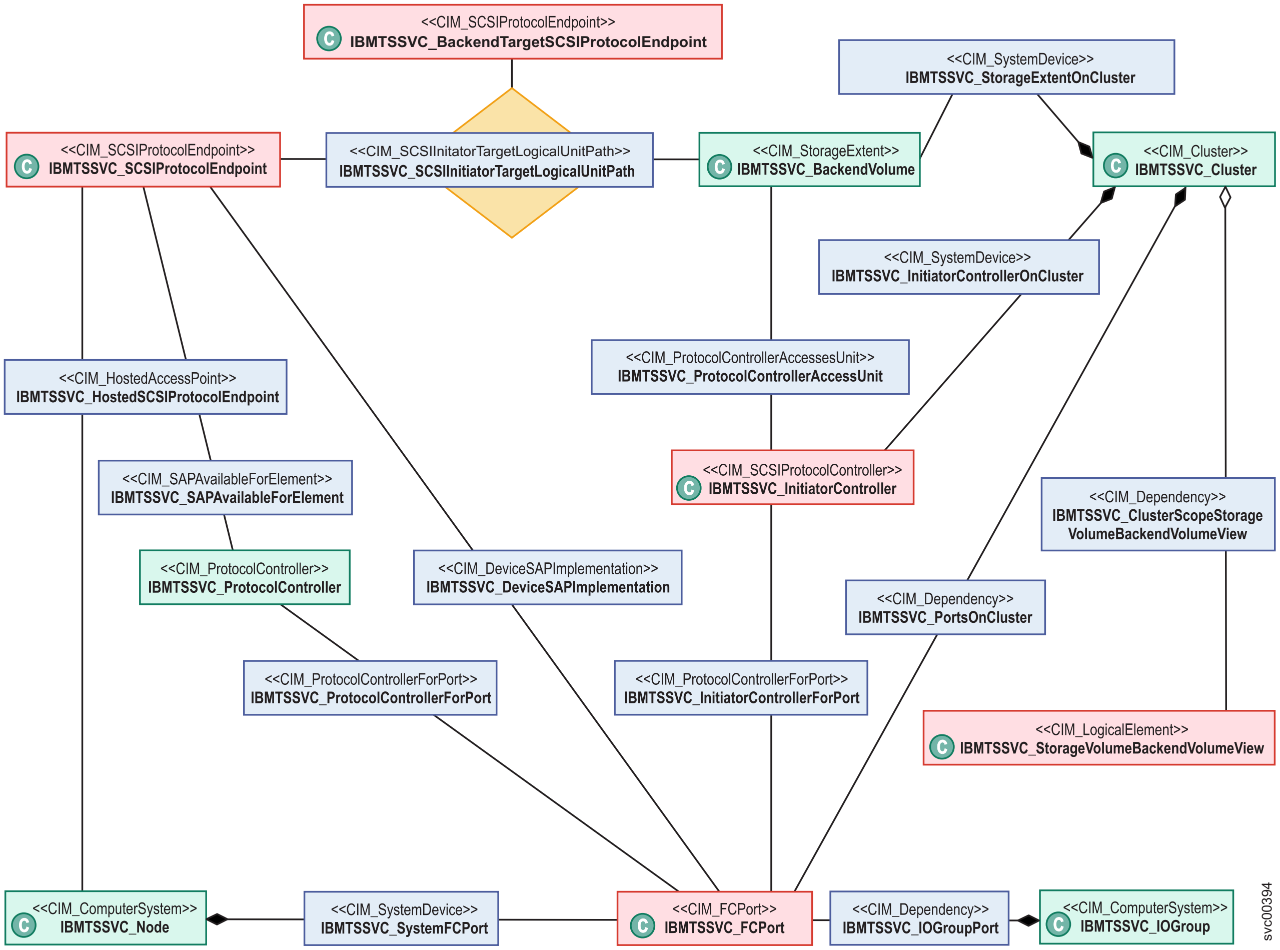
Figure 15. Extent composition subprofile

These extents are instances of CompositeExtents and StorageExtents linked by a combination of CompositeExtentBasedOn and BasedOn associations. In Figure 15, the blue blocks represent classes and the red blocks represent associations.

FC port profile

The FC port profile models the fibre-channel connection relationship between the Storwize V7000 and the backend storage that the Storwize V7000 virtualizes.

Figure 16 on page 19 shows the classes and associations for the FC port profile.



<<CIM_SCSIProtocolEndpoint>>
C IBMTSSVC_BackendTargetSCSIProtocolEndpoint
<<CIM_SystemDevice>>
IBMTSSVC_StorageExtentOnCluster
<<CIM_SCSIProtocolEndpoint>>
C IBMTSSVC_SCSIProtocolEndpoint
<<CIM_SCSIInitatorTargetLogicalUnitPath>>
IBMTSSVC_SCSIInitiatorTargetLogicalUnitPath
<<CIM_StorageExtent>>
C IBMTSSVC_BackendVolume
<<CIM_Cluster>>
C IBMTSSVC_Cluster
<<CIM_SystemDevice>>
IBMTSSVC_InitiatorControllerOnCluster
<<CIM_ProtocolControllerAccessesUnit>>
IBMTSSVC_ProtocolControllerAccessUnit
<<CIM_HostedAccessPoint>>
IBMTSSVC_HostedSCSIProtocolEndpoint
<<CIM_SCSIProtocolController>>
C IBMTSSVC_InitiatorController
<<CIM_SAPAvailableForElement>>
IBMTSSVC_SAPAvailableForElement
<<CIM_Dependency>>
IBMTSSVC_ClusterScopeStorage VolumeBackendVolumeView
<<CIM_ProtocolController>>
IBMTSSVC_ProtocolController
<<CIM_DeviceSAPImplementation>>
IBMTSSVC_DeviceSAPImplementation
<<CIM_Dependency>>
IBMTSSVC_PortsOnCluster
<<CIM_ProtocolControllerForPort>>
IBMTSSVC_ProtocolControllerForPort
<<CIM_ProtocolControllerForPort>>
IBMTSSVC_InitiatorControllerForPort
<<CIM_LogicalElement>>
C IBMTSSVC_StorageVolumeBackendVolumeView
<<CIM_ComputerSystem>>
C IBMTSSVC_Node
<<CIM_SystemDevice>>
IBMTSSVC_SystemFCPort
<<CIM_FCPort>>
C IBMTSSVC_FCPort
<<CIM_Dependency>>
IBMTSSVC_IOGroupPort
<<CIM_ComputerSystem>>
C IBMTSSVC_IOGroup
svc00394

Figure 16. Class diagram of FC port instances

iSCSI relationships

The iSCSI-specific aspects of a target device include instance and association classes.

Figure 17 on page 20 shows the iSCSI-specific aspects of a target device.

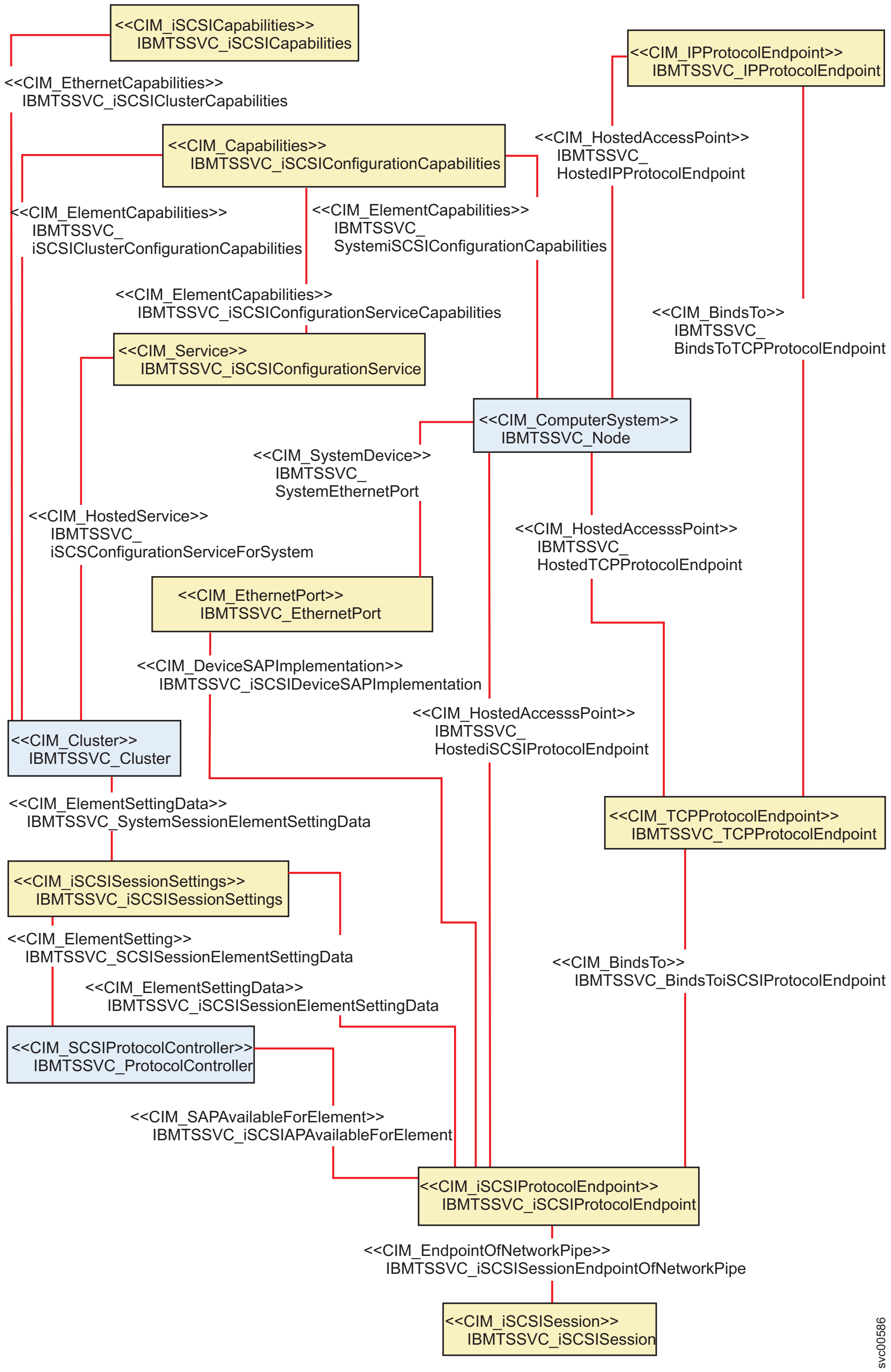

<<CIM_iSCSICapabilities>>
IBMTSSVC_iSCSICapabilities
<<CIM_IPProtocolEndpoint>>
IBMTSSVC_IPProtocolEndpoint
<<CIM_EthernetCapabilities>>
IBMTSSVC_iSCSIClusterCapabilities
<<CIM_Capabilities>>
IBMTSSVC_iSCSIConfigurationCapabilities
<<CIM_HostedAccessPoint>>
IBMTSSVC_
HostedIPProtocolEndpoint
<<CIM_ElementCapabilities>>
IBMTSSVC_
iSCSIClusterConfigurationCapabilities
<<CIM_ElementCapabilities>>
IBMTSSVC_
SystemiSCSIConfigurationCapabilities
<<CIM_ElementCapabilities>>
IBMTSSVC_iSCSIConfigurationServiceCapabilities
<<CIM_BindsTo>>
IBMTSSVC_
BindsToTCPProtocolEndpoint
<<CIM_Service>>
IBMTSSVC_iSCSIConfigurationService
<<CIM_ComputerSystem>>
IBMTSSVC_Node
<<CIM_SystemDevice>>
IBMTSSVC_
SystemEthernetPort
<<CIM_HostedService>>
IBMTSSVC_
iSCSConfigurationServiceForSystem
<<CIM_HostedAccesssPoint>>
IBMTSSVC_
HostedTCPProtocolEndpoint
<<CIM_EthernetPort>>
IBMTSSVC_EthernetPort
<<CIM_DeviceSAPImplementation>>
IBMTSSVC_iSCSIDeviceSAPImplementation
<<CIM_HostedAccesssPoint>>
IBMTSSVC_
HostediSCSIProtocolEndpoint
<<CIM_Cluster>>
IBMTSSVC_Cluster
<<CIM_ElementSettingData>>
IBMTSSVC_SystemSessionElementSettingData
<<CIM_TCPProtocolEndpoint>>
IBMTSSVC_TCPProtocolEndpoint
<<CIM_iSCSISessionSettings>>
IBMTSSVC_iSCSISessionSettings
<<CIM_ElementSetting>>
IBMTSSVC_SCSISessionElementSettingData
<<CIM_BindsTo>>
IBMTSSVC_BindsToiSCSIProtocolEndpoint
<<CIM_ElementSettingData>>
IBMTSSVC_iSCSISessionElementSettingData
<<CIM_SCSIProtocolController>>
IBMTSSVC_ProtocolController
<<CIM_SAPAvailableForElement>>
IBMTSSVC_iSCSIAPAvailableForElement
<<CIM_iSCSIProtocolEndpoint>>
IBMTSSVC_iSCSIProtocolEndpoint
<<CIM_EndpointOfNetworkPipe>>
IBMTSSVC_iSCSISessionEndpointOfNetworkPipe
<<CIM_iSCSISession>>
IBMTSSVC_iSCSISession
svc00586

Figure 17. Overview of iSCSI-specific aspects of a target device

In Figure 17, the classes in yellow and relevant associations are now supported.

.

Job control profile

The job control profile contains classes that allow you to monitor asynchronous commands that format, migrate, or run copy operations on a device.

Figure 18 shows the classes and associations for the job control profile.

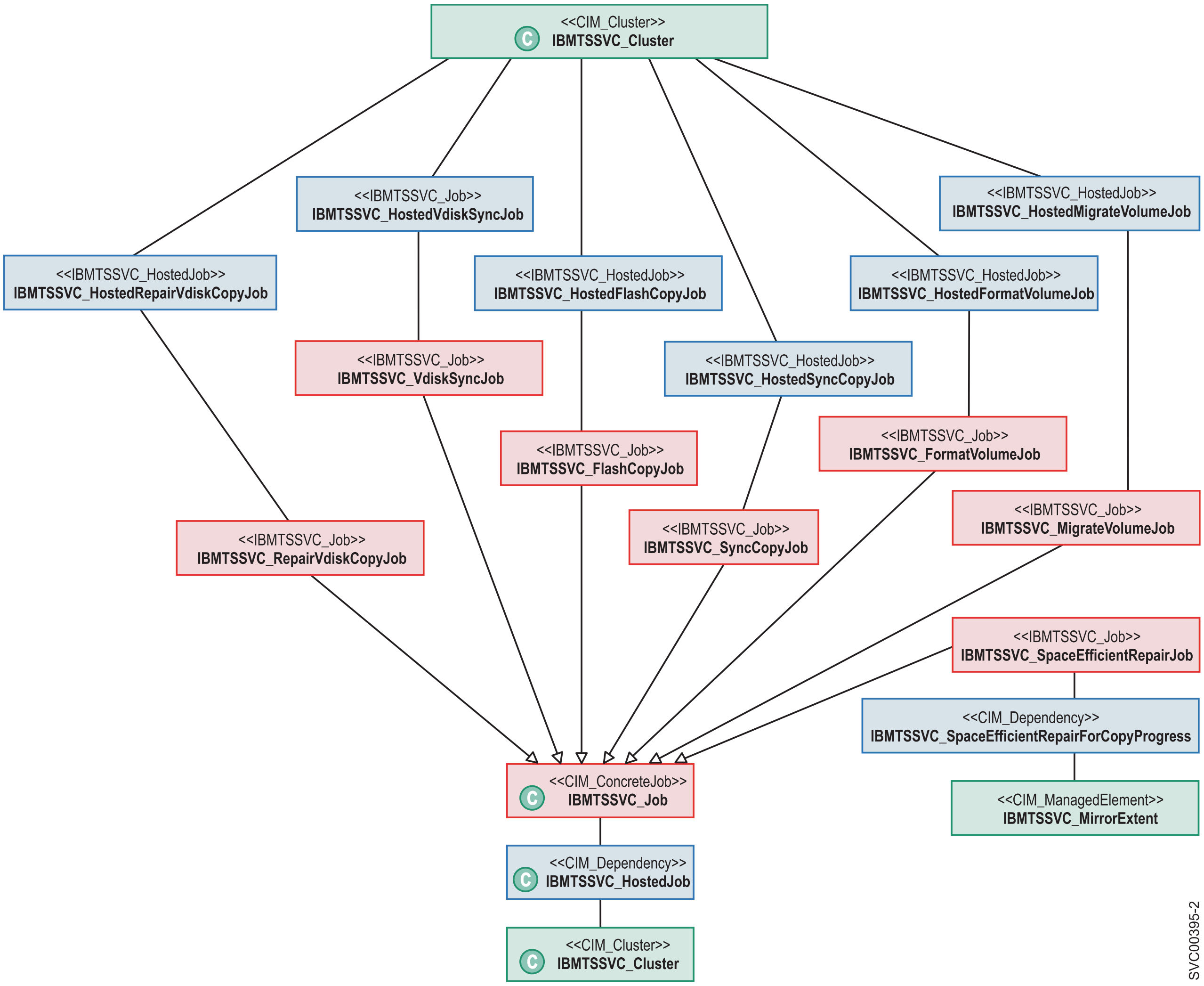


Figure 18. Class diagram of job control instances

Location subprofile

The location subprofile represents the relationship between location and physical package.

Figure 19 on page 22 shows the basic elements of the location subprofile.

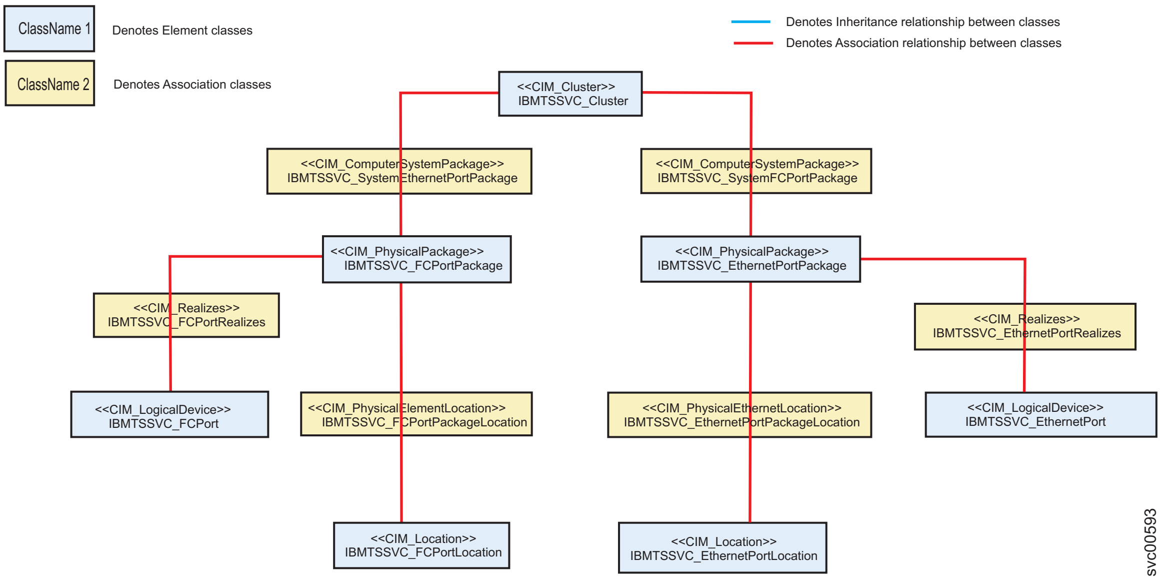

ClassName 1
Denotes Element classes
ClassName 2
Denotes Association classes
Denotes Inheritance relationship between classes
Denotes Association relationship between classes
<<CIM_Cluster>> IBMTSSVC_Cluster
<<CIM_ComputerSystemPackage>> IBMTSSVC_SystemEthernetPortPackage
<<CIM_ComputerSystemPackage>> IBMTSSVC_SystemFCPortPackage
<<CIM_PhysicalPackage>> IBMTSSVC_FCPortPackage
<<CIM_PhysicalPackage>> IBMTSSVC_EthernetPortPackage
<<CIM_Realizes>> IBMTSSVC_FCPortRealizes
<<CIM_Realizes>> IBMTSSVC_EthernetPortRealizes
<<CIM_LogicalDevice>> IBMTSSVC_FCPort
<<CIM_PhysicalElementLocation>> IBMTSSVC_FCPortPackageLocation
<<CIM_PhysicalEthernetLocation>> IBMTSSVC_EthernetPortPackageLocation
<<CIM_LogicalDevice>> IBMTSSVC_EthernetPort
<<CIM_Location>> IBMTSSVC_FCPortLocation
<<CIM_Location>> IBMTSSVC_EthernetPortLocation
svc00593

Figure 19. Overview of the relationship between location and physical package

In Figure 19, the class in yellow blocks denotes the association class. For example, IBMTSSVC_FCPortRealizes is the association between IBMTSSVC_FCPort and IBMTSSVC_FCPortPackage. The class in blue blocks denote the element class.

.

Masking and mapping profile

The masking and mapping profile provides an interface to create, modify, delete, and mask hosts.

Figure 20 on page 23 shows the classes and associations for the masking and mapping profile.

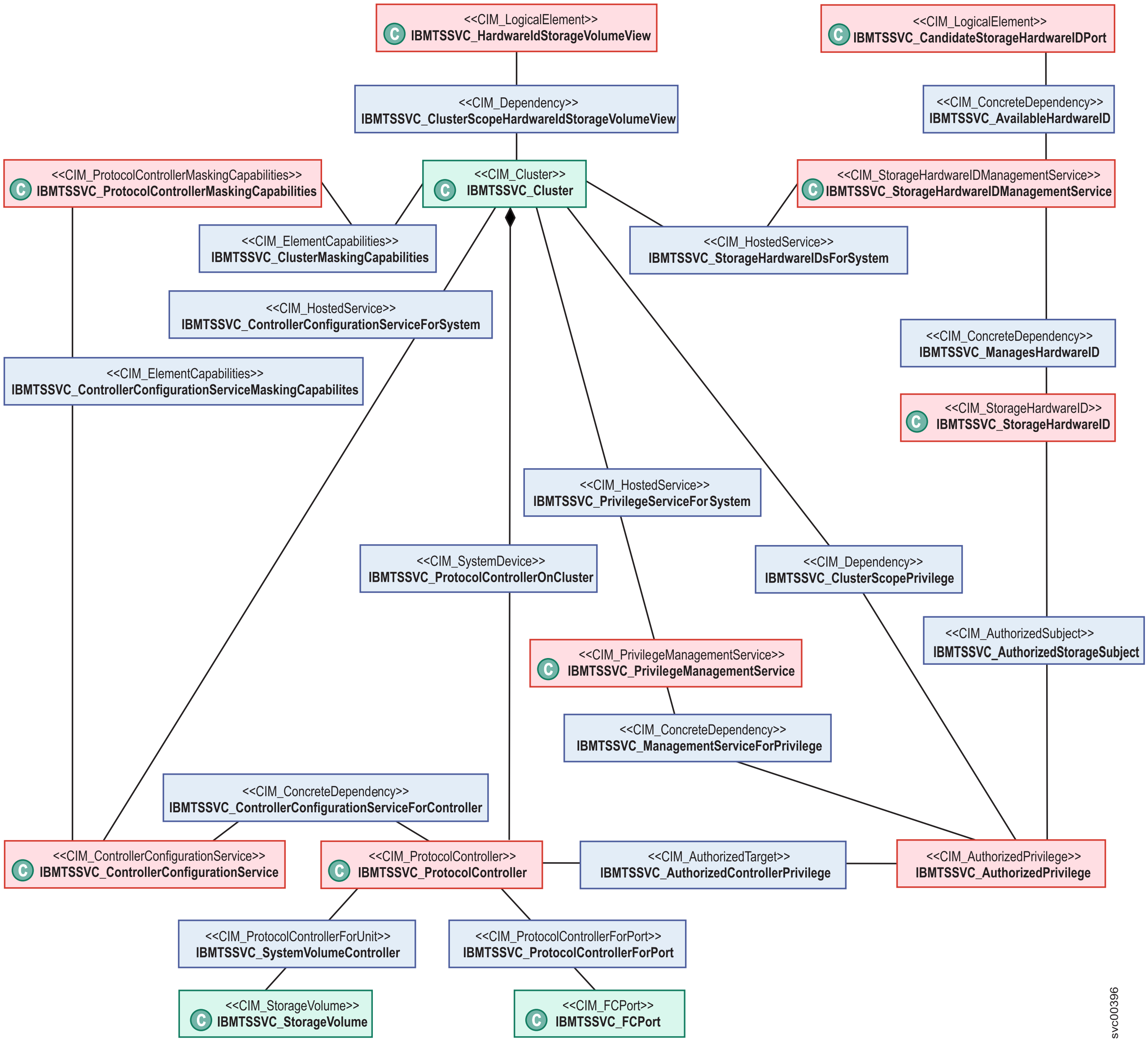

<<CIM_LogicalElement>>
IBMTSSVC_HardwareIdStorageVolumeView
<<CIM_LogicalElement>>
IBMTSSVC_CandidateStorageHardwareIDPort
<<CIM_Dependency>>
IBMTSSVC_ClusterScopeHardwareIdStorageVolumeView
<<CIM_ConcreteDependency>>
IBMTSSVC_AvailableHardwareID
<<CIM_ProtocolControllerMaskingCapabilities>>
IBMTSSVC_ProtocolControllerMaskingCapabilities
<<CIM_Cluster>>
IBMTSSVC_Cluster
<<CIM_StorageHardwareIDManagementService>>
IBMTSSVC_StorageHardwareIDManagementService
<<CIM_ElementCapabilities>>
IBMTSSVC_ClusterMaskingCapabilities
<<CIM_HostedService>>
IBMTSSVC_StorageHardwareIDsForSystem
<<CIM_HostedService>>
IBMTSSVC_ControllerConfigurationServiceForSystem
<<CIM_ConcreteDependency>>
IBMTSSVC_ManagesHardwareID
<<CIM_ElementCapabilities>>
IBMTSSVC_ControllerConfigurationServiceMaskingCapabilites
<<CIM_StorageHardwareID>>
IBMTSSVC_StorageHardwareID
<<CIM_HostedService>>
IBMTSSVC_PrivilegeServiceFor System
<<CIM_SystemDevice>>
IBMTSSVC_ProtocolControllerOnCluster
<<CIM_Dependency>>
IBMTSSVC_ClusterScopePrivilege
<<CIM_AuthorizedSubject>>
IBMTSSVC_AuthorizedStorageSubject
<<CIM_PrivilegeManagementService>>
IBMTSSVC_PrivilegeManagementService
<<CIM_ConcreteDependency>>
IBMTSSVC_ManagementServiceForPrivilege
<<CIM_ConcreteDependency>>
IBMTSSVC_ControllerConfigurationServiceForController
<<CIM_ControllerConfigurationService>>
IBMTSSVC_ControllerConfigurationService
<<CIM_ProtocolController>>
IBMTSSVC_ProtocolController
<<CIM_AuthorizedTarget>>
IBMTSSVC_AuthorizedControllerPrivilege
<<CIM_AuthorizedPrivilege>>
IBMTSSVC_AuthorizedPrivilege
<<CIM_ProtocolControllerForUnit>>
IBMTSSVC_SystemVolumeController
<<CIM_ProtocolControllerForPort>>
IBMTSSVC_ProtocolControllerForPort
<<CIM_StorageVolume>>
IBMTSSVC_StorageVolume
<<CIM_FCPort>>
IBMTSSVC_FCPort
svc00396

Figure 20. Class diagram of masking and mapping instances

Multiple computer system profile

The multiple computer system profile utilizes multiple systems to present a virtual computer system.

Figure 21 on page 24 shows the classes and associations for the multiple computer system profile.

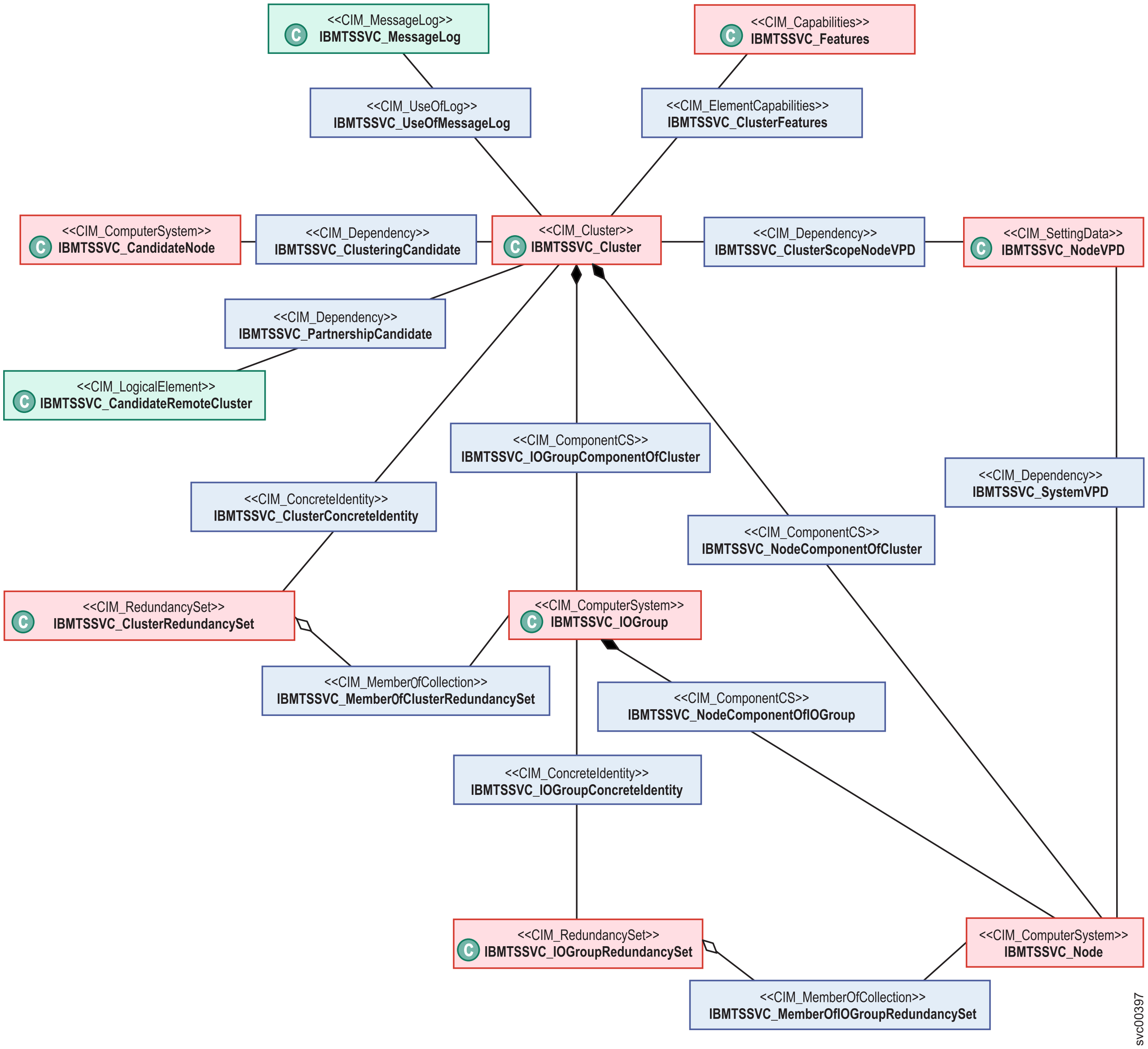


Figure 21. Class diagram of multiple computer system instances

Object manager adapter profile

The primary function of the object manager adapter profile is to turn the indication service on or off.

Figure 22 on page 25 shows the basic elements of the Object Manager Adapter profile.

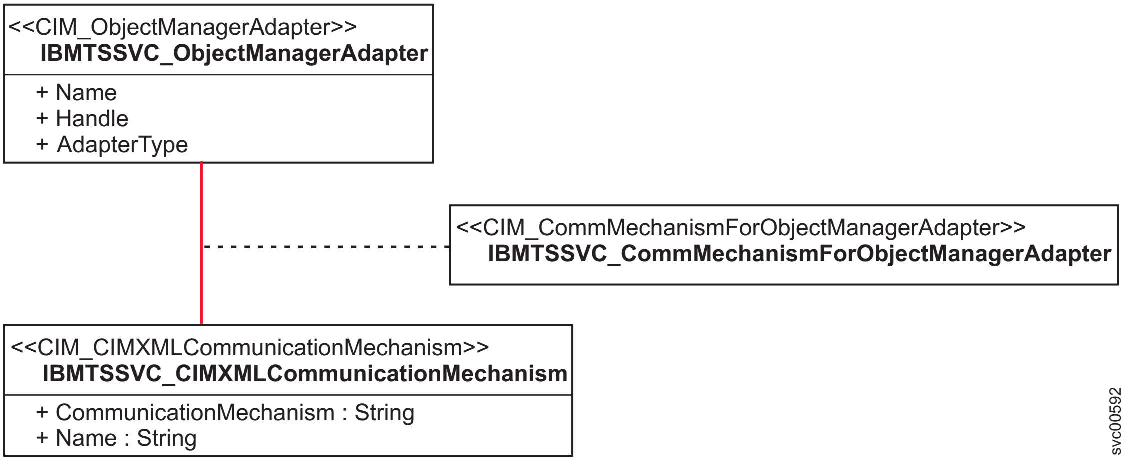

<<CIM_ObjectManagerAdapter>>
IBMTSSVC_ObjectManagerAdapter
+ Name
+ Handle
+ AdapterType
<<CIM_CommMechanismForObjectManagerAdapter>>
IBMTSSVC_CommMechanismForObjectManagerAdapter
<<CIM_CIMXMLCommunicationMechanism>>
IBMTSSVC_CIMXMLCommunicationMechanism
+ CommunicationMechanism : String
+ Name : String
svc00592

Figure 22. Overview of the Unified Modeling Language elements of the object manager adapter profile

.

Physical package

The physical package of the Common Information Model (CIM) agent for the Storwize V7000 consists of two classes and two association classes.

Figure 23 shows the basic classes (building blocks) for the model.

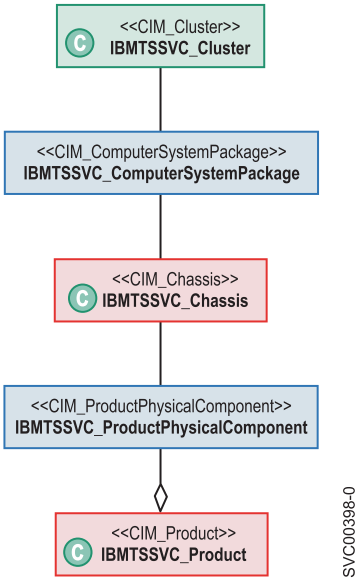

<<CIM_Cluster>>
C
IBMTSSVC_Cluster
<<CIM_ComputerSystemPackage>>
IBMTSSVC_ComputerSystemPackage
<<CIM_Chassis>>
C
IBMTSSVC_Chassis
<<CIM_ProductPhysicalComponent>>
IBMTSSVC_ProductPhysicalComponent
<<CIM_Product>>
C
IBMTSSVC_Product
SVC00398-0

Figure 23. High-level overview of the physical package of the CIM agent for the Storwize V7000.

Server profile

The server profile of the Common Information Model (CIM) agent for the Storwize V7000 consists of several basic classes.

Figure 24 on page 26 shows the basic classes (building blocks) for the model.

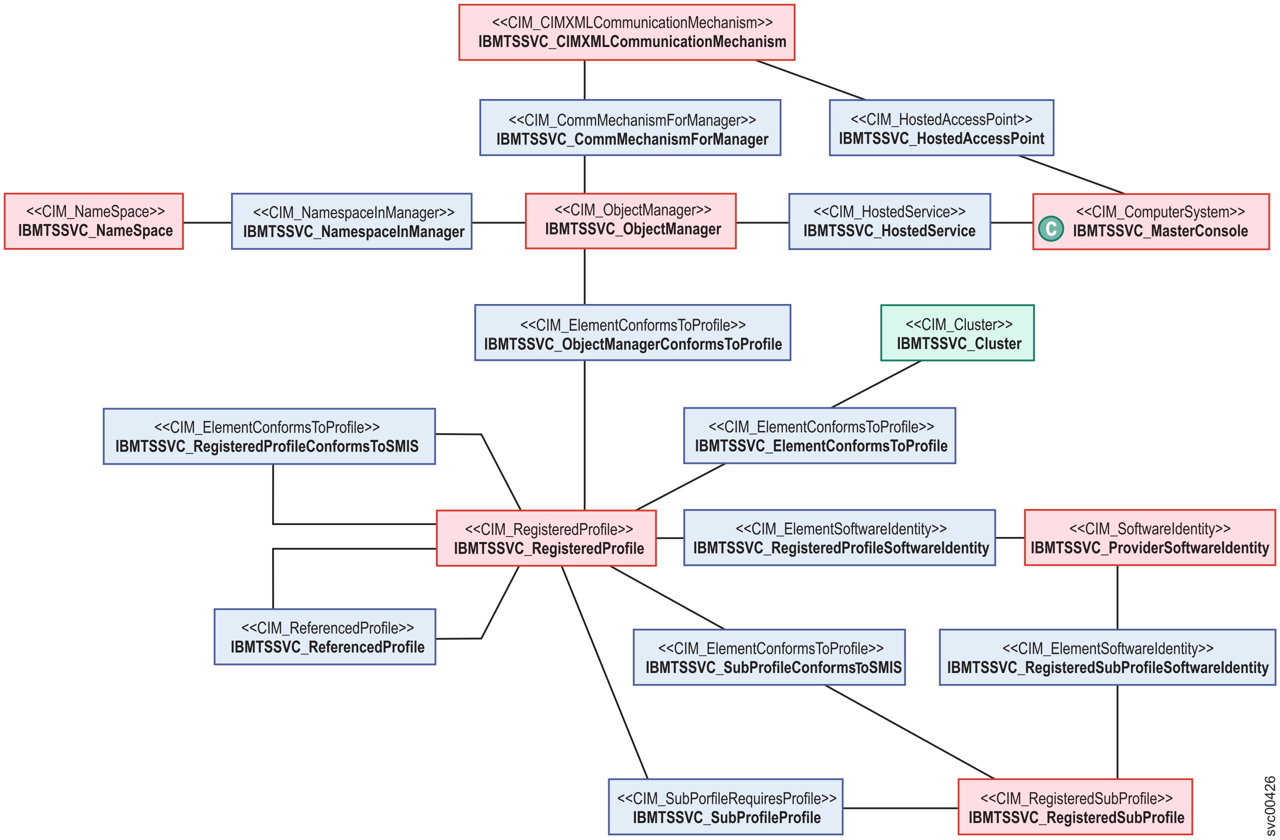

<<CIM_CIMXMLCommunicationMechanism>>
IBMTSSVC_CIMXMLCommunicationMechanism
<<CIM_CommMechanismForManager>>
IBMTSSVC_CommMechanismForManager
<<CIM_HostedAccessPoint>>
IBMTSSVC_HostedAccessPoint
<<CIM_NameSpace>>
IBMTSSVC_NameSpace
<<CIM_NamespaceInManager>>
IBMTSSVC_NamespaceInManager
<<CIM_ObjectManager>>
IBMTSSVC_ObjectManager
<<CIM_HostedService>>
IBMTSSVC_HostedService
<<CIM_ComputerSystem>>
C
IBMTSSVC_MasterConsole
<<CIM_ElementConformsToProfile>>
IBMTSSVC_ObjectManagerConformsToProfile
<<CIM_Cluster>>
IBMTSSVC_Cluster
<<CIM_ElementConformsToProfile>>
IBMTSSVC_RegisteredProfileConformsToSMIS
<<CIM_ElementConformsToProfile>>
IBMTSSVC_ElementConformsToProfile
<<CIM_RegisteredProfile>>
IBMTSSVC_RegisteredProfile
<<CIM_ElementSoftwareIdentity>>
IBMTSSVC_RegisteredProfileSoftwareIdentity
<<CIM_SoftwareIdentity>>
IBMTSSVC_ProviderSoftwareIdentity
<<CIM_ReferencedProfile>>
IBMTSSVC_ReferencedProfile
<<CIM_ElementConformsToProfile>>
IBMTSSVC_SubProfileConformsToSMIS
<<CIM_ElementSoftwareIdentity>>
IBMTSSVC_RegisteredSubProfileSoftwareIdentity
<<CIM_SubPorfileRequiresProfile>>
IBMTSSVC_SubProfileProfile
<<CIM_RegisteredSubProfile>>
IBMTSSVC_RegisteredSubProfile
svc00426

Figure 24. High-level overview of the server profile of the CIM agent for the Storwize V7000.

Software profile

The software profile allows the CIM agent to model the software for the Storwize V7000 cluster and for the CIM agent.

Figure 25 shows the classes and associations for the software profile.

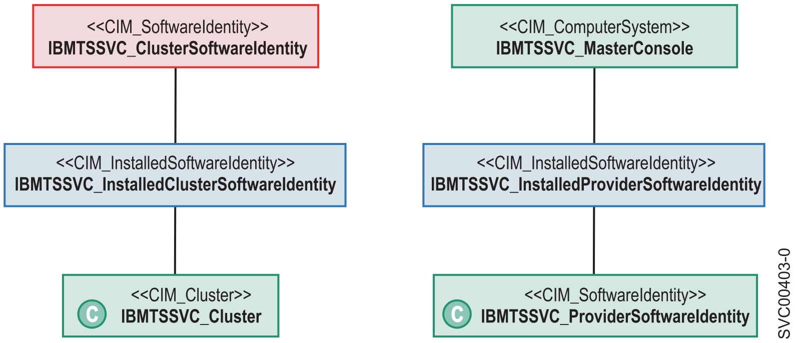

<<CIM_SoftwareIdentity>>
IBMTSSVC_ClusterSoftwareIdentity
<<CIM_ComputerSystem>>
IBMTSSVC_MasterConsole
<<CIM_InstalledSoftwareIdentity>>
IBMTSSVC_InstalledClusterSoftwareIdentity
<<CIM_InstalledSoftwareIdentity>>
IBMTSSVC_InstalledProviderSoftwareIdentity
<<CIM_Cluster>>
C
IBMTSSVC_Cluster
<<CIM_SoftwareIdentity>>
C
IBMTSSVC_ProviderSoftwareIdentity
SVC00403-0

Figure 25. Class diagram of software instances

Storage volume mirroring

You can create two mirrored copies of a storage volume to increase data availability. The volume remains online and available as long as one copy is available.

Figure 26 shows the classes and associations that are involved in mirroring storage volumes.

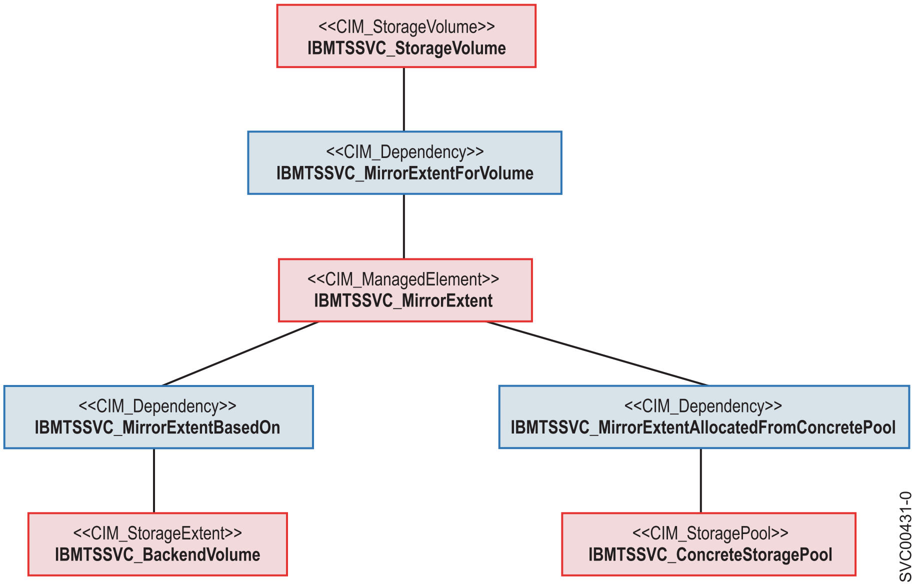


Figure 26. Classes and associations in mirroring storage volumes

Chapter 2. Performing storage configuration tasks

Storage configuration is the mapping of the back-end storage to the storage pools and the allocation of volumes from those pools. After you complete the initial setup of the Storwize V7000, you use the Common Information Model (CIM) agent object class instances to complete basic storage configuration tasks.

In the Common Information Model (CIM) agent for the Storwize V7000, storage configuration involves three layers of objects: back-end, middle and front-end. The objects in the *back-end* layer contain the back-end controllers and volumes, those in the *middle* layer contain the storage pools, and those in the *front-end* layer contain the storage volumes that are exposed to the hosts.

The IBMTSSVC_StorageConfigurationService class provides the CreateOrModifyStoragePool() and CreateOrModifyElementFromStoragePool() methods for performing basic storage configuration.

You can use the CreateOrModifyStoragePool() method to create an IBMTSSVC_StoragePool and add or remove an IBMTSSVC_BackendVolume. You can use the CreateOrModifyElementFromStoragePool() method to allocate, expand, or shrink an IBMTSSVC_StorageVolume from an IBMTSSVC_ConcreteStoragePool.

To complete basic storage configuration, perform the following tasks:

1. Add a node to the cluster.
2. Create a storage pool.
3. Modify the storage pool.
4. Create a storage volume.
5. Modify a storage volume.

Creating a new storage pool

The IBMTSSVC_StorageConfigurationService class provides the methods for creating a new IBMTSSVC_ConcreteStoragePool.

To create a new IBMTSSVC_ConcreteStoragePool instance, follow these steps:

1. Obtain the reference (CIMObjectPath) of an IBMTSSVC_StorageConfigurationService instance that is associated with the IBMTSSVC_Cluster in which you will create the new storage pool using one of the following two methods:
   - Obtain the reference of the IBMTSSVC_StorageConfigurationService instance directly. There will be only one instance obtained by enumeration because CIMOM is embedded in cluster.
   - Obtain the reference of the IBMTSSVC_StorageConfigurationService instance by traversing the IBMTSSVC_HostedStorageConfigurationService association.
2. Invoke the IBMTSSVC_StorageConfigurationService.CreateOrModifyStoragePool method while you specify the InExtents[] parameter with a list of IBMTSSVC_BackendVolume instances.

   The InExtents[] parameter is a string array that contains the representation of the CIMObjectPath to an IBMTSSVC_BackendVolume.

   You can also just specify the size, and the CIM agent performs a best-fit heuristic to match it. You can specify the name of the pool by using the ElementName parameter and the block size by using the BlockSize parameter.

Modifying a storage pool

You can modify an IBMTSSVC_ConcreteStoragePool instance by changing the pool name and adding or removing an IBMTSSVC_BackendVolume instance from the pool.

To modify an IBMTSSVC_ConcreteStoragePool instance, follow these steps:

1. Select the IBMTSSVC_ConcreteStoragePool instance that you want to modify from an IBMTSSVC_Cluster.
2. Identify the IBMTSSVC_StoragePoolSetting instance that contains the parameter settings of the IBMTSSVC_ConcreteStoragePool Setting instance. You can do this by calling the IBMTSSVC_StorageCapabilities.CreateSetting() method or by enumerating the IBMTSSVC_StoragePoolSetting that is associated through IBMTSSVC_StorageSettingsGeneratedFromCapabilities to the IBMTSSVC_StorageCapabilities that is associated to the IBMTSSVC_ConcreteStoragePool that is being modified.
3. Modify the name of the selected IBMTSSVC_ConcreteStoragePool by calling IBMTSSVC_StorageConfigurationService.CreateOrModifyPool with a new element name.
4. If necessary, you can further modify the IBMTSSVC_ConcreteStoragePool by adding or removing an IBMTSSVC_BackendVolume instance to the pool.

Creating a new storage volume

In the Common Information Model (CIM) agent for the Storwize V7000, the IBMTSSVC_StorageConfigurationService class provides all the methods that are required for creating, modifying, and deleting an IBMTSSVC_StorageVolume instance.

To create a new IBMTSSVC_StorageVolume instance, follow these steps:

Use the CreateOrModifyElementFromStoragePool() method on the IBMTSSVC_Cluster to which you will assign the new volume to create the new IBMTSSVC_StorageVolume with the following parameter specifications:

- The Virtualization Type is set using the VirtualizationType parameter (0,1,2 for striped, sequential or image).
- A Format flag can be used to specify that the volume is formatted on creation or expansion.
- You can specify the BackendVolumes to place the extents for the volume using the BackendVolume REF parameter.
- The PreferredNode parameter is used to set the preferred node for the volume.
- The UnitDeviceID parameter can be used to set the Volume's Unit Device ID on clusters that run software level 4.1.0 or higher.
- The ElementName parameter can be used to set the Volumes Name at creation.
- The Autoexpand parameter

1. Set ElementType to 2.
2. Set Size to the preferred volume size in bytes.
3. Obtain the reference (CIMObjectPath) of the IBMTSSVC_ConcreteStoragePool instance from which you will allocate an IBMTSSVC_StorageVolume.
4. Set InPool to the reference (obtained in the previous step) of the pool from which the volume will be allocated.

The following additional parameters are examples of those that are available as part of the Volume Mirroring feature or Thin-Provisioned Volume feature.

Autoexpand

Used to attempt to maintain a fixed amount of unused real capacity on the volume, which is called the *contingency capacity*. The capacity is initially set to the *real capacity*, which is assigned when the volume is created. If you modify the real capacity, the contingency capacity is reset to be the difference between the *used capacity* and the *real capacity*.

Copies
Indicates the number of mirrored copies to create. Specify either 1 or 2.

Grainsize
Sets the grain size of a thin-provisioned volume. This is valid only if RealSize is set. Valid options include 32, 64, 128, or 256.

Import
Used to request the Storwize V7000 to import a thin-provisioned storage volume from the storage pool.

IsSpaceEfficient
Flag that is used to indicate that the created volumes are thin-provisioned volumes. This does not apply to a modify operation.

RealSizeBytes and RealSizePercent
Indicates the real size of a thin-provisioned copy. These parameters cannot be used together.

WarningSizeBytes
Indicates the set point of used capacity at which point a warning is triggered. This parameter cannot be used with the WarningSizeBytes parameter parameter.

WarningSizePercent
Indicates the ratio of used capacity versus the volume virtual capacity that triggers a warning. This parameter cannot be used with the WarningSizeBytes parameter.

Creating a RAID array

You can create a RAID array using local drives.

To create a RAID array, follow these steps:

1. Obtain the reference (CIMObjectPath) of an IBMTSSVC_StorageConfigurationService instance that is associated with the IBMTSSVC_Cluster in which you will create the new RAID array using one of the following two methods:
   - Obtain the reference of the IBMTSSVC_StorageConfigurationService instance directly. There will be only one instance obtained by enumeration because CIMOM is embedded in cluster.
   - Obtain the reference of the IBMTSSVC_StorageConfigurationService instance by traversing the IBMTSSVC_HostedStorageConfigurationService association.
2. Invoke the IBMTSSVC_StorageConfigurationService CreateOrModifyElementFromElements method.

   You must specify the following parameters:
   - The InElements [] parameter with a list of IBMTSSVC_DiskDriveExtent instances. The Extent [] parameter is a string array that contains the representation of the CIMObjectPath to an IBMTSSVC_DiskDriveExtent.
   - StoragePool parameter with an instance of IBMTSSVC_ConcreteStoragePool, which identifies the StoragePool to add the created RAID array to.
   - Goal parameter with an instance of IBMTSSVC_RAIDSetting, which contains the RAID-level information.

   You can also specify the StripSize, WriteVerify, and SpareGoal. Also see the `BlockServices.mof` file for more details. To view the complete Managed Object Format (MOF) documentation, select the documentation information from the following website:

   Support for Storwize V7000 website at www.ibm.com/storage/support/storwize/v7000

Chapter 3. Establishing Copy Services relationships

You can use the Common Information Model agent object class instances to establish new Copy Services relationships.

Copy Services

FlashCopy, Global Mirror, and Metro Mirror are Copy Services that are provided by the Storwize V7000.

These Copy Services are available to all supported hosts that are connected to the Storwize V7000.

The FlashCopy service enables you to make an instant, point-in-time copy of a source IBMTSSVC_StorageVolume instance to a target IBMTSSVC_StorageVolume instance.

The synchronous copy service (Metro Mirror) provides a consistent copy of the source IBMTSSVC_StorageVolume on the target IBMTSSVC_StorageVolume. Data is written to the target volume synchronously after it is written to the source volume, both of which can belong to the same IBMTSSVC_Cluster instance or different IBMTSSVC_Cluster instances.

The asynchronous copy service (Global Mirror) provides a copy of the source IBMTSSVC_StorageVolume on the target IBMTSSVC_StorageVolume. Data is written to the target volume asynchronously after it is written to the source volume, both of which can belong to the same IBMTSSVC_Cluster instance or different IBMTSSVC_Cluster instances.

Creating a new FlashCopy relationship between storage volumes

The IBMTSSVC_StorageConfigurationService class provides the methods for establishing a FlashCopy relationship between two IBMTSSVC_StorageVolume instances that are the same size and belong to the same IBMTSSVC_Cluster instance.

To create a FlashCopy relationship between two IBMTSSVC_StorageVolume instances, perform the following steps:

1. Select an IBMTSSVC_StorageVolume instance as the source volume for the preferred FlashCopy relationship.
2. Select a valid IBMTSSVC_StorageVolume as a target.

   Make sure that the source IBMTSSVC_StorageVolume and target IBMTSSVC_StorageVolume instances belong to the same IBMTSSVC_Cluster instance.
3. Retrieve the IBMTSSVC_StorageConfigurationService instance that is associated with the IBMTSSVC_Cluster instance to which the selected IBMTSSVC_StorageVolume instances belong.
4. Invoke the IBMTSSVC_StorageConfigurationService.AttachReplica() method with the following parameter specifications:
   a. Set SourceElement to the reference (CIMObjectPath) of the source IBMTSSVC_StorageVolume.
   b. Set TargetElement to the reference (CIMObjectPath) of the target IBMTSSVC_StorageVolume.
   c. Optionally set ElementName to the name of the synchronization.
   d. Optionally set BackgroundCopyRate to the preferred priority of the background copy rate (0 - 100).
   e. Optionally specify Set to add the newly created FlashCopySynchronization to the set. If you specify a null value, the newly created FlashCopySynchronization will not be a member of a synchronized set.

f. Optionally set CopyType to 5. This sets AutoDelete to true, which automatically deletes the FlashCopy mapping after the background copy is complete.

g. If you do not want to automatically delete FlashCopy mappings, Set CopyType to 4.

The source IBMTSSVC_StorageVolume and target IBMTSSVC_StorageVolume instances are now connected through the IBMTSSVC_LocalStorageSynchronized association.

Creating a FlashCopy relationship for a synchronized set

The IBMTSSVC_StorageConfigurationService class provides the methods for establishing a FlashCopy relationship between two IBMTSSVC_StorageVolume instances and then adding it to an IBMTSSVC_FlashCopyStorageSynchronizedSet instance.

To create a FlashCopy relationship between two IBMTSSVC_StorageVolume instances and add it to an IBMTSSVC_FlashCopyStorageSynchronizedSet instance, perform the following steps:

1. Select an IBMTSSVC_StorageVolume instance as the source volume for the preferred FlashCopy relationship.
2. Select a valid IBMTSSVC_StorageVolume as a target. Valid volumes can be determined using IBMTSSVC_CandidateVolume.

   Make sure that the source IBMTSSVC_StorageVolume and target IBMTSSVC_StorageVolume instances belong to the same IBMTSSVC_Cluster instance.
3. Retrieve the IBMTSSVC_StorageConfigurationService instance that is associated with the IBMTSSVC_Cluster instance to which the selected IBMTSSVC_StorageVolume instances belong.
4. Invoke the IBMTSSVC_StorageConfigurationService.AttachReplica() method with the following parameter specifications:
   a. Set SourceElement to the reference (CIMObjectPath) of the source IBMTSSVC_StorageVolume.
   b. Set TargetElement to the reference (CIMObjectPath) of the target IBMTSSVC_StorageVolume.
   c. Optionally set ElementName to the name of the synchronization.
   d. Optionally set BackgroundCopyRate to the preferred priority of the background copy rate in percent (0 - 100%).
   e. Optionally specify Set to add the newly created FlashCopySynchronization to the set. If you specify a null value, the newly created FlashCopySynchronization will not be a member of a synchronized set.
   f. Optionally set CopyType to 5. This sets AutoDelete to true, which automatically deletes the FlashCopy mapping after the background copy is complete.
   g. If you do not want to automatically delete FlashCopy mappings, set CopyType to 4.
5. Create an IBMTSSVC_FlashCopyStorageSynchronizedSet instance by invoking the IBMTSSVC_StorageConfigurationService.CreateSynchronizedSet() method with the following parameter specifications:
   a. Set CopyType to 4 (flash).
   b. Optionally set ElementName to the name of the newly created IBMTSSVC_FlashCopyStorageSynchronizedSet instance.
6. Add the IBMTSSVC_StorageSynchronized instance to the IBMTSSVC_FlashCopySynchronizedSet instance by invoking the IBMTSSVC_StorageConfigurationService.ModifySynchronizedSet() method with the Operation parameter set to 0 (add). If the FlashCopy added to the set is CopyType 5, the set becomes CopyType 5, and any other FlashCopy mappings that are added to the set must have CopyType 5, or the add fails. Similarly, if a set contains a FlashCopy of CopyType 4, all other FlashCopy mappings in the set must be of CopyType 4 or the modification operation fails.

   The synchronization must belong to the same cluster as the hosting service.

Creating a synchronous copy relationship between volumes in the same cluster

The IBMTSSVC_StorageConfigurationService class provides the methods for creating a synchronous copy relationship between a source IBMTSSVC_StorageVolume and a target IBMTSSVC_StorageVolume.

Perform the following steps to create the synchronous copy relationship:

1. Select an IBMTSSVC_StorageVolume instance as the source volume for the preferred source volume.
2. Select an IBMTSSVC_StorageVolume instance as the target volume.
3. Obtain the reference (CIMObjectPath) of the IBMTSSVC_StorageConfigurationService instance that is associated with the IBMTSSVC_Cluster instance to which the selected volumes belong.
4. Invoke the IBMTSSVC_StorageConfigurationService.AttachReplica() method with the following parameter specifications:
   a. Set SourceElement to the reference (CIMObjectPath) of the source IBMTSSVC_StorageVolume instance.
   b. Set TargetElement to the reference (CIMObjectPath) of the target IBMTSSVC_StorageVolume.
   c. Optionally set ElementName to the name of the synchronization.
   d. Set CopyType to 3 for Synchronous and 2 for Asynchronous.

The source IBMTSSVC_StorageVolume instance and the target IBMTSSVC_StorageVolume are now connected through the RemoteStorageSynchronized association.

Creating a synchronous copy relationship between volumes in different clusters

The IBMTSSVC_StorageConfigurationService class provides the methods for creating a synchronous copy relationship between a source IBMTSSVC_StorageVolume instance and a target IBMTSSVC_RemoteStorageVolume instance belonging to different IBMTSSVC_Cluster instances.

To create a synchronous copy relationship between two volumes with the source located in a local cluster and the target located in a remote cluster, perform the following steps:

1. Identify an IBMTSSVC_Cluster instance as the source cluster for the preferred synchronous copy relationship.
2. Obtain the reference (CIMObjectPath) of the IBMTSSVC_StorageConfigurationService instance that is associated with the source cluster.
3. Identify the IBMTSSVC_RemoteCluster on which you want the synchronous copy to reside by traversing the IBMTSSVC_RemotePartnership association.
4. Create a remote cluster partnership by invoking the IBMTSSVC_CascadingAllocationService.Allocate() method with the following parameter specifications:
   a. Set RemoteCluster to the reference (CIMObjectPath) of the IBMTSSVC_RemoteCluster instance.
   b. Optionally set Bandwidth to the preferred bandwidth in megabytes (MB).

   Make sure to issue the method from both the source and candidate clusters to establish a fully configured partnership; otherwise, the synchronous copy relationship cannot be established.
5. Select an IBMTSSVC_StorageVolume as the source volume from the source IBMTSSVC_Cluster.
6. Select an IBMTSSVC_CandidateVolume as the target volume from the IBMTSSVC_RemoteCluster. (IBMTSSVC_StorageVolume instances on the remote cluster are seen on the local cluster as IBMTSSVC_CandidateVolume instance). To determine a candidate volume, use the IBMTSSVC_CopyCandidate association from the Source Volume.
7. Invoke the IBMTSSVC_StorageConfigurationService.AttachReplica() method with the following parameter specifications:

a. Set SourceElement to the reference (CIMObjectPath) of the source IBMTSSVC_StorageVolume instance.

b. Set TargetElement to the reference (CIMObjectPath) of the target IBMTSSVC_CandidateVolume.

c. Optionally set ElementName to the name of the synchronization.

d. Set CopyType to 3.

The source IBMTSSVC_StorageVolume instance and the target IBMTSSVC_StorageVolume or IBMTSSVC_RemoteStorageVolume instance are now connected through the IBMTSSVC_MasterRemoteStorageSynchronized association.

Sample code to create and start a FlashCopy mapping

This information demonstrates how CIMOM Methods can control the cluster. The sample code includes a main method from a Java class that is designed to create and start a FlashCopy mapping and other methods that are called from the main method.

In this topic, the term *method* refers to a Java method. The term *Method* (initial capital) refers to a CIM Method.

Java main method

This example shows the Java main method for creating and starting a FlashCopy mapping. The assumption in this example is that your Java program is designed to control the same cluster every time. It is a relatively simple process to make it more flexible, but that decision is left to you.

```
/*
* FC Mapping states
*/
private static UnsignedInt16 INITIALIZED = new UnsignedInt16(2);
private static UnsignedInt16 PREPARING = new UnsignedInt16(3);
private static UnsignedInt16 PREPARED = new UnsignedInt16(4);

public static void main(String[] args) throws CIMException
{
   /*
    * First step is to connect to the CIMOM
   */
   UserPrincipal user = new UserPrincipal("superuser");
   PasswordCredential pwd = new PasswordCredential("itso13sj");
   CIMNameSpace ns = new CIMNameSpace("https://9.43.86.115:5989/root/ibm");

   CIMClient client = null;

   client = new CIMClient(ns,user,pwd);

   /*
    * Next, select the cluster that we are interested in
   */
   CIMInstance chosenCluster = getCluster("ITSOCL1",client);

   /*
    * At this point, the relevant cluster has been selected
    * and 'chosenCluster' is a CIMInstance of this cluster
    *
    * Get the Config Service of this cluster
    */
   CIMObjectPath cService = getConfigService(chosenCluster, client);

   /*
    * Now, get all of the Volumes in this cluster
    */
   Map<Integer,CIMObjectPath> volumesById = getVolumes(chosenCluster,client);
   /*
    * Select the FlashCopy Source
    *
    * In this case, Volume 10 is our source
    * Volume 11 is our target
    */
   CIMObjectPath fcSrc = volumesById.get(new Integer(10));
   CIMObjectPath fcTgt = volumesById.get(new Integer(11));/*

   /*
    * Now create FC Mapping
    */
   CIMValue rc = makeFlashCopyMapping("CIMOMTestMap", fcSrc, fcTgt, cService,
      client,false);

   /*
    * Now that this has been created, we need to get an
    * Object Path to the newly created Association
    */
   List<CIMObjectPath> fcMaps = getFCMappings(fcSrc, client);
   CIMObjectPath fcMapping = fcMaps.get(0);

   /*
    * Now we prepare the FC Mapping
    */
   CIMArgument[] outArgs = new CIMArgument[2];
   rc = prepareFCMapping(cService, fcMapping, client, outArgs);
   System.out.println("Got value:"+
      Integer.toHexString(Integer.parseInt(rc.toString())));

   /*
    * Loop until it is prepared
    */
   CIMValue fcMapState = new CIMValue(PREPARING);
   while(fcMapState.equals(new CIMValue(PREPARING)))
   {
      CIMInstance fcMapInfo = client.getInstance(fcMapping);
      fcMapState = fcMapInfo.getProperty("SyncState").getValue();
   }

   /*
    * Now start the FC Mapping
    */
   rc = startFCMapping(cService, fcMapping, client, outArgs);
   System.out.println("Got value:"+
      Integer.toHexString(Integer.parseInt(rc.toString())));
}
```

getCluster method

The getCluster method returns the CIM Instance that corresponds to the cluster with the supplied name. It does this by enumerating all of the instances of the class IBMTSSVC_Cluster and then checking the name of each one. When one is found that matches the supplied name, an object path to that instance is returned.

```
static private CIMInstance getCluster(String clusterName, CIMClient client) throws
CIMException
{
   CIMInstance chosenCluster = null;
   Enumeration<CIMInstance> clusters =
      client.enumerateInstances(new CIMObjectPath("/root/ibm:IBMTSSVC_Cluster"));

   while(clusters.hasMoreElements())
   {
      CIMInstance possibleCluster = clusters.nextElement();
      String possibleName =
         possibleCluster.getProperty("ElementName").getValue().toString();

      if(possibleName.equals("\""+clusterName+"\""))
      {
         chosenCluster = possibleCluster;
      }
   }
   return chosenCluster;
}
```

getConfigService method

The CIM_StorageConfigurationService class has no direct equivalent in an SVC, but an Instance of this Class is required for invoking Methods against.

In this method, all of the instances that are associated with the supplied cluster are requested. The association that connects a cluster to its configuration service is CIM_HostedService. Because a cluster will only have configuration service associated with it, the first object path in the enumeration is selected.

```
static private CIMObjectPath getConfigService(CIMInstance cluster, CIMClient
client) throws CIMException
{
   Enumeration<CIMObjectPath> configServices = null;
   configServices = client.associatorNames(
         cluster.getObjectPath(),
         "CIM_HostedService",
         "CIM_StorageConfigurationService",
         null,
         null);
   return configServices.nextElement();
}
```

getVolumes method

This method returns a map that relates Volume IDs (as integers) to IBMTSSVC_StorageVolume object paths. The method requests all of the IBMTSSVC_StorageVolume instances that are associated with the provided cluster instance.

```
static private Map<Integer,CIMObjectPath> getVolumes(CIMInstance cluster, CIMClient
client) throws CIMException
{
   Enumeration<CIMObjectPath> volumes = client.associatorNames(
         cluster.getObjectPath(),
         null,
         "IBMTSSVC_StorageVolume",
         null,
         null);

   Map<Integer,CIMObjectPath> volumesById = new HashMap<Integer, CIMObjectPath>();

   while(volumes.hasMoreElements())
   {
      CIMObjectPath volumeOP = volumes.nextElement();
      CIMValue volumesId = volumeOP.getKey("DeviceID").getValue();
      String idAsString = volumeId.toString();
      String idNoQuotes = idAsString.substring(1, idAsString.length()-1);
      volumesById.put(Integer.parseInt(idNoQuotes), volumeOP);
   }
   return volumesById;
}
```

makeFlashCopyMapping method

This example invokes the AttachReplica against the cluster configuration service. CIM Methods take typed parameters. In this method, you can see the use of the argRef, argString, and argUint16 methods. These methods act as shortcuts to generating the required arguments for the CIM Method. The AttachReplica method is used for FlashCopy, Metro Mirror and Global Mirror. The CopyType argument indicates which type is required.

```
static private CIMValue makeFlashCopyMapping(
           String name,
           CIMObjectPath source,
           CIMObjectPath target,
           CIMObjectPath configService,
           CIMClient client,
           boolean autodelete) throws CIMException
{
   CIMArgument src = argRef("SourceElement", source, "IBMTSSVC_StorageVolume");
   CIMArgument tgt = argRef("TargetElement", target, "IBMTSSVC_StorageVolume");
   CIMArgument fcName = argString("ElementName",name);
   CIMArgument type = argUint16("CopyType",autodelete?5:4);
   CIMArgument[] inArgs = {src,tgt,fcName,type};
   CIMArgument[] outArgs = new CIMArgument[1];

   CIMValue rc = client.invokeMethod(configService,
      "AttachReplica",
      inArgs,
      outArgs);
   return rc;
}
```

getFCMappings method

The getFCMappings method returns a list of all the FCMappings that are associated with the provided VDisk. This method requests a list of all of the associations that reference the provided IBMTSSVC_StorageVolume. Currently, all of the Java WBEM Services methods of this type return enumerations. This method converts this to a list for ease of use.

```
static private List<CIMObjectPath> getFCMappings(CIMObjectPath volume, CIMClient
client) throws CIMException
{
   Enumeration<CIMObjectPath> assocs = client.referenceNames(
              volume,
              "IBMTSSVC_LocalStorageSynchronized",
              null);
   return Collections.list(assocs);
}
```

prepareFCMapping method

The prepareFCMapping method prepares a FlashCopy mapping. Much like the AttachReplica Method, the ModifySynchronization Method is used to control FlashCopy, Metro Mirror and Global Mirror. The operation parameter indicates what you actually want to do.

```
private static CIMValue prepareFCMapping(
           CIMObjectPath configService,
           CIMObjectPath fcMapping,
           CIMClient client,
           CIMArgument[] outArgs) throws CIMException
{
   CIMArgument operation = argUint16("Operation", 6);
   CIMArgument synch = argRef("Synchronization",
fcMapping,"IBMTSSVC_FlashCopyStorageSynchronized");

   CIMArgument[] inArgs = new CIMArgument[]{operation,synch};
   outArgs = new CIMArgument[2];
   return client.invokeMethod(configService,
         "ModifySynchronization",
         inArgs,
         outArgs);
```

startFCMapping method

The startFCMapping method starts a FlashCopy mapping. This method invokes the ModifySynchronization Method as in "prepareFCMapping method" on page 39 but uses different Operation parameter.

```
private static CIMValue startFCMapping(
      CIMObjectPath configService,
      CIMObjectPath fcMapping,
      CIMClient client,
      CIMArgument[] outArgs) throws CIMException
{
   CIMArgument operation = argUint16("Operation", 4);
   CIMArgument synch = argRef("Synchronization",
fcMapping,"IBMTSSVC_FlashCopyStorageSynchronized");

   CIMArgument[] inArgs = new CIMArgument[]{operation,synch};
   outArgs = new CIMArgument[2];

   return client.invokeMethod(configService,
         "ModifySynchronization",
         inArgs,
         outArgs);
}
```

Argument generators class

This class uses the following argument generators:

- The **argUint16 method** returns an unsigned 16-bit integer typed argument.

```
static private CIMArgument argUint16(String name, int arg)
{
   return new CIMArgument(
         name,
         new CIMValue(
         new UnsignedInt16(arg),
         new CIMDataType(CIMDataType.UINT16)
         )
   );
}
```

- The **argString method** returns a string-typed argument.

```
static private CIMArgument argString(String name, String str )
{
   return new CIMArgument(
         name,
         new CIMValue(
               str,
               new CIMDataType(CIMDataType.STRING)
               )
         );
}
```

- The **argRef method** returns a reference typed argument. It is a reference to the instance that the provided object path indicates.

```
static private CIMArgument argRef(
      String name,
      CIMObjectPath path,
      String className )
{
   return new CIMArgument(
         name,
         new CIMValue(
               path,
               new CIMDataType(className)
               )
         );
}
```

Chapter 4. CIM agent network considerations

You can manually set the CIM agent service or the user interface connection information for remote connection access..

RemoteServiceAccessPoint instance

In an environment with multiple network adapters, it might be necessary to manually set the connection data of the RemoteServiceAccessPoint (RSAP) instance.

The IBMTSSVC_RemoteServiceAccessPoint class hosts the information necessary for connection to the Web user interface. Management applications can obtain an instance of the RSAP from the CIMOM to launch the user interface through the Web.

You can manually set the connection data of the RSAP. This is helpful in an environment with multiple network cards.

To set the connection data, follow these steps:

1. Obtain the IBMTSSVC_Cluster instance.
2. Modify the ConsoleIP and ConsolePort properties through the Modify instance on the Cluster with a property list that contains the Properties that have changed.

   The CIM agent automatically updates the RSAP.

Chapter 5. Maintenance and diagnostic tasks

Perform maintenance, informational and diagnostic tasks.

You can perform tasks that provide information about your system.

Getting a list of classes

Use the IBMTSSVC_ConfigurationReportingService to get information about enclosure classes.

To retrieve a list of classes that the IBMTSSVC_Enclosure supports or has installed, perform the following steps:

1. Select an IBMTSSVC_Enclosure instance as the container.
2. Retrieve the IBMTSSVC_ConfigurationReportingService instance.
3. Invoke the IBMTSSVC_ConfigurationReportingService.GetClassTypes () method with the following parameter specifications:
   a. Set InquiryType to 2 or 3.
   b. Set CIM_ManagedElement to the reference (CIMObjectPath) of the target IBMTSSVC_Enclosure.

The returned ClassTypes[] contains the class names of the classes that the IBMTSSVC_Enclosure supports or has installed. The IBMTSSVC_Enclosure currently supports class type IBMTSSVC_DiskDrive.

Getting a list of UnitTypes

Use the IBMTSSVC_ConfigurationReportingService to get information about enclosure UnitTypes.

To retrieve a list of unit types that the IBMTSSVC_Enclosure supports or has installed, perform the following steps:

1. Select an IBMTSSVC_Enclosure instance as the container.
2. Obtain the IBMTSSVC_ConfigurationReportingService instance directly.
3. Invoke the IBMTSSVC_ConfigurationReportingService.GetClassTypes () method with the following parameter specifications:
   a. Set InquiryType to 2 or 3.
   b. Set ClassType to be an entry returned in the GetClassTypes.ClassTypes property.
   c. Set CIM_ManagedElement to the reference (CIMObjectPath) of the target IBMTSSVC_Enclosure.

The returned UnitTypes[] contains the unit types of the ClassTypes supported or installed in the IBMTSSVC_Enclosure.

Getting the number of currently installed units

Use the IBMTSSVC_ConfigurationReportingService to get information about the maximum number of supported units or the number of currently installed units for the given UnitType, for a given ClassType .

To retrieve information about the maximum number of supported units or the number of currently installed units for the given UnitType, for a given ClassType, perform the following steps:

1. Select an IBMTSSVC_Enclosure instance as the container.
2. Retrieve the IBMTSSVC_ConfigurationReportingService instance that is associated with the IBMTSSVC_Cluster instance to which the selected IBMTSSVC_Enclosure instances belong.

3. Invoke the IBMTSSVC_ConfigurationReportingService.GetClassTypes () method with the following parameter specifications:
   a. Set InquiryType to 2 if the supported number is to be returned, or 3 if the currently installed number is to be returned.
   b. Set ClassType to be a entry returned in the GetClassTypes.ClassTypes property
   c. Set CIM_ManagedElement to the reference (CIMObjectPath) of the target IBMTSSVC_Enclosure.

The returned NumberOfUnits is the maximum number of supported or the number of currently installed units of the ClassTypes supported or installed in the IBMTSSVC_Enclosure.

Using the indication service

You can turn the indication service on or off.

The IBMTSSVC_ObjectManagerAdapter class provides the methods for accessing the indication service. To turn the indication service on or off, perform the following steps:

1. Retrieve the IBMTSSVC_ObjectManagerAdapter instance that is associated with the IBMTSSVC_Cluster instance.
2. Invoke the IBMTSSVC_ObjectManagerAdapter.StartService() to start the indication service, and the IBMTSSVC_ObjectManagerAdapter.StopService() method to stop the indication service.

Getting performance statistics

The IBMTSSVC_PerformanceStatisticsService class provides the method for getting the performance statistics for IBMTSSVC_Node, IBMTSSVC_DiskDrive, IBMTSSVC_StorageVolume, and IBMTSSVC_BackendVolume.

To get the performance statistics instance, follow these steps:

1. Obtain the reference (CIMObjectPath) of an IBMTSSVC_PerformanceStatisticsService instance that is associated with the IBMTSSVC_Cluster in which you will get the performance statistics by traversing the IBMTSSVC_HostedPerformanceStatisticsService association.
2. Invoke the IBMTSSVC_PerformanceStatisticsService.GetStatisticsCollection method to get the performance statistics. Use the following parameter specifications:
   - The ElementTypes parameter is used to set the preferred statistics that would be returned according to the preferred source:
     - 4 to return the performance statistics of IBMTSSVC_StorageVolume
     - 8 to return the performance statistics of the IBMTSSVC_DiskDrive
     - 32768 to return the performance statistics of the IBMTSSVC_BackendVolume
     - 32769 to return the performance statistics of the IBMTSSVC_Node
   - Set ManifestCollection to the reference (CIMObjectPath) of CIM_BlockStatisticsManifestCollection.
   - Set StatisticsFormat to 2 (which will return a comma-separated value [CSV]).

   The order of statistic properties returned by the GetStatisticsCollection method is the same as the order of properties definitions in the Managed Object Format (MOF) file.

Collecting CIM log files

You can generate CIM log files for diagnostic reporting to the IBM Support Center.

CIMOM logs are collected from the management GUI.

To collect all cluster logs and CIMOM logs, complete the following steps:

1. Launch the management GUI.
2. Click **Troubleshooting** → **Support** → **Download Support Package**.
3. Select the type of log you need and click **Download**.

   Note: Setting the highest level of logging on the CIMOM significantly decreases the performance of the CIMOM.

Chapter 6. CIM agent classes, methods, and SMI-S profiles supported

The CIM agent classes are the building blocks of the Common Information Model (CIM) agent and use functions such as storage configuration, Copy Services, and logical unit number (LUN) masking. The Storage Management Initiative Specification (SMI-S) is based on a number of existing technologies that include the CIM.

To view the complete Managed Object Format (MOF) documentation of these classes and methods, select the documentation information from the following website:

Support for Storwize V7000 website at www.ibm.com/storage/support/storwize/v7000

Table 3 identifies the SMI-S profiles that CIM agent for Storwize V7000 supports.

Table 3. SMI-S profiles and subprofiles supported by CIM

Profiles	Subprofiles
SNIA:Storage Virtualizer Storage virtualizers act in a manner similar to RAID arrays but can use storage that is provided by systems that are external to the storage virtualizer and local disks. A storage virtualizer system combines both remote and local storage to create a seamless pool. The virtualization system allocates volumes from the pool for host systems to use. The basic virtualizer system profile provides a read-only view of the system. The various subprofiles extend this description and also enable configuration.	SNIA:Storage Virtualizer:Access Points
	SNIA:Storage Virtualizer:Block Server Performance
	SNIA:Storage Virtualizer:Block Services
	SNIA:Storage Virtualizer:Cascading
	SNIA:Storage Virtualizer:Copy Services
	SNIA:Storage Virtualizer:Disk Drive Lite
	SNIA:Storage Virtualizer:Disk Sparing
	SNIA:Storage Virtualizer:Extent Composition
	SNIA:Storage Virtualizer:FC Initiator Ports
	SNIA:Storage Virtualizer:FC Target Ports
	SNIA:Storage Virtualizer:Health
	SNIA:Storage Virtualizer:iSCSI
	SNIA:Storage Virtualizer:Location
	SNIA:Storage Virtualizer:Masking and Mapping
	SNIA:Storage Virtualizer:Multiple Computer System
	SNIA:Storage Virtualizer:Physical Package
	SNIA:Storage Virtualizer:RAID
	SNIA:Storage Virtualizer:Software
SNIA:Server The server profile is mandatory for all compliant SMI-S servers. The object manager part of the model defines the capabilities of a CIM object manager based on the communication mechanisms that it supports.	SNIA:Server:Profile Registration
	SNIA:Server:Indication Object Manager Adapter

Chapter 7. Return codes

The Common Information Model (CIM) return codes provide information on the status of CIM agent operations.

Common Information Model

Table 4 is a condensed list of possible CIM return codes.

Table 4. Common Information Model return codes

CIM return code	Description	Method	Explanation
0	Success	GetFreeExtends() ListConfiguration Backups()	The parameters are valid. The method completed successfully.
		AddNode()	The node was successfully added.
		CheckValidity()	The information about the certificate was successfully obtained.
		DeleteAccount()	The account was successfully deleted.
		GenerateCIMOM Certificate() EnableAuto Generation() DisableAuto Generation()	The certificate was successfully deleted.
		CreateCode()	The new account was successfully created.
		SetDefault Validity()	The validity was successfully set.
		GrantGlobal Access() GrantSystem Access()	The role was successfully changed.
		CreateGatewayID() AddHardwareIDs ToCollection()	The collection was successfully created.
		DeleteStorage HardwareID()	The StorageHardwareID was successfully deleted.
		CreateStorage HardwareID()	The StorageHardwareID was successfully created.

Table 4. Common Information Model return codes (continued)

CIM return code	Description	Method	Explanation
0	Success (continued)	AttachDevice()	The volume was successfully attached.
		DeleteProtocol Controller()	The controller was successfully deleted.
		CreateProtocol ControllerWith Ports()	A clone was successfully created.
		DeleteRemote Cluster Partnership()	The cluster partnership was successfully deleted.
		CreateRemote Cluster Partnership()	The cluster partnership was successfully established.
		DeleteHardwareID Collection()	The collection was successfully created.
		DeleteCertificate()	The certificate was successfully deleted.
		DeleteSynchronized Set()	The SynchronizedSet was successfully deleted.
		Modify Synchronisation()	The method was successfully run.
		EvictNode()	The node was successfully evicted.

Table 4. Common Information Model return codes (continued)

CIM return code	Description	Method	Explanation
0	Success (continued)	RestartService()	The CIMOM reboots.
		Shutdown()	The shutdown for the node/cluster was successfully initiated.
		SetLocale()	The locale was set.
		SetTimezone()	The time zone for the cluster was successfully set.
		SetPasswords() ModifyReset Password ChangeFeature()	The passwords were changed.
		GetResetPassword ChangeFeature Status()	The feature status was successfully retrieved.
		StartStatistics Collection()	The statistics collection was started.
		DetachDevice()	The volume was successfully detached.
		StopStatistics Collection()	The statistics collection was stopped.
		Backup Configuration()	A backup was successfully created.
		Reload Configuration()	The configuration was reloaded.
		Restore Configuration() Delete Configuration Backup()	A restore was successfully made.
		AttachReplica()	The copy relationship was successfully established.
		CreateSynchronized Set()	The SynchronizedSet was successfully created.
		SetPassword()	The password was successfully changed.

Table 4. Common Information Model return codes (continued)

CIM return code	Description	Method	Explanation
0	Job completed with no error	CreateOrModify StoragePool()	The pool was successfully created.
		CreateOrModify ElementFrom StoragePool()	The volume was successfully created. The pool was successfully modified.
		DeleteStorage Pool()	The pool was successfully deleted.
		ReturnToStorage Pool()	The volume was successfully deleted.
		RequestDiscovery()	BackendVolume discovery was successful.
		SetIOGroup()	The change was successful.
		SetQuorum()	The method was successful.
		IncludeBackend Volume()	The volume was successfully included.
		ModifySynchronized Set()	The CLI command was successfully run.
0	Job started successfully	MigrateVolume() MigrateVolumeTo lmageMode()	The migration job was started.
1	Not supported	SetLocales()	The cluster does not support locales.
		SetPasswords()	The cluster does not support password change (CISCO).
		ModifyReset PasswordChange Feature() GetResetPassword ChangeFeature Status()	The cluster does not support the reset password change feature.
		Upgrade()	The method was called on a 2145 Cluster Configuration Service.

Table 4. Common Information Model return codes (continued)

CIM return code	Description	Method	Explanation
2	Failed	Dump() Clean() Enter() Exit() Clean() GetDump() ClearLog() GetRecord() GetAllRecords() FixRecord() UnfixRecord() ModifyError Settings() Create2062 Cluster()	An unexpected error occurred. A CLI command failed.
		GetDump()	Failure during command processing. The file was not found.
		Reload Configuration()	Failed to reload the configuration.
		CreateCode()	Failed to create the account.
		GrantGlobal Access() GrantSystem Access()	Failed to change the role.
		SetPassword()	Failed to change the password.
		DeleteAccount()	Failed to delete the account.
2	Unknown error	GenerateCIMOM Certificate() DeleteCertificate()	Failed to generate new certificate due to provider internal reasons.
		EnableAuto Generation() DisableAuto Generation()	Failed due to provider internal reasons.
		SetDefault Validity()	Failed to set validity.
		CheckValidity()	Failed to obtain information on certificate.
4	Failed	GenerateCIMOM Certificate() DeleteCertificate()	Unexpected error occurred. Failed to generate new certificate due to truststore problems.

Table 4. Common Information Model return codes (continued)

CIM return code	Description	Method	Explanation
5	Wrong Parameter Set	Delete Configuration Backup()	The wrong number or type of parameters were passed. The given backup could not be found.
		Modify Synchronisation()	The wrong number or type of parameters were passed or other parameter checking failed.

		CreateCode()
		GrantGlobal Access()
		SetPassword()
		GrantSystem Access()
		DeleteAccount()
		DeleteCertificate()
		SetDefault Validity()
		CheckValidity()
			One of the parameters is not valid.

5	Parameter not valid	Dump(),
		GetDump()
		PositionToFirst RecordRoot()
		GetRecord()
		FixRecord()
		UnfixRecord()
			One of the mandatory parameters is missing.

		ModifyIP Address()
		Create2062 Cluster()
		Add2062Cluster()
		Add2145Cluster()
		Reset2062Node()
		Reload2062Node()
			One of the mandatory parameters is missing or not valid.

		CreateOrModify StoragePool() CreateOrModify ElementFrom StoragePool() Delete StoragePool() ReturnTo StoragePool()	At least one of the parameters is not valid.
		CreateGatewayID() GenerateCIMOM Certificate()	One of the parameters was not valid.
		PositionAtRecord()	One of the mandatory parameters is missing or having a negative number of records is not allowed.
6	CopyType not supported	ModifySynchronized Set()	The type of copy passed in was different from 3 or 4.

Table 4. Common Information Model return codes (continued)

CIM return code	Description	Method	Explanation
6	Operation not supported	ModifySynchronized Set()	The operation code submitted is not valid for the copy type of the synchronized set.
6	SynchronizedSet is not empty	DeleteSynchronized Set()	There are still StorageSynchronized associations in the set. All StorageSynchronized associations must be removed before deletion of the set can be run or the Force flag must be set.
6	User ID already exists	CreateCode()	The user ID that you submitted exists in another account.
6	In use	GenerateCIMOM Certificate()	Failed to generate new certificate. Existing certificate is still valid and in use.
7	StorageSynchronized not in the Set		The synchronized storage does not exist in the set.
8	StorageSynchronized already in the Set		The synchronized storage already exists in the set and cannot be added.
9	StorageSynchronized incompatible with Set		The synchronized storage is not compatible with the Set. For example, a Flash Copy is synchronized to a Sync Copy set.
0*x*1000	Parameters checked – Job started		The CLI copy command was run and a job object was returned.
0*x*1000	LogicalDevices associated to other ProtocolControllers not deleted	DeleteProtocol Controller()	At least one of the attached storage volumes is attached to another controller so it cannot be deleted.
0*x*1000	LogicalDevice instance not valid	AttachDevice()	The device is not a volume of the RedundancyGroup of the controller.
0*x*1000	LogicalDevice not associated to Controller	DetachDevice()	The device has no ProtocolControllerForUnit association to this controller.
0*x*1000	ID already created	CreateStorage HardwareID()	The WWPN is already assigned to an existing storage hardware ID.
0*x*1000	Specified instance not found	DeleteStorage HardwareID()	The storage hardware ID could not be found.
0*x*1000	HardwareID instance not valid	CreateGateway ID() AddHardwareIDs ToCollection()	The storage hardware ID could not be found or is already a member of another collection.

Table 4. Common Information Model return codes (continued)

CIM return code	Description	Method	Explanation
0*x*1001	Size not supported	CreateOrModify ElementFrom StoragePool()	The requested size is not supported by the primordial pools. The size parameter contains the nearest supported size larger than the requested one. The size requested was not a multiple of 512. The nearest supported size that can be requested is returned in size.
0*x*1001	Device Number Conflict	AttachDevice()	The specified device number is already occupied.
0*x*1001	Hardware implementation does not support specified IDType	CreateStorage HardwareID()	The type of ID is different from 2.
0*x*8000	ComputerSystem not valid	AddNode()	The submitted ComputerSystem was not a IBMTSSVC_ CandidateNode.
		EvictNode()	The submitted ComputerSystem was not a IBMTSSVC_Node.
		Shutdown()	The submitted ComputerSystem was not a IBMTSSVC_Node of IBMTSSVC_Cluster.
0*x*8000	Locale not valid	SetLocale()	The submitted Locale was greater than 9.
0*x*8000	Type not valid	Dump()	A type greater than 2 was passed in.
0*x*8000	Connection refused	GetDump()	We lost connection to the cluster or failed to connect to a node (CISCO only).
0*x*8000	Backup not found	Restore Configuration()	The specified backup was not found.
0*x*8000	Delete failed	Delete Configuration Backup()	The deletion of the backup directory failed. The failure might have been caused by a sharing violation.
0*x*8000	IOGroup must have Nodes aggregated	SetIOGroup()	The I/O group does not have any nodes.
0*x*8000	ID not valid	SetQuorum()	The quorum ID is a number greater than 2.
0*x*8000	Volume not valid	IncludeBackend Volume()	The volume is not expelled.
0*x*8000	CopyType not supported	AttachReplica()	The type of copy passed in was different from 2 or 3.
		CreateReplica()	The type of copy passed in was different from 3 or 4.

Table 4. Common Information Model return codes (continued)

CIM return code	Description	Method	Explanation
0*x*8000	Ports are from multiple IOGroups	CreateProtocol ControllerWith Ports()	All ports are required to belong to the same I/O group.
0*x*8000	HardwareID still bound to AuthorizationSubject. Force required	DeleteStorage HardwareID()	The hardware ID has access granted to a storage volume and Force was not specified.
0*x*8000	Host is member of a LUN mapping	DeleteHardware IDCollection()	To delete this host either use this host to run the RemoveAccess method for each privilege and controller this host is associated to or set "Force" equal to "True."

0*x*8000	Record(s) not found	GetRecord()
		GetAllRecords()
			No records were found.

0*x*8000	Cannot connect to cluster	Create2062 Cluster() Add2062 Cluster()	Unable to connect to the cluster.
0*x*8000	Connection to cluster refused	Add2145Cluster()	The connection to the cluster was refused.

0*x*8000	Connection to switch refused	Reset2062Node()
		Reload2062Node()
			The connection to the switch was refused.

0*x*8000	Cluster IP not found	RemoveCluster()	The IP for the cluster cannot be found.
0*x*8001	Maximum number of Nodes for Cluster exceeded	AddNode()	All I/O groups already have two nodes assigned to them.
0*x*8001	Prefix not valid	Dump()	A file prefix and feature log type were passed in at the same time.
0*x*8001	File not found	GetDump()	The given file path was not found (CISCO).
0*x*8001	Backup script failed	Backup Configuration()	The backup script returned with an error.
0*x*8001	Restore script failed	Restore Configuration()	The backup script returned with an error.
0*x*8001	Operation not allowed for current state	Modify Configuration()	The operation submitted is not allowed in the current state of the synchronized storage. For example, you cannot have a "prepare" operation on a synchronized storage in "synchronized" state.
0*x*8001	Operation not allowed for current SyncState	Modify Synchronized Set()	The operation is not allowed with the current SyncState of the set.

Table 4. Common Information Model return codes (continued)

CIM return code	Description	Method	Explanation
0x8001	Unsupported protocol	CreateProtocol ControllerWith Ports()	Protocol != 2.

0x8001	Syntax error in ClusterName	Create2062 Cluster()
		Add2062Cluster()
		Reset2062Node()
		Reload2062Node()
			The cluster name is not valid because of a syntax error.

0x8002	ExtraCapacitySet not valid	AddNode()	The submitted ExtraCapacitySet was not a IBMTSSVC_ IOGroupSet.
0x8002	Secure copy failed	Backup Configuration()	The download of the backup file using secure copy failed.
0x8002	Secure copy failed	Upload Configuration()	The upload of the backup file using secure copy failed.
		CreateStorage HardwareID()	The name of the element and setting are required to be null.

0x8002	Syntax error in Node or Node is not valid	Create2062 Cluster()
		Add2062Cluster()
		Reset2062Node()
		Reload2062 Node()
			The node contains a syntax error, or the specified node is not valid.

0x8003	Maximum number of Nodes for IOGroup exceeded	AddNode()	The submitted I/O group set already has two nodes assigned to it.
0x8003	Creation of backup dir failed	Backup Configuration()	The backup directory cannot be created.
0x8003	Clear command failed	Upload Configuration()	The cluster /tmp/ directory cannot be cleared.

0x8003	Username or password not valid (only ResetNode)	Add2062Cluster()
		Reset2062Node()
		Reload2062Node()
			The user name or password are not valid.

0x8004	Delete/rename of old backup files failed	N/A	The backup directory cannot be renamed or deleted.
0x8004	Wrong SwitchIP / can't connect to switch	Create2062 Cluster() Add2062Cluster()	The IP for the switch is not correct, so a connection to the switch cannot be made.

0x8004	SwitchIP is not configured	Reset2062Node()
		Reload2062Node()
			The IP for the switch is not configured.

0x8005	Syntax error in ClusterIP	N/A	The IP for the cluster contains a syntax error.
0x8006	Slot not valid	N/A	The slot is not valid.
0x8007	Cannot upload public key to switch	N/A	The public key cannot be uploaded to the switch.

Table 4. Common Information Model return codes (continued)

CIM return code	Description	Method	Explanation
0*x*8100	Cluster Scope Violation	N/A	One or more parameters were out of the cluster scope.
0*x*8200	N/A	N/A	The method was run successfully but one or more parameters were ignored.

Appendix. Accessibility

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully.

Features

This list includes the major accessibility features in the management GUI:

- You can use screen-reader software and a digital speech synthesizer to hear what is displayed on the screen. The following screen reader has been tested: JAWS 11.
- Most of the GUI features are accessible by using the keyboard. For those features that are not accessible, equivalent function is available by using the command-line interface (CLI).

Navigating by keyboard

You can use keys or key combinations to perform operations and initiate many menu actions that can also be done through mouse actions. You can navigate the management GUI and help system from the keyboard by using the following key combinations:

- To navigate between different GUI panels, select the `Low-graphics mode` option on the GUI login panel. You can use this option to navigate to all the panels without manually typing the web addresses.
- To go to the next frame, press Ctrl+Tab.
- To move to the previous frame, press Shift+Ctrl+Tab.
- To navigate to the next link, button, or topic within a panel, press Tab inside a frame (page).
- To move to the previous link, button, or topic within a panel, press Shift+Tab.
- To select GUI objects, press Enter.
- To print the current page or active frame, press Ctrl+P.
- To expand a tree node, press the Right Arrow key. To collapse a tree node, press the Left Arrow key.
- To scroll all the way up, press Home; to scroll all the way down, press End.
- To go back, press Alt+Left Arrow key.
- To go forward, press Alt+Right Arrow key.
- For actions menus:
  - Press Tab to navigate to the grid header.
  - Press the Left or Right Arrow keys to reach the drop-down field.
  - Press Enter to open the drop-down menu.
  - Press the Up or Down Arrow keys to select the menu items.
  - Press Enter to launch the action.
- For filter panes:
  - Press Tab to navigate to the filter panes.
  - Press the Up or Down Arrow keys to change the filter or navigation for nonselection.
  - Press Tab to navigate to the magnifying glass icon in the filter pane and press Enter.
  - Type the filter text.
  - Press Tab to navigate to the red X icon and press Enter to reset the filter.
- For information areas:
  - Press Tab to navigate to information areas.
  - Press Tab to navigate to the fields that are available for editing.
  - Type your edit and press Enter to issue the change command.

Accessing the publications

You can find the HTML version of the IBM Storwize V7000 information at the following website:

publib.boulder.ibm.com/infocenter/storwize/ic/index.jsp

You can access this information using screen-reader software and a digital speech synthesizer to hear what is displayed on the screen. The information was tested using the following screen reader: JAWS Version 10 or later.

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 give 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 (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

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

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

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

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites 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 measurement 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 may be addressed to the suppliers of those products.

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

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

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

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

Trademarks

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

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

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

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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

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

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

Other company, product, or service names may be trademarks or service marks of others.

Index

Printed in USA