

EMC VNXe Series

Release 3.1

SMI-S Provider Programmer's Guide for VNXe

P/N 302-000-225 Rev 02

EMC Corporation

Corporate Headquarters:

Hopkinton, MA 01748-9103

1-508-435-1000

www.EMC.com

Copyright © 2015, EMC Corporation. All rights reserved.

Published January, 2015

EMC believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED “AS IS.” EMC CORPORATION MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Use, copying, and distribution of any EMC software described in this publication requires an applicable software license.

For the most up-to-date regulatory document for your product line, go to EMC Online Support (<https://support.emc.com>).

For the most up-to-date listing of EMC product names, see EMC Corporation Trademarks on EMC.com.

All other trademarks used herein are the property of their respective owners.

Corporate Headquarters: Hopkinton, MA 01748-9103

SMI-S Provider Programmer's Guide for VNXe

Contents

- Contents 3
- Table of Figures 24
- Table of Tables 26
- Introduction 47
 - Audience* 47
 - VNXe basic requirements* 47
 - Format of Profile Chapter* 47
 - Definitions and conventions* 47
 - Recommendations* 48
- Server Profile 50
 - Overview 50
 - Class diagram 51
 - Method of the Profile 51
 - Client considerations 51
 - Model specifications 51
 - Use case: View Server Profile implementation for EMC CIM server 52
- CIM Element 53
 - ECOM_ObjectManager 53
 - ECOM_RegisteredProfile 54
 - ECOM_System 55
 - ECOM_Namespace 55
 - ECOM_CIMXMLCommunicationMechanism 56
 - ECOM_ElementConformsToProfile 57
 - ECOM_HostedService 57
 - ECOM_CommMechanismForManager 57
 - ECOM_NamespaceInManager 58
- Profile Registration Profile 59
 - Overview 59
 - Class diagram 59
 - Method of the Profile 60
 - Client consideration 60
 - Model specification 60
 - Use case: Find all Profiles on a CIM Server 60

Use case: Identify the ManagedElement defined by a Profile	60
Use case: Enumerate autonomous profiles supported by a given CIM server	61
Use case: Determine the SNIA Version of a SMI-S Profile	62
CIM Elements	62
ECOM_RegisteredProfile	63
ECOM_RegisteredSubProfile	63
ECOM_ReferencedProfile	63
ECOM_SubProfileRequiresProfile	63
ECOM_ElementConformsToProfile	64
Multiple Computer System Subprofile	65
Overview	65
Class diagram	65
Methods of the Profile	65
Client considerations	66
Model Specification	66
Use case: Find top-level ComputerSystem	67
Use case: Check system redundancy	67
Use case: Find top-level ComputerSystem for any LogicalDevice	67
CIM Elements	68
EMC_VNXe_StorageSystemLeaf	69
EMC_VNXe_StorageProcessorSystemLeaf	69
EMC_VNXe_RedundancySetLeaf	70
EMC_VNXe_StorageSystem_StorageProcessorSystem_ComponentCSAssocLeaf	70
EMC_VNXe_StorageSystem_RedundancySet_ConcretelIdentityAssocLeaf	71
EMC_VNXe_RedundancySet_StorageProcessorSystem_MemberOfCollectionAssocLeaf	71
Software Subprofile	72
Overview	72
Class diagram	72
Method of the Profile	72
Client considerations	72
CIM Element	73
EMC_VNXe_StorageSystemSoftwareIdentityLeaf	73
EMC_VNXe_StorageSystem_StorageSystemSoftwareIdentity_InstalledSoftwareIdentityAssocLeaf	73
Physical Package Package	74
Overview	74
Class diagram	74
Methods of the Profile	75

Client considerations	75
Model Specification	75
Use case: Get product information for a ComputerSystem	75
Use case: Find asset information for subcomponents	76
CIM Elements	76
EMC_VNXe_ArrayChassisLeaf	79
EMC_VNXe_DAEEnclosureChassisLeaf	79
EMC_VNXe_DPEEnclosureChassisLeaf	80
EMC_VNXe_SPChassisLeaf	80
EMC_VNXe_DiskModuleLeaf	80
EMC_VNXe_FanModuleLeaf	81
EMC_VNXe_PowerSupplyModuleLeaf	81
EMC_VNXe_ArrayChassisProductLeaf	82
EMC_VNXe_DAEEnclosureChassisProductLeaf	82
EMC_VNXe_DPEEnclosureChassisProductLeaf	82
EMC_VNXe_SPChassisProductLeaf	82
EMC_VNXe_DiskProductLeaf	83
EMC_VNXe_FanProductLeaf	83
EMC_VNXe_PowerSupplyProductLeaf	83
EMC_VNXe_StorageSystem_ArrayChassis_SystemPackagingAssocLeaf	84
EMC_VNXe_StorageProcessorSystem_SPChassis_SystemPackagingAssocLeaf	84
EMC_VNXe_ArrayChassis_DAEEnclosureChassis_ContainerAssocLeaf	84
EMC_VNXe_ArrayChassis_DPEEnclosureChassis_ContainerAssocLeaf	84
EMC_VNXe_DAEEnclosureChassis_DiskModule_ContainerAssocLeaf	84
EMC_VNXe_DAEEnclosureChassis_FanModule_ContainerAssocLeaf	85
EMC_VNXe_DAEEnclosureChassis_PowerSupplyModule_ContainerAssocLeaf	85
EMC_VNXe_DPEEnclosureChassis_DiskModule_ContainerAssocLeaf	85
EMC_VNXe_DPEEnclosureChassis_FanModule_ContainerAssocLeaf	85
EMC_VNXe_DPEEnclosureChassis_PowerSupplyModule_ContainerAssocLeaf	85
EMC_VNXe_DPEEnclosureChassis_SPChassis_ContainerAssocLeaf	86
EMC_VNXe_ArrayChassis_ArrayChassisProduct_ProductPhysicalComponentAssocLeaf	86
EMC_VNXe_DAEEnclosureChassis_DAEEnclosureChassisProduct_ProductPhysicalComponentAssocLeaf	86
EMC_VNXe_DPEEnclosureChassis_DPEEnclosureChassisProduct_ProductPhysicalComponentAssocLeaf	86
EMC_VNXe_SPChassis_SPChassisProduct_ProductPhysicalComponentAssocLeaf	87
EMC_VNXe_DiskModule_DiskProduct_ProductPhysicalComponentAssocLeaf	87
EMC_VNXe_FanModule_FanProduct_ProductPhysicalComponentAssocLeaf	87

EMC_VNXe_PowerSupplyModule_PowerSupplyProduct_ProductPhysicalComponentAssocLeaf ...	87
EMC_VNXe_ArrayChassisProduct_DAEEnclosureChassisProduct_ProductParentChildAssocLeaf	87
EMC_VNXe_ArrayChassisProduct_DPEEnclosureChassisProduct_ProductParentChildAssocLeaf	88
EMC_VNXe_DAEEnclosureChassisProduct_DiskProduct_ProductParentChildAssocLeaf	88
EMC_VNXe_DAEEnclosureChassisProduct_FanProduct_ProductParentChildAssocLeaf	88
EMC_VNXe_DAEEnclosureChassisProduct_PowerSupplyProduct_ProductParentChildAssocLeaf	88
EMC_VNXe_DPEEnclosureChassisProduct_DiskProduct_ProductParentChildAssocLeaf	89
EMC_VNXe_DPEEnclosureChassisProduct_FanProduct_ProductParentChildAssocLeaf	89
EMC_VNXe_DPEEnclosureChassisProduct_PowerSupplyProduct_ProductParentChildAssocLeaf	89
EMC_VNXe_DPEEnclosureChassisProduct_SPChassisProduct_ProductParentChildAssocLeaf	89
Access Points Subprofile	90
Overview	90
Class diagram	90
Method of the Profile	90
Client considerations	90
CIM Element	90
EMC_VNXe_RemoteServiceAccessPointLeaf	91
EMC_VNXe_StorageSystem_RemoteAccessPoint_HostedAccessPointLeaf	91
FC Target Ports Subprofile	92
Overview	92
Class diagram	92
Methods of the Profile	92
Client considerations	92
CIM Elements	93
EMC_VNXe_FCPortLeaf	93
EMC_VNXe_FCSCSIProtocolEndpointLeaf	94
EMC_VNXe_StorageProcessorSystem_FCPort_SystemDeviceAssocLeaf	94
EMC_VNXe_StorageProcessorSystem_FCSCSIProtocolEndpoint_HostedAccessPointAssocLeaf	94
EMC_VNXe_FCPort_FCSCSIProtocolEndpoint_DeviceSAPImplementationAssocLeaf	95
iSCSI Target Ports Subprofile	96
Overview	96
Class diagram	96
Methods of the Profile	96
Client considerations	97
Model Specification	97
Use case: Discover the iSCSI target port capabilities	97
Use case: Identify the iSCSI nodes in a target system (NetworkEntity)	97

Use case: Identify the iSCSI ports on a given iSCSI node.....	97
CIM Elements	97
EMC_VNXe_iSCSICapabilitiesLeaf.....	98
EMC_VNXe_iSCSIProtocolControllerLeaf	99
EMC_VNXe_EthernetPortLeaf	99
EMC_VNXe_iSCSIProtocolEndpointLeaf.....	100
EMC_VNXe_TCPProtocolEndpointForiSCSILeaf	100
EMC_VNXe_IPProtocolEndpointForiSCSILeaf	101
EMC_VNXe_StorageProcessorSystem_iSCSICapabilities_ElementCapabilitiesAssocLeaf.....	101
EMC_VNXe_StorageProcessorSystem_IPProtocolEndpointForiSCSI_HostedAccessPointAssocLeaf	102
EMC_VNXe_StorageProcessorSystem_TCPProtocolEndpointForiSCSI_HostedAccessPointAssocLeaf	102
EMC_VNXe_StorageProcessorSystem_iSCSIProtocolEndpoint_HostedAccessPointAssocLeaf ...	102
EMC_VNXe_iSCSIProtocolEndpoint_iSCSIProtocolController_SAPAvailableForElementAssocLeaf	102
EMC_VNXe_EthernetPort_IPProtocolEndpointForiSCSI_DeviceSAPImplementationAssocLeaf....	103
EMC_VNXe_EthernetPort_iSCSIProtocolEndpoint_DeviceSAPImplementationAssocLeaf	103
EMC_VNXe_iSCSIProtocolEndpoint_TCPProtocolEndpointForiSCSI_BindsToAssocLeaf	103
EMC_VNXe_TCPProtocolEndpointForiSCSI_IPProtocolEndpointForiSCSI_BindsToAssocLeaf	103
Fan Profile	104
Overview.....	104
Class diagram.....	104
Method of the Profile	104
Client considerations	104
Model Specification.....	104
Use case: Discover fan devices.....	104
CIM Element.....	105
EMC_VNXe_FanDevice	105
EMC_VNXe_FanModule_FanDevice_RealizesAssocLeaf	105
EMC_VNXe_StorageSystem_FanDevice_SystemDeviceAssocLeaf	106
Power Supply Profile	107
Overview.....	107
Class diagram.....	107
Method of the Profile	107
Client considerations	107
Model Specification.....	107
Use case: Discover power supplies.....	107

CIM Element	108
EMC_VNXe_PowerSupplyDevice	108
EMC_VNXe_PowerSupplyModule_PowerSupplyDevice_RealizesAssocLeaf	108
EMC_VNXe_StorageSystem_PowerSupplyDevice_SystemDeviceAssocLeaf	109
Health Package	110
Overview	110
Class diagram	110
Method of the Profile	110
Client considerations	110
CIM Element	110
Indication Subprofile	111
Overview	111
Class diagram	111
Method of the Profile	111
Client considerations	112
Model specification	112
Use case: CQL for VNXe profiles	119
Use case: Create IndicationFilter and ListenerDestination instances	119
Use case: Create IndicationSubscription instances	119
Use case: Subscribe to Indications using pre-defined filters	119
CIM Element	120
CIM_AlertIndecation	121
CIM_InstCreation	122
CIM_InstDeletion	122
CIM_InstModification	122
EMC_IndicationFilter	122
EMC_VNXe_BlockIndicationFilterLeaf	123
EMC_VNXe_FileIndicationFilterLeaf	123
EMC_ListenerDestinationCIMXML	124
EMC_IndicationSubscription	124
Job Control Subprofile	125
Overview	125
Class diagram	125
Method of the Profile	125
Client considerations	126
Model specification	126
Use case: Obtain job-affected element	126

CIM Element	126
EMC_VNXe_ConcreteJobLeaf	127
EMC_VNXe_MethodResultLeaf	128
EMC_VNXe_ConcreteJob_MethodResult_AssociatedJobMethodResultAssocLeaf	128
EMC_VNXe_StorageConfigurationService_ConcreteJob_OwningJobElementAssocLeaf	129
EMC_VNXe_ReplicationService_ConcreteJob_OwningJobElementAssocLeaf	129
EMC_VNXe_FileServerConfigurationService_ConcreteJob_OwningJobElementAssocLeaf	129
EMC_VNXe_FileSystemConfigurationService_ConcreteJob_OwningJobElementAssocLeaf	129
EMC_VNXe_FileExportService_ConcreteJob_OwningJobElementAssocLeaf	130
EMC_VNXe_MappedStoragePool_ConcreteJob_AffectedJobElementAssocLeaf	130
EMC_VNXe_StorageVolume_ConcreteJob_AffectedJobElementAssocLeaf	130
EMC_VNXe_SnapVolume_ConcreteJob_AffectedJobElementAssocLeaf	130
EMC_VNXe_SnapGroup_ConcreteJob_AffectedJobElementAssocLeaf	130
EMC_VNXe_CIFSServer_ConcreteJob_AffectedJobElementAssocLeaf	131
EMC_VNXe_NFSServer_ConcreteJob_AffectedJobElementAssocLeaf	131
EMC_VNXe_CIFSShare_ConcreteJob_AffectedJobElementAssocLeaf	131
EMC_VNXe_NFSShare_ConcreteJob_AffectedJobElementAssocLeaf	131
EMC_VNXe_UxfsLocalFileSystem_ConcreteJob_AffectedJobElementAssocLeaf	132
Array Profile	133
Overview	133
Class diagram	134
Method of the Profile	134
Client considerations	135
Use case: Discover Block Server (Array ComputerSystem)	135
Use case: Discover the capacity optimization support in an array	135
CIM Element	136
EMC_VNXe_StorageSystemLeaf	136
EMC_VNXe_ImplementationCapabilitiesLeaf	136
EMC_VNXe_SorageSystem_ImplementationCapabilities_ElementCapabilitiesAssocLeaf	136
Disk Drive Lite Subprofile	137
Overview	137
Class diagram	137
Method of the Profile	137
Client considerations	138
Model Specification	138
CIM Element	138
EMC_VNXe_DiskDriveLeaf	139

EMC_VNXe_DiskExtentLeaf	140
EMC_VNXe_DiskModuleLeaf.....	141
EMC_VNXe_DiskSoftwareIdentityLeaf.....	141
EMC_VNXe_DiskDrive_DiskExtent_MediaPresentAssocLeaf.....	141
EMC_VNXe_DiskSoftwareIdentity_DiskDrive_ElementSoftwareIdentityAssocLeaf	141
EMC_VNXe_StorageSystem_DiskDrive_SystemDeviceAssocLeaf	141
EMC_VNXe_StorageSystem_DiskExtent_SystemDeviceAssocLeaf	142
EMC_VNXe_DiskModule_DiskDrive_RealizesAssocLeaf	142
EMC_VNXe_PrimordialPool_DiskExtent_AssociatedComponentExtentAssocLeaf	142
EMC_VNXe_PrimordialPool_DiskExtent_ConcreteComponentAssocLeaf	142
EMC_VNXe_DiskExtent_DiskRemainingExtent_BasedOnAssocLeaf.....	142
EMC_VNXe_DiskExtent_PoolExtent_BasedOnAssocLeaf.....	143
EMC_VNXe_DAEEnclosureChassis_DiskModule_ContainerAssocLeaf.....	143
EMC_VNXe_DPEEnclosureChassis_DiskModule_ContainerAssocLeaf.....	143
Extent Composition Subprofile	144
Overview.....	144
Class diagram.....	145
Method of the Profile	145
Client considerations	145
Use case: Traverse the virtualization hierarchy of a StorageVolume or LogicalDisk.....	145
Use case: Find the Primordial Extents used by a StorageVolume or LogicalDisk	145
CIM Element.....	146
EMC_VNXe_DiskExtentLeaf	148
EMC_VNXe_PoolExtentLeaf	148
EMC_VNXe_DiskRemainingExtentLeaf.....	149
EMC_VNXe_PoolRemainingExtentLeaf.....	150
EMC_VNXe_StorageSystem_DiskExtent_SystemDeviceAssocLeaf	151
EMC_VNXe_StorageSystem_PoolExtent_SystemDeviceAssocLeaf	151
EMC_VNXe_StorageSystem_DiskRemainingExtent_SystemDeviceAssocLeaf	152
EMC_VNXe_StorageSystem_PoolRemainingExtent_SystemDeviceAssocLeaf	152
EMC_VNXe_DiskExtent_PoolExtent_CompositeExtentBasedOnAssocLeaf	152
EMC_VNXe_DiskExtent_DiskRemainingExtent_BasedOnAssocLeaf.....	152
EMC_VNXe_PoolExtent_PoolRemainingExtent_BasedOnAssocLeaf	152
EMC_VNXe_PoolExtent_LogicalDisk_BasedOnAssocLeaf.....	153
EMC_VNXe_PoolExtent_StorageVolume_BasedOnAssocLeaf	153
EMC_VNXe_PoolExtent_SnapVolume_BasedOnAssocLeaf	153
EMC_VNXe_PoolExtent_PoolGapExtent_BasedOnAssocLeaf.....	153

EMC_VNXe_MappedStoragePool_PoolExtent_AssociatedComponentExtentAssocLeaf	153
EMC_VNXe_PrimordialPool_DiskExtent_AssociatedComponentExtentAssocLeaf	154
EMC_VNXe_MappedStoragePool_PoolRemainingExtent_AssociatedRemainingExtentAssocLeaf	154
EMC_VNXe_PrimordialPool_DiskRemainingExtent_AssociatedRemainingExtentAssocLeaf	154
EMC_VNXe_MappedStoragePool_PoolExtent_ConcreteComponentAssocLeaf	154
EMC_VNXe_MappedStoragePool_PoolRemainingExtent_ConcreteComponentAssocLeaf	154
EMC_VNXe_PrimordialPool_DiskExtent_ConcreteComponentAssocLeaf	155
EMC_VNXe_PrimordialPool_DiskRemainingExtent_ConcreteComponentAssocLeaf	155
Block Services Package	156
Overview	156
Class diagram	156
Methods of the Profile	156
Extrinsic methods on StorageCapabilities	156
Extrinsic methods on StoragePool	161
Extrinsic methods on StorageConfiguration	163
Client considerations	167
Model specification	167
Use case: Discover system's Storage Configuration Capabilities	167
Use case: Discover pool's Storage configuration Capabilities	167
Use case: create StorageVolume (LUN) from StoragePool	168
Use case: Expand a StorageVolume (LUN)	169
Use case: Rename a StorageVolume (LUN)	169
Use case: Delete StorageVolume (LUN)	170
CIM Elements	170
EMC_VNXe_PrimordialPoolLeaf	173
EMC_VNXe_MappedStoragePoolLeaf	173
EMC_VNXe_LogicalDiskLeaf	174
EMC_VNXe_StorageVolumeLeaf	175
EMC_VNXe_StorageConfigurationServiceLeaf	176
EMC_VNXe_PrimordialPoolStorageConfigurationCapabilitiesLeaf	177
EMC_VNXe_MappedStoragePoolStorageConfigurationCapabilitiesLeaf	177
EMC_VNXe_StorageConfigurationCapabilitiesLeaf	178
EMC_VNXe_PrimordialPoolCapabilitiesLeaf	178
EMC_VNXe_MappedStoragePoolCapabilitiesLeaf	179
EMC_VNXe_StorageConfigurationServiceCapabilitiesLeaf	179
EMC_VNXe_GeneratedSettingLeaf	180
EMC_VNXe_StorageSettingLeaf	181

EMC_VNXe_LogicalDiskSettingLeaf	182
EMC_VNXe_StorageVolumeSettingLeaf	182
EMC_VNXe_PrimordialPool_MappedStoragePool_AllocatedFromStoragePoolAssocLeaf	183
EMC_VNXe_MappedStoragePool_LogicalDisk_AllocatedFromStoragePoolAssocLeaf	183
EMC_VNXe_MappedStoragePool_StorageVolume_AllocatedFromStoragePoolAssocLeaf	183
EMC_VNXe_StorageConfigurationService_StorageConfigurationCapabilities_ElementCapabilitiesAssocLeaf	184
EMC_VNXe_PrimordialPool_PrimordialPoolStorageConfigurationCapabilities_ElementCapabilitiesAssocLeaf	184
EMC_VNXe_MappedStoragePool_MappedStoragePoolStorageConfigurationCapabilities_ElementCapabilitiesAssocLeaf	184
EMC_VNXe_StorageConfigurationService_StorageConfigurationServiceCapabilities_ElementCapabilitiesAssocLeaf	184
EMC_VNXe_PrimordialPool_PrimordialPoolCapabilities_ElementCapabilitiesAssocLeaf	185
EMC_VNXe_MappedStoragePool_MappedStoragePoolCapabilities_ElementCapabilitiesAssocLeaf	185
EMC_VNXe_LogicalDisk_LogicalDiskSetting_ElementSettingDataAssocLeaf	185
EMC_VNXe_StorageVolume_StorageVolumeSetting_ElementSettingDataAssocLeaf	185
EMC_VNXe_MappedStoragePoolCapabilities_StorageSetting_StorageSettingsAssociatedToCapabilitiesAssocLeaf	185
EMC_VNXe_MappedStoragePoolCapabilities_GeneratedSetting_StorageSetting_StorageSettingsGeneratedFromCapabilitiesAssocLeaf	186
EMC_VNXe_StorageSystem_MappedStoragePool_HostedStoragePoolAssocLeaf	186
EMC_VNXe_StorageSystem_PrimordialPool_HostedStoragePoolAssocLeaf	186
EMC_VNXe_StorageSystem_LogicalDisk_SystemDeviceAssocLeaf	186
EMC_VNXe_StorageSystem_PoolGapExtent_SystemDeviceAssocLeaf	187
EMC_VNXe_StorageSystem_StorageVolume_SystemDeviceAssocLeaf	187
Thin Provisioning Profile	188
Overview	188
Class diagram	188
Method of the Profile	188
Client considerations	188
Model specification	188
Use case: Creating a Thinly provisioned StorageVolume	188
CIM Element	188
Automated Storage Tiering Profile	189
Overview	189
Class diagram	190
Method of the Profile	190

Client considerations	190
Model Specification	190
Use case: Discover Automated Storage Tiering Capabilities	190
Use case: Create StorageVolume with Storage Tiering	191
Use case: Modify Tiering Policy of StorageVolume	191
CIM Element	192
EMC_VNXe_TierServiceLeaf	193
EMC_VNXe_TierServiceCapabilitiesLeaf	193
EMC_VNXe_StorageTierLeaf	194
EMC_VNXe_TierDomainLeaf	194
EMC_VNXe_StorageTierExtentLeaf	195
EMC_VNXe_StorageSystem_TierService_HostedServiceAssocLeaf	196
EMC_VNXe_TierService_TierServiceCapabilities_ElementCapabilitiesAssocLeaf	196
EMC_VNXe_StorageSystem_TierDomain_SystemComponentAssocLeaf	196
EMC_VNXe_TierService_StorageTier_ServiceAffectsElementAssocLeaf	196
EMC_VNXe_TierService_TierDomain_ServiceAffectsElementAssocLeaf	197
EMC_VNXe_TierDomain_StorageTier_ConcreteDependencyAssocLeaf	197
EMC_VNXe_StorageTier_StorageTierExtent_MemberOfCollectionAssocLeaf	197
EMC_VNXe_StorageTier_StorageVolume_AssociatedElementTierAssocLeaf	197
EMC_VNXe_TierDomain_MappedStoragePool_AssociatedResourcePoolAssocLeaf	198
Copy Service Subprofile	199
Overview	199
Class diagram	199
Method of the Profile	200
Intrinsic methods on StorageConfigurationService	200
Extrinsic methods on ReplicationService	203
Client considerations	204
Model Specification	204
Use case: Discover Copy support and capabilities	204
Use case: Create Standalone-LUN Snapshot	204
Use case: Delete Standalone-LUN Snapshot	205
CIM Element	205
EMC_VNXe_StorageConfigurationServiceLeaf	206
EMC_VNXe_ReplicationServiceLeaf	206
EMC_VNXe_StorageConfigurationCapabilitiesLeaf	206
EMC_VNXe_SnapReplicationCapabilitiesLeaf	206
EMC_VNXe_ReplicationServiceCapabilitiesLeaf	206

EMC_VNXe_StorageConfigurationService_StorageConfigurationCapabilities_ElementCapabilitiesAssocLeaf	206
EMC_VNXe_StorageConfigurationService_StorageReplicationCapabilities_ElementCapabilitiesAssocLeaf	207
EMC_VNXe_ReplicationService_ReplicationServiceCapabilities_ElementCapabilitiesAssocLeaf ..	207
EMC_VNXe_StorageSystem_StorageConfigurationService_HostedServiceAssocLeaf	207
EMC_VNXe_StorageSystem_ReplicationService_HostedServiceAssocLeaf	207
Replication Services Profile	208
Overview	208
Class diagram	208
Services and Capabilities Discovery	208
Element and Group Replication	209
Using StoragePool for Replicas	210
Access and Protocol Endpoint	211
Method of the Profile	211
Extrinsic Methods on Group Management	211
Extrinsic Methods on Replication Management	216
Extrinsic Methods on ReplicationServiceCapabilities	231
Client considerations	237
Model Specification	237
Use case: Disvocer Replication Services Capabilities	237
Use case: Create Promoted Snapshot of Standalone-LUN	237
Use case: Delete Promoted Snapshot of Standalone-LUN	238
Use case: Demote Promoted Snapshot of Standalone-LUN	238
Use case: Create SynchronizationAspect of Storage Element	238
Use case: Delete SynchronizationAspect of Storage Element	239
Use case: Restore Storage Element from SynchronizationAspect	239
Use case: Promote Non-Promoted Snapshot of Storage Element	239
Use case: Create Replication Group	240
Use case: Delete Replication Group	240
Use case: Add LUN to a Replication Group	240
Use case: Remove LUN from Replication Group	240
Use case: Create Promoted Snapshot of Replication Group	240
Use case: Delete Promoted Snapshot of Replication Group	241
Use case: Demote Promoted Snapshot of Replication Group	241
Use case: Create SynchronizationAspect of Storage Group	241
Use case: Delete SynchronizationAspect of Storage Group	242
Use case: Restore Storage Group from SynchronizationAspect	242

Use case: Promote Non-Promoted Snapshot of Storage Group.....	242
Use case: Setup Standalone-LUN Mirror	243
Use case: Detach Standalone-LUN Mirror	243
Use case: Split Standalone-LUN Mirror.....	243
Use case: Resync Standalone-LUN Mirror after Split	244
Use case: Failover Standalone-LUN Mirror	244
Use case: Failback Standalone-LUN Mirror after Failover	244
Use case: Reverse Roles of Standalone-LUN Mirror	244
Use case: Setup Replication Group Mirror	245
Use case: Detach Replication Group Mirror	245
Use case: Split Replication Group Mirror.....	245
Use case: Resync Replication Group Mirror after Split	246
Use case: Failover Replication Group Mirror.....	246
Use case: Failback Replication Group Mirror after Failover	246
Use case: Reverse Roles of Replication Group Mirror.....	246
CIM Element.....	247
EMC_VNXe_ReplicationServiceLeaf.....	250
EMC_VNXe_ReplicationServiceCapabilitiesLeaf	250
EMC_VNXe_SnapReplicationCapabilitiesLeaf	251
EMC_VNXe_MirrorReplicationCapabilitiesLeaf	251
EMC_VNXe_StorageVolumeLeaf.....	252
EMC_VNXe_SnapVolumeLeaf.....	252
EMC_VNXe_ReplicationGroupLeaf.....	253
EMC_VNXe_SnapGroupLeaf	253
EMC_VNXe_SynchronizationAspectForSourceLeaf	254
EMC_VNXe_SynchronizationAspectForSourceGroupLeaf	254
EMC_VNXe_StorageSystem_ReplicationService_HostedServiceAssocLeaf	254
EMC_VNXe_ReplicationService_ReplicationServiceCapabilities_ElementCapabilitiesAssocLeaf ..	255
EMC_VNXe_StorageSystem_ReplicationGroup_HostedCollectionAssocLeaf.....	255
EMC_VNXe_StorageSystem_SnapGroup_HostedCollectionAssocLeaf	255
EMC_VNXe_StorageVolume_SnapVolume_StorageSynchronizedAssocLeaf	255
EMC_VNXe_StorageVolume_StorageVolume_MirrorSynchronizedAssocLeaf	256
EMC_VNXe_StorageVolume_RemoteStorageVolume_MirrorSynchronizedAssocLeaf.....	257
EMC_VNXe_StorageVolume_SynchronizationAspect_SettingsDefineStateAssocLeaf	257
EMC_VNXe_RemoteStorageVolume_StorageVolume_MirrorSynchronizedAssocLeaf.....	258
EMC_VNXe_SnapVolume_SynchronizationAspect_SettingsDefineStateAssocLeaf	258
EMC_VNXe_ReplicationGroup_SnapGroup_GroupSynchronizedAssocLeaf	259

EMC_VNXe_ReplicationGroup_ReplicationGroup_GroupMirrorSynchronizedAssocLeaf	259
EMC_VNXe_ReplicationGroup_SynchronizationAspect_SettingsDefineStateAssocLeaf	260
EMC_VNXe_SnapGroup_SynchronizationAspect_SettingsDefineStateAssocLeaf	260
EMC_VNXe_ReplicationGroup_StorageVolume_OrderedMemberOfCollectionAssocLeaf	260
EMC_VNXe_ReplicationGroup_RemoteStorageVolume_OrderedMemberOfCollectionAssocLeaf	261
EMC_VNXe_SnapGroup_SnapVolume_OrderedMemberOfCollectionAssocLeaf	261
Masking and Mapping Subprofile	262
Overview.....	262
Class diagram.....	263
Method of the Profile	263
Extrinsic Methods on ControllerConfigurationService	263
Extrinsic Methods on StorageHardwareIDManagementService	268
Extrinsic Methods on PrivilegeManagementService	268
Client considerations	268
Model Specification.....	268
Use case: Create a New Host	269
Use case: Register Initiators to a Host	269
Use case: Expose LUs to a Host	269
Use case: Unregister InitiatorPortIDs from a Host.....	270
Use case: Detach LUs from a Host	271
Use case: Delete a Host	272
Use case: Rename a Host.....	272
CIM Element.....	272
EMC_VNXe_ControllerConfigurationServiceLeaf	274
EMC_VNXe_PrivilegeManagementServiceLeaf	274
EMC_VNXe_StorageHardwareIDManagementServiceLeaf	275
EMC_VNXe_ProtocolControllerMaskingCapabilitiesLeaf	275
EMC_VNXe_LunMaskingProtocolControllerLeaf	276
EMC_VNXe_StorageHardwareIDLeaf.....	276
EMC_VNXe_AuthorizedPrivilegeLeaf	277
EMC_VNXe_FCSCSIProtocolEndpointLeaf.....	277
EMC_VNXe_iSCSIProtocolEndpointLeaf.....	277
EMC_VNXe_SystemSpecificCollectionLeaf	277
EMC_VNXe_StorageClientSettingDataLeaf.....	277
EMC_VNXe_StorageSystem_ControllerConfigurationService_HostedServiceAssocLeaf	277
EMC_VNXe_StorageSystem_PrivilegeManagementService_HostedServiceAssocLeaf	278
EMC_VNXe_StorageSystem_StorageHardwareIDManagementService_HostedServiceAssocLeaf.....	278

EMC_VNXe_StorageSystem_ProtocolControllerMaskingCapabilities_ElementCapabilitiesAssocLeaf	278
EMC_VNXe_StorageSystem_SystemSpecificCollection_HostedCollectionAssocLeaf	278
EMC_VNXe_StorageSystem_StorageClientSettingData_ElementSettingDataAssocLeaf	279
EMC_VNXe_SystemSpecificCollection_StorageHardwareID_MemberOfCollectionAssocLeaf	279
EMC_VNXe_LunMaskingProtocolController_AuthorizedPrivilege_AuthorizedTargetAssocLeaf	279
EMC_VNXe_LunMaskingProtocolController_StorageVolume_ProtocolControllerForUnitAssocLeaf	279
EMC_VNXe_StorageHardwareID_LunMaskingProtocolController_AssociatedPrivilegeAssocLeaf	280
EMC_VNXe_StorageHardwareID_AuthorizedPrivilege_AuthorizedSubjectAssocLeaf	280
EMC_VNXe_StorageHardwareID_StorageClientSettingData_ElementSettingDataAssocLeaf	280
EMC_VNXe_iSCSIProtocolEndpoint_LunMaskingProtocolController_SAPAvailableForElementAssocLeaf	280
EMC_VNXe_FCSCSIProtocolEndpoint_LunMaskingProtocolController_SAPAvailableForElementAssocLeaf	281
EMC_VNXe_ControllerConfigurationService_LunMaskingProtocolController_ConcreteDependencyAssocLeaf	281
EMC_VNXe_PrivilegeManagementService_AuthorizedPrivilege_ConcreteDependencyAssocLeaf	281
EMC_VNXe_StorageHardwareIDManagementService_StorageHardwareID_ConcreteDependencyAssocLeaf	281
EMC_VNXe_StorageHardwareIDManagementService_SystemSpecificCollection_ConcreteDependencyAssocLeaf	282
NAS Head Profile	283
Overview	283
Class diagram	284
Method of the Profile	285
Client considerations	285
Use case: Discover NAS Head	285
CIM Element	286
EMC_VNXe_StorageSystemLeaf	287
EMC_VNXe_StorageProcessorSystemLeaf	287
EMC_VNXe_CIFSServerLeaf	287
EMC_VNXe_NFSServerLeaf	287
EMC_VNXe_LogicalDiskLeaf	287
EMC_VNXe_DiskExtentLeaf	287
EMC_VNXe_PrimordialPoolLeaf	287
EMC_VNXe_MappedStoragePoolLeaf	287
EMC_VNXe_UxfsLocalFileSystemLeaf	287
NAS Network Port Profile	288

Overview.....	288
Class diagram.....	289
Methods of the Profile.....	289
Client considerations	289
CIM Elements	290
EMC_VNXe_IPProtocolEndpointForNASLeaf.....	291
EMC_VNXe_TCPProtocolEndpointForNASLeaf	292
EMC_VNXe_CIFSProtocolEndpointLeaf.....	292
EMC_VNXe_NFSProtocolEndpointLeaf.....	292
EMC_VNXe_IPInterfaceSettingDataLeaf	293
EMC_VNXe_NetworkVLANLeaf.....	293
EMC_VNXe_IPProtocolEndpoint_IPInterfaceSettingData_ElementSettingDataAssocLeaf	293
EMC_VNXe_NetworkVLAN_IPProtocolEndpoint_MemberOfCollectionAssocLeaf	293
EMC_VNXe_TCPProtocolEndpointForNAS_CIFSProtocolEndpoint_BindsToAssocLeaf	294
EMC_VNXe_TCPProtocolEndpointForNAS_NFSProtocolEndpoint_BindsToAssocLeaf	294
EMC_VNXe_IPProtocolEndpointForNAS_TCPProtocolEndpointForNAS_BindsToAssocLeaf	294
EMC_VNXe_EthernetPort_CIFSProtocolEndpoint_DeviceSAPImplementationAssocLeaf	294
EMC_VNXe_EthernetPort_NFSProtocolEndpoint_DeviceSAPImplementationAssocLeaf	294
EMC_VNXe_EthernetPort_IPProtocolEndpointForNAS_DeviceSAPImplementationAssocLeaf	295
EMC_VNXe_CIFSServer_CIFSProtocolEndpoint_HostedAccessPointAssocLeaf	295
EMC_VNXe_NFSServer_NFSProtocolEndpoint_HostedAccessPointAssocLeaf.....	295
EMC_VNXe_CIFSServer_TCPProtocolEndpoint_HostedAccessPointAssocLeaf.....	295
EMC_VNXe_NFSServer_TCPProtocolEndpoint_HostedAccessPointAssocLeaf	295
EMC_VNXe_CIFSServer_IPProtocolEndpoint_HostedAccessPointAssocLeaf	296
EMC_VNXe_NFSServer_IPProtocolEndpoint_HostedAccessPointAssocLeaf	296
File Server Manipulation Subprofile.....	297
Overview.....	297
Class diagram.....	298
Methods of the Profile.....	298
Extrinsic Methods on Capabilities.....	298
Extrinsic Methods on FileServerConfigurationService	301
Client considerations	311
Model Specification.....	311
Use case: Create Supported Settings for File Server.....	311
Use case: Create a File Server.....	311
Use case: Modify a File Server.....	312
Use case: Delete a File Server	313

Use case: Add a New IP Interface to an Existing CIFS File Server	313
Use case: Modify an Existing IP Interface of a File Server.....	314
Use case: Delete an Existing IP Interface from a CIFS File Server	315
CIM Elements	316
EMC_VNXe_FileServerConfigurationServiceLeaf	318
EMC_VNXe_FileServerConfigurationCapabilitiesLeaf	319
EMC_VNXe_FileServerCapabilitiesLeaf	319
EMC_VNXe_CIFSServerLeaf.....	320
EMC_VNXe_NFSServerLeaf.....	320
EMC_VNXe_FileServerSettingsLeaf	321
EMC_VNXe_IPInterfaceSettingDataLeaf	321
EMC_VNXe_DNSSettingDataLeaf	321
EMC_VNXe_NISSettingDataLeaf.....	322
EMC_VNXe_CIFSSettingDataLeaf	322
EMC_VNXe_NFSSettingDataLeaf	322
EMC_VNXe_StorageSystem_FileServerConfigurationService_HostedServiceAssocLeaf	323
EMC_VNXe_FileServerConfigurationService_FileServerCapabilities_ElementCapabilitiesAssocLeaf	323
EMC_VNXe_FileServerConfigurationService_FileServerConfigurationCapabilities_ElementCapabilitiesAssocLeaf	323
EMC_VNXe_CIFSServer_FileServerSettings_SettingsDefineStateAssocLeaf	323
EMC_VNXe_NFSServer_FileServerSettings_SettingsDefineStateAssocLeaf	323
EMC_VNXe_FileServerSettings_CIFSSettingData_ConcreteComponentAssocLeaf	324
EMC_VNXe_FileServerSettings_DNSSettingData_ConcreteComponentAssocLeaf	324
EMC_VNXe_FileServerSettings_IPInterfaceSettingData_ConcreteComponentAssocLeaf	324
EMC_VNXe_FileServerSettings_NFSSettingData_ConcreteComponentAssocLeaf	324
EMC_VNXe_FileServerSettings_NISSettingData_ConcreteComponentAssocLeaf	324
EMC_VNXe_FileServerCapabilities_FileServerSettings_SettingsDefineCapabilitiesAssocLeaf.....	325
EMC_VNXe_FileServerCapabilities_CIFSSettingData_SettingsDefineCapabilitiesAssocLeaf	325
EMC_VNXe_FileServerCapabilities_DNSSettingData_SettingsDefineCapabilitiesAssocLeaf.....	325
EMC_VNXe_FileServerCapabilities_IPInterfaceSettingData_SettingsDefineCapabilitiesAssocLeaf	325
EMC_VNXe_FileServerCapabilities_NFSSettingData_SettingsDefineCapabilitiesAssocLeaf	326
EMC_VNXe_FileServerCapabilities_NISSettingData_SettingsDefineCapabilitiesAssocLeaf	326
File Storage Profile.....	327
Overview.....	327
Class diagram.....	327
Methods of the Profile.....	327

Client considerations	327
CIM Element	328
EMC_VNXe_LogicalDisk_UxfsLocalFileSystem_ResidesOnExtentAssocLeaf	328
File System Profile	329
Overview	329
Class diagram	329
Methods of the Profile	330
Client considerations	330
Model Specification	330
Use case: Discover all File Systems Hosted by the Referencing Profile	330
Use case: Discover all Exportable File Systems	330
Use case: Get the Settings of a File System	330
Use case: Get the ComputerSystem that Hosts a File System	330
Use case: Get all File Servers that Have Local Access to a File System and their Access Paths ...	331
Use case: Get the Access Path to a File System on the Specified File Server	331
Use case: Get the Local Access Settings for a File System on the Specified File Server	331
Use case: Get the File Shares of a given File System exported from the specified File Server	331
CIM Elements	331
EMC_VNXe_UxfsLocalFileSystemLeaf	332
EMC_VNXe_FileSystemSettingLeaf	333
EMC_VNXe_LocallyAccessibleFileSystemSettingLeaf	334
EMC_VNXe_UxfsLocalFileSystem_LocallyAccessibleFileSystemSetting_ElementSettingDataAssocLeaf	334
EMC_VNXe_UxfsLocalFileSystem_FileSystemSetting_ElementSettingDataAssocLeaf	334
EMC_VNXe_StorageSystem_UxfsLocalFileSystem_HostedFileSystemAssocLeaf	334
EMC_VNXe_CIFSServer_UxfsLocalFileSystem_LocalAccessAvailableAssocLeaf	335
EMC_VNXe_NFSServer_UxfsLocalFileSystem_LocalAccessAvailableAssocLeaf	335
EMC_VNXe_CIFSServer_LocallyAccessibleFileSystemSetting_HostedDependencyAssocLeaf	335
EMC_VNXe_NFSServer_LocallyAccessibleFileSystemSetting_HostedDependencyAssocLeaf	335
File System Manipulation Subprofile	337
Overview	337
Class diagram	337
Methods of the Profile	338
Extrinsic Methods on Capabilities	338
Extrinsic Methods on FileSystemConfigurationService	340
Client considerations	345
Model Specification	345

Use case: Create Locally Accessible Setting for Filesystem	345
Use case: Create a Locally Accessible Filesystem on a StoragePool	345
Use case: Modify the Name of a Filesystem	346
Use case: Increase the size of a File System.....	347
Use case: Delete a File System and return underlying StorageExtent	348
CIM Elements	348
EMC_VNXe_FileSystemConfigurationServiceLeaf	349
EMC_VNXe_FileSystemConfigurationCapabilitiesLeaf	349
EMC_VNXe_UxfsFileSystemCapabilitiesLeaf	350
EMC_VNXe_LocallyAccessibleFileSystemCapabilitiesLeaf	350
EMC_VNXe_StorageSystem_FileSystemConfigurationService_HostedServiceAssocLeaf	350
EMC_VNXe_CIFSServer_LocallyAccessibleFileSystemCapabilities_HostedDependencyAssocLeaf	350
EMC_VNXe_NFSServer_LocallyAccessibleFileSystemCapabilities_HostedDependencyAssocLeaf	351
EMC_VNXe_FileSystemConfigurationService_FileSystemCapabilities_ElementCapabilitiesAssocLeaf	351
EMC_VNXe_FileSystemConfigurationService_FileSystemConfigurationCapabilities_ElementCapabilitiesAssocLeaf.....	351
EMC_VNXe_FileSystemConfigurationService_LocallyAccessibleFileSystemCapabilities_ElementCapabilitiesAssocLeaf	351
EMC_VNXe_FileSystemCapabilities_FileSystemSetting_SettingsDefineCapabilitiesAssocLeaf.....	352
EMC_VNXe_LocallyAccessibleFileSystemCapabilities_LocallyAccessibleFileSystemSetting_SettingsDefineCapabilitiesAssocLeaf	352
File Export Profile	353
Overview.....	353
Class diagram.....	353
Methods of the Profile.....	353
Client considerations	353
Use case: Discover all File Shares Hosted on a File Server	353
Use case: Discover all File Shares of a File System	354
Use case: Get the Settings of a File Share.....	354
CIM Elements	354
EMC_VNXe_CIFSShareLeaf.....	355
EMC_VNXe_NFSShareLeaf.....	355
EMC_VNXe_CIFSShareSettingLeaf	355
EMC_VNXe_NFSShareSettingLeaf	356
EMC_VNXe_CIFSServer_CIFSShare_HostedShareAssocLeaf.....	356
EMC_VNXe_NFSServer_NFSShare_HostedShareAssocLeaf.....	357

EMC_VNXe_CIFSShare_CIFSShareSetting_ElementSettingDataAssocLeaf	357
EMC_VNXe_NFSShare_NFSShareSetting_ElementSettingDataAssocLeaf	357
EMC_VNXe_CIFSProtocolEndpoint_CIFSShare_SAPAvailableForElementAssocLeaf	357
EMC_VNXe_NFSProtocolEndpoint_NFSShare_SAPAvailableForElementAssocLeaf	357
EMC_VNXe_UxfsLocalFileSystem_CIFSShare_SharedElementAssocLeaf	358
EMC_VNXe_UxfsLocalFileSystem_NFSShare_SharedElementAssocLeaf	358
File Export Manipulation Subprofile	359
Overview.....	359
Class Diagram	359
Methods of the Profile.....	360
Extrinsic Methods on Capabilities.....	360
Extrinsic Methods on FileExportService	362
Client considerations	366
Use case: Check the Supported Capabilities Pattern of a FileExportService	366
Use case: Create Settings for File Share	366
Use case: Create a File Share.....	366
Use case: Modify the Settings of a CIFS File Share	367
Use case: Release an Exported File Share.....	368
CIM Elements	369
EMC_VNXe_FileExportServiceLeaf	370
EMC_VNXe_FileExportCapabilitiesLeaf.....	371
EMC_VNXe_CIFSShareCapabilitiesLeaf	371
EMC_VNXe_NFSShareCapabilitiesLeaf	371
EMC_VNXe_CIFSShareSettingLeaf	371
EMC_VNXe_NFSShareSettingLeaf	372
EMC_VNXe_CIFSServer_FileExportService_HostedServiceAssocLeaf	372
EMC_VNXe_NFSServer_FileExportService_HostedServiceAssocLeaf	372
EMC_VNXe_FileExportService_FileExportCapabilities_ElementCapabilitiesAssocLeaf	372
EMC_VNXe_FileExportService_CIFSShareCapabilities_ElementCapabilitiesAssocLeaf.....	372
EMC_VNXe_FileExportService_NFSShareCapabilities_ElementCapabilitiesAssocLeaf.....	372
EMC_VNXe_FileExportService_CIFSShare_ServiceAffectsElementAssocLeaf	373
EMC_VNXe_FileExportService_NFSShare_ServiceAffectsElementAssocLeaf	373
EMC_VNXe_CIFSShareCapabilities_CIFSShareSetting_SettingsDefineCapabilitiesAssocLeaf.....	373
EMC_VNXe_NFSShareCapabilities_NFSShareSetting_SettingsDefineCapabilitiesAssocLeaf.....	373
Simple Identity Management Profile	375
Overview.....	375
Class diagram.....	375

Method of the Profile	375
Intrinsic/Extrinsic Methods on UserContact Manipulation	375
Intrinsic/Extrinsic Methods on Privilege Manipulation.....	378
Client consideration	380
Model Specification	380
Use case: Create UserContact and Identities for a domain user	381
Use case: Get the UserContact for a specified domain user whose UserContact has been created	382
Use case: Removal of an existing UserContact	383
Use case: Get the Privilege and UserContact by an Exported File Share	383
Use case: Assign a privilege to a pair of UserContact and Exported FileShare	383
Use case: Modify privilege to a pair of UserContact and Exported FileShare.....	384
Use case: Removal of the existing privilege	384
CIM Element.....	384
EMC_VNXe_AccountManagementServiceLeaf	385
EMC_VNXe_UserContactLeaf	385
EMC_VNXe_IdentityLeaf	385
EMC_VNXe_StorageSystem_AccountManagementService_HostedServiceAssocLeaf	386
EMC_VNXe_AccountManagementService_Identity_ServiceAffectsElementAssocLeaf	386
EMC_VNXe_UserContact_Identity_AssignedIdentityAssocLeaf	386
EMC_VNXe_CIFSShare_Identity_AssociatedPrivilegeAssocLeaf	386
Appendix A: Known Issues in Third Party Integration	387
Microsoft System Center 2012 Virtual Machine Manager (SCVMM)	387

Table of Figures

Figure 1 - Server Profile class diagram.....	51
Figure 2 - Flowchart of viewing Server Profile implementation.....	52
Figure 3 - Profile Registration Profile class diagram.....	59
Figure 4 - Flowchart of enumerating autonomous profiles	61
Figure 5 - Flowchart of determining the SNIA version of an SMI-S Profile.....	62
Figure 6 - Multiple Computer System Subprofile class diagram.....	65
Figure 7 - Flowchart of find top-level ComputerSystem for any LogicalDevice	68
Figure 8 - Software Subprofile class diagram	72
Figure 9 - Physical Package Package class diagram.....	74
Figure 10 - Flowchart of getting product information for a ComputerSystem	76
Figure 11 - Access Points Subprofile class diagram	90
Figure 12 - FC Target Ports Subprofile class diagram	92
Figure 13 - iSCSI Target Ports Subprofile class diagram	96
Figure 14 - Fan Profile class diagram	104
Figure 15 - Power Supply Profile class diagram	107
Figure 16 - Indication Subprofile class diagram.....	111
Figure 17 - Flowchart of Indication subscription	120
Figure 18 - Job Control Subprofile class diagram.....	125
Figure 19- Array Profile class diagram.....	134
Figure 20 - Flowchart of Block Server discovery	135
Figure 21 - Disk Drive Lite Subprofile class diagram.....	137
Figure 22 - Extent Composition Subprofile class diagram.....	145
Figure 23 - Flowchart of finding Primordial Extents used by a StorageVolume or LogicalDisk.....	146
Figure 24 - Block Services Package class diagram.....	156
Figure 25 - Flowchart of creating LUN from StoragePool.....	168
Figure 26 - Flowchart of expanding a LUN	169
Figure 27 - Flowchart of Renaming a LUN	170
Figure 28 - Automated Storage Tiering class diagram	190
Figure 29 - Copy Services Subprofile Class Diagram	200
Figure 31 - Replication Services and Capabilities Discovery	208
Figure 32 - Element and Group Replication.....	209
Figure 33 - Using StoragePool for Replicas.....	210
Figure 34 - Replication Services Access and Protocol Endpoint.....	211
Figure 35 - Masking and Mapping Subprofile Class Diagram	263
Figure 36 - Flowchart of Unregistering Initiators from Host	270

Figure 37 - Flowchart of Detaching LUs from Host.....	271
Figure 38 - NAS Head Profile class diagram	284
Figure 39 - Flowchart of NAS Head discovery.....	285
Figure 40 - NAS Network Port Profile class diagram	289
Figure 41 - File Server Manipulation Subprofile class diagram	298
Figure 42 - Flowchart of Creating File Server	312
Figure 43 - Flowchart of Modifying File Server	313
Figure 44 - Flowchart of Adding New IP Interface to a File Server.....	314
Figure 45 - Flowchart of Modifying an Existing IP Interface of the File Server.....	315
Figure 46 - Flowchart of Deleting an Existing IP Interface from a File Server.....	316
Figure 47 - File Storage Profile Class Diagram	327
Figure 48 - File System Profile Class Diagram	330
Figure 49 - File System Manipulation Subprofile Class Diagram	337
Figure 50 - Flowchart of Creating a File System.....	346
Figure 51 - Flowchart of Modifying Filesystem Name.....	347
Figure 52 - Flowchart of Extending a File System	347
Figure 53 - File Export Profile Class Diagram.....	353
Figure 54 - File Export Manipulation Subprofile Class Diagram	359
Figure 55 - Flowchart of Creating Exported File Share	367
Figure 56 - Flowchart of Modifying the Settings of a CIFS File Share.....	368
Figure 57 - Flowchart of Releasing an Exported File Share	369
Figure 58 - Simple Identity Management Instance diagram	375
Figure 59 - Flowchart of creating UserContact	382
Figure 60 - Flowchart of getting UserContact	382
Figure 61 - Flowchart of getting Privilege and UserContact by an Exported FileShare	383
Figure 62 - Flowchart of assigning privilege to Exported FileShare	384
Figure 63 - Storage Pool Selection in Different Wizards	387
Figure 64 - Import Certificate Dialog	388

Table of Tables

Table 1 - CIM Elements for Server Profile	53
Table 2 - Referenced properties/methods for ECOM_ObjectManager	53
Table 3 - Referenced properties/methods for ECOM_RegisteredProfile	54
Table 4 - Referenced properties/methods for ECOM_System	55
Table 5 - Referenced properties/methods for ECOM_Namespace	55
Table 6 - Referenced properties/methods for ECOM_CIMXMLCommunicationMechanism	56
Table 7 - Referenced properties/methods for ECOM_ElementConformsToProfile	57
Table 8 - Referenced properties/methods for ECOM_HostedService	57
Table 9 - Referenced properties/methods for ECOM_CommMechanismForManager	57
Table 10 - Referenced properties/methods for ECOM_NamespaceInManager	58
Table 11 - CIM elements for Profile Registration profile	62
Table 12 - Referenced properties/methods for ECOM_RegisteredSubProfile	63
Table 13 - Referenced properties/methods for ECOM_ReferencedProfile	63
Table 14 - Referenced properties/methods for ECOM_SubProfileRequiresProfile	63
Table 15 - Value map of RedundancyStatus	66
Table 16 - CIM Elements for Multiple Computer System Subprofile	68
Table 17 - Referenced properties/methods for EMC_VNXe_StorageSystemLeaf	69
Table 18 - Referenced properties/methods for EMC_VNXe_StorageProcessorSystemLeaf	69
Table 19 - Referenced properties/methods for EMC_VNXe_RedundancySetLeaf	70
Table 20 - Referenced properties/methods for EMC_VNXe_StorageSystem_StorageProcessorSystem_ComponentCSAssocLeaf	70
Table 21 - Referenced properties/methods for EMC_VNXe_StorageSystem_RedundancySet_ConcretIdentityAssocLeaf	71
Table 22 - Referenced properties/methods for EMC_VNXe_RedundancySet_StorageProcessorSystem_MemberOfCollectionAssocLeaf	71
Table 23 - CIM Elements for Software Subprofile	73
Table 24 - Referenced properties/methods for EMC_VNXe_StorageSystemSoftwareIdentityLeaf	73
Table 25 - Referenced properties/methods for EMC_VNXe_StorageSystem_StorageSystemSoftwareIdentity_InstalledSoftwareIdentityAssocLeaf	73
Table 26 - CIM Elements for Physical Package Package	76
Table 27 - Referenced properties/methods for EMC_VNXe_ArrayChassisLeaf	79
Table 28 - Referenced properties/methods for EMC_VNXe_DAEEnclosureChassisLeaf	79
Table 29 - Referenced properties/methods for EMC_VNXe_DPEEnclosureChassisLeaf	80
Table 30 - Referenced properties/methods for EMC_VNXe_SPChassisLeaf	80
Table 31 - Referenced properties/methods for EMC_VNXe_DiskModuleLeaf	80

Table 32 - Referenced properties/methods for EMC_VNXe_FanModuleLeaf	81
Table 33 - Referenced properties/methods for EMC_VNXe_PowerSupplyModuleLeaf	81
Table 34 - Referenced properties/methods for EMC_VNXe_ArrayChassisProductLeaf.....	82
Table 35 - Referenced properties/methods for EMC_VNXe_DAEEnclosureChassisProductLeaf.....	82
Table 36 - Referenced properties/methods for EMC_VNXe_DPEEnclosureChassisProductLeaf.....	82
Table 37 - Referenced properties/methods for EMC_VNXe_SPChassisProductLeaf	82
Table 38 - Referenced properties/methods for EMC_VNXe_DiskProductLeaf	83
Table 39 - Referenced properties/methods for EMC_VNXe_FanProductLeaf.....	83
Table 40 - Referenced properties/methods for EMC_VNXe_PowerSupplyProductLeaf.....	83
Table 41 - Referenced properties/methods for EMC_VNXe_StorageSystem_ArrayChassis_SystemPackagingAssocLeaf.....	84
Table 42 - Referenced properties/methods for EMC_VNXe_StorageProcessorSystem_SPChassis_SystemPackagingAssocLeaf	84
Table 43 - Referenced properties/methods for EMC_VNXe_ArrayChassis_DAEEnclosureChassis_ContainerAssocLeaf	84
Table 44 - Referenced properties/methods for EMC_VNXe_ArrayChassis_DPEEnclosureChassis_ContainerAssocLeaf	84
Table 45 - Referenced properties/methods for EMC_VNXe_DAEEnclosureChassis_DiskModule_ContainerAssocLeaf	84
Table 46 - Referenced properties/methods for EMC_VNXe_DAEEnclosureChassis_FanModule_ContainerAssocLeaf.....	85
Table 47 - Referenced properties/methods for EMC_VNXe_DAEEnclosureChassis_PowerSupplyModule_ContainerAssocLeaf.....	85
Table 48 - Referenced properties/methods for EMC_VNXe_DPEEnclosureChassis_DiskModule_ContainerAssocLeaf	85
Table 49 - Referenced properties/methods for EMC_VNXe_DPEEnclosureChassis_FanModule_ContainerAssocLeaf.....	85
Table 50 - Referenced properties/methods for EMC_VNXe_DPEEnclosureChassis_PowerSupplyModule_ContainerAssocLeaf.....	85
Table 51 - Referenced properties/methods for EMC_VNXe_DPEEnclosureChassis_SPChassis_ContainerAssocLeaf	86
Table 52 - Referenced properties/methods for EMC_VNXe_ArrayChassis_ArrayChassisProduct_ProductPhysicalComponentAssocLeaf.....	86
Table 53 - Referenced properties/methods for EMC_VNXe_DAEEnclosureChassis_DAEEnclosureChassisProduct_ProductPhysicalComponentAssocLe af	86
Table 54 - Referenced properties/methods for EMC_VNXe_DPEEnclosureChassis_DPEEnclosureChassisProduct_ProductPhysicalComponentAssocLe af	86
Table 55 - Referenced properties/methods for EMC_VNXe_SPChassis_SPChassisProduct_ProductPhysicalComponentAssocLeaf	87
Table 56 - Referenced properties/methods for EMC_VNXe_DiskModule_DiskProduct_ProductPhysicalComponentAssocLeaf.....	87

Table 57 - Referenced properties/methods for EMC_VNXe_FanModule_FanProduct_ProductPhysicalComponentAssocLeaf	87
Table 58 - Referenced properties/methods for EMC_VNXe_PowerSupplyModule_PowerSupplyProduct_ProductPhysicalComponentAssocLeaf	87
Table 59 - Referenced properties/methods for EMC_VNXe_ArrayChassisProduct_DAEEnclosureChassisProduct_ProductParentChildAssocLeaf	87
Table 60 - Referenced properties/methods for EMC_VNXe_ArrayChassisProduct_DPEEnclosureChassisProduct_ProductParentChildAssocLeaf	88
Table 61 - Referenced properties/methods for EMC_VNXe_DAEEnclosureChassisProduct_DiskProduct_ProductParentChildAssocLeaf	88
Table 62 - Referenced properties/methods for EMC_VNXe_DAEEnclosureChassisProduct_FanProduct_ProductParentChildAssocLeaf	88
Table 63 - Referenced properties/methods for EMC_VNXe_DAEEnclosureChassisProduct_PowerSupplyProduct_ProductParentChildAssocLeaf	88
Table 64 - Referenced properties/methods for EMC_VNXe_DPEEnclosureChassisProduct_DiskProduct_ProductParentChildAssocLeaf	89
Table 65 - Referenced properties/methods for EMC_VNXe_DPEEnclosureChassisProduct_FanProduct_ProductParentChildAssocLeaf	89
Table 66 - Referenced properties/methods for EMC_VNXe_DPEEnclosureChassisProduct_PowerSupplyProduct_ProductParentChildAssocLeaf	89
Table 67 - Referenced properties/methods for EMC_VNXe_DPEEnclosureChassisProduct_SPChassisProduct_ProductParentChildAssocLeaf	89
Table 68 - CIM Elements for Access Points Subprofile	90
Table 69 - Referenced properties/methods for EMC_VNXe_RemoteServiceAccessPointLeaf	91
Table 70 - Referenced properties/methods for EMC_VNXe_StorageSystem_RemoteAccessPoint_HostedAccessPointLeaf	91
Table 71 - CIM Elements for FC Target Port Subprofile	93
Table 72 - Referenced properties/methods for EMC_VNXe_FCPortLeaf	93
Table 73 - Referenced properties/methods for EMC_VNXe_FCSCSIProtocolEndpointLeaf	94
Table 74 - Referenced properties/methods for EMC_VNXe_StorageProcessorSystem_FCPort_SystemDeviceAssocLeaf	94
Table 75 - Referenced properties/methods for EMC_VNXe_StorageProcessorSystem_FCSCSIProtocolEndpoint_HostedAccessPointAssocLeaf	94
Table 76 - Referenced properties/methods for EMC_VNXe_FCPort_FCSCSIProtocolEndpoint_DeviceSAPImplementationAssocLeaf	95
Table 77 - CIM Elements for iSCSI Target Port	97
Table 78 - Referenced properties/methods for EMC_VNXe_iSCSICapabilitiesLeaf	98
Table 79 - Referenced properties/methods for EMC_VNXe_iSCSIProtocolControllerLeaf	99
Table 80 - Referenced properties/methods for EMC_VNXe_EthernetPortLeaf	99
Table 81 - Referenced properties/methods for EMC_VNXe_iSCSIProtocolEndpointLeaf	100
Table 82 - Referenced properties/methods for EMC_VNXe_TCPProtocolEndpointForiSCSILeaf	100
Table 83 - Referenced properties/methods for EMC_VNXe_IPProtocolEndpointForiSCSILeaf	101

Table 84 - Referenced properties/methods for EMC_VNXe_StorageProcessorSystem_iSCSICapabilities_ElementCapabilitiesAssocLeaf	101
Table 85 - Referenced properties/methods for EMC_VNXe_StorageProcessorSystem_IPProtocolEndpointForiSCSI_HostedAccessPointAssocLeaf ..	102
Table 86 - Referenced properties/methods for EMC_VNXe_StorageProcessorSystem_TCPProtocolEndpointForiSCSI_HostedAccessPointAssocLeaf	102
Table 87 - Referenced properties/methods for EMC_VNXe_StorageProcessorSystem_iSCSIProtocolEndpoint_HostedAccessPointAssocLeaf	102
Table 88 - Referenced properties/methods for EMC_VNXe_iSCSIProtocolEndpoint_iSCSIProtocolController_SAPAvailableForElementAssocLeaf	102
Table 89 - Referenced properties/methods for EMC_VNXe_EthernetPort_IPProtocolEndpointForiSCSI_DeviceSAPImplementationAssocLeaf	103
Table 90 - Referenced properties/methods for EMC_VNXe_EthernetPort_iSCSIProtocolEndpoint_DeviceSAPImplementationAssocLeaf.....	103
Table 91 - Referenced properties/methods for EMC_VNXe_iSCSIProtocolEndpoint_TCPProtocolEndpointForiSCSI_BindsToAssocLeaf	103
Table 92 - Referenced properties/methods for EMC_VNXe_TCPProtocolEndpointForiSCSI_IPProtocolEndpointForiSCSI_BindsToAccocLeaf	103
Table 93 - CIM Elements for Fan	105
Table 94 - Referenced properties/methods for EMC_VNXe_FanDeviceLeaf	105
Table 95 - Referenced properties/methods for EMC_VNXe_FanModule_FanDevice_RealizesAssocLeaf	105
Table 96 - Referenced properties/methods for EMC_VNXe_StorageSystem_FanDevice_SystemDeviceAssocLeaf.....	106
Table 97 - CIM Elements for Power Supply	108
Table 98 - Referenced properties/methods for EMC_VNXe_PowerSupplyDeviceLeaf	108
Table 99 - Referenced properties/methods for EMC_VNXe_PowerSupplyModule_PowerSupplyDevice_RealizesAssocLeaf	108
Table 100 - Referenced properties/methods for EMC_VNXe_StorageSystem_PowerSupplyDevice_SystemDeviceAssocLeaf.....	109
Table 101 - Supported Intrinsic Methods of Indication Profile	112
Table 102 - Supported pre-defined Indication filters	112
Table 103 - Supported alert Indications	115
Table 104 - Supported life cycle Indications	115
Table 105 - CIM Element in Indication Profile.....	120
Table 106 - Referenced properties/methods for CIM_AlertIndication	121
Table 107 - Referenced properties/methods for CIM_InstCreation	122
Table 108 - Referenced properties/methods for CIM_InstDeletion	122
Table 109 - Referenced properties/methods for CIM_InstModification	122
Table 110 - Referenced properties/methods for EMC_IndicationFilter	122
Table 111 - Referenced properties/methods for EMC_VNXe_BlockIndicationFilterLeaf	123

Table 112 - Referenced properties/methods for EMC_VNXe_FileIndicationFilterLeaf	123
Table 113 - Referenced properties/methods for EMC_ListenerDestinationCIMXML	124
Table 114 - Referenced properties/methods for EMC_IndicationSubscription.....	124
Table 115 - CIM Elements for Job Control Subprofile	126
Table 116 - Referenced properties/methods for EMC_VNXe_ConcreteJobLeaf	127
Table 117 - Referenced properties/methods for EMC_VNXe_MethodResultLeaf	128
Table 118 - Referenced properties/methods for EMC_VNXe_ConcreteJob_MethodResult_AssociatedJobMethodResultAssocLeaf	128
Table 119 - Referenced properties/methods for EMC_VNXe_StorageConfigurationService_ConcreteJob_OwningJobElementAssocLeaf	129
Table 120 - Referenced properties/methods for EMC_VNXe_ReplicationService_ConcreteJob_OwningJobElementAssocLeaf	129
Table 121 - Referenced properties/methods for EMC_VNXe_FileServerConfigurationService_ConcreteJob_OwningJobElementAssocLeaf	129
Table 122 - Referenced properties/methods for EMC_VNXe_FileSystemConfigurationService_ConcreteJob_OwningJobElementAssocLeaf	129
Table 123 - Referenced properties/methods for EMC_VNXe_FileExportService_ConcreteJob_OwningJobElementAssocLeaf.....	130
Table 124 - Referenced properties/methods for EMC_VNXe_MappedStoragePool_ConcreteJob_AffectedJobElementAssocLeaf	130
Table 125 - Referenced properties/methods for EMC_VNXe_StorageVolume_ConcreteJob_AffectedJobElementAssocLeaf	130
Table 126 - Referenced properties/methods for EMC_VNXe_SnapVolume_ConcreteJob_AffectedJobElementAssocLeaf	130
Table 127 - Referenced properties/methods for EMC_VNXe_SnapGroup_ConcreteJob_AffectedJobElementAssocLeaf	130
Table 128 - Referenced properties/methods for EMC_VNXe_CIFSServer_ConcreteJob_AffectedJobElementAssocLeaf	131
Table 129 - Referenced properties/methods for EMC_VNXe_NFSServer_ConcreteJob_AffectedJobElementAssocLeaf	131
Table 130 - Referenced properties/methods for EMC_VNXe_CIFSShare_ConcreteJob_AffectedJobElementAssocLeaf	131
Table 131 - Referenced properties/methods for EMC_VNXe_NFSShare_ConcreteJob_AffectedJobElementAssocLeaf	131
Table 132 - Referenced properties/methods for EMC_VNXe_UxfsLocalFileSystem_ConcreteJob_AffectedJobElementAssocLeaf	132
Table 133 - CIM Elements for Array Profile	136
Table 134 - Referenced properties/methods for EMC_VNXe_ImplementationCapabilitiesLeaf	136
Table 135 - Referenced properties/methods for EMC_VNXe_SorageSystem_ImplementationCapabilities_ElementCapabilitiesAssocLeaf	136
Table 136 - CIM Elements for Disk Drive Lite Subprofile	138
Table 137 - Referenced properties/methods for EMC_VNXe_DiskDriveLeaf	139

Table 138 - Referenced properties/methods for EMC_VNXe_DiskExtentLeaf	140
Table 139 - Referenced properties/methods for EMC_VNXe_DiskSoftwareIdentityLeaf	141
Table 140 - Referenced properties/methods for EMC_VNXe_DiskDrive_DiskExtent_MediaPresentAssocLeaf	141
Table 141 - Referenced properties/methods for EMC_VNXe_DiskSoftwareIdentity_DiskDrive_ElementSoftwareIdentityAssocLeaf	141
Table 142 - Referenced properties/methods for EMC_VNXe_StorageSystem_DiskDrive_SystemDeviceAssocLeaf.....	141
Table 143 - Referenced properties/methods for EMC_VNXe_StorageSystem_DiskExtent_SystemDeviceAssocLeaf.....	142
Table 144 - Referenced properties/methods for EMC_VNXe_DiskModule_DiskDrive_RealizesAssocLeaf	142
Table 145 - Referenced properties/methods for EMC_VNXe_PrimordialPool_DiskExtent_AssociatedComponentExtentAssocLeaf	142
Table 146 - Referenced properties/methods for EMC_VNXe_PrimordialPool_DiskExtent_ConcreteComponentAssocLeaf.....	142
Table 147 - Referenced properties/methods for EMC_VNXe_DiskExtent_DiskRemainingExtent_BasedOnAssocLeaf	142
Table 148 - Referenced properties/methods for EMC_VNXe_DiskExtent_PoolExtent_BasedOnAssocLeaf	143
Table 149 - CIM Elements for Extent Composition Subprofile	146
Table 150 - Referenced properties/methods for EMC_VNXe_PoolExtentLeaf	148
Table 151 - Referenced properties/methods for EMC_VNXe_DiskRemainingExtentLeaf	149
Table 152 - Referenced properties/methods for EMC_VNXe_PoolRemainingExtentLeaf	150
Table 153 - Referenced properties/methods for EMC_VNXe_StorageSystem_DiskExtent_SystemDeviceAssocLeaf.....	151
Table 154 - Referenced properties/methods for EMC_VNXe_StorageSystem_PoolExtent_SystemDeviceAssocLeaf	151
Table 155 - Referenced properties/methods for EMC_VNXe_StorageSystem_DiskRemainingExtent_SystemDeviceAssocLeaf	152
Table 156 - Referenced properties/methods for EMC_VNXe_StorageSystem_PoolRemainingExtent_SystemDeviceAssocLeaf	152
Table 157 - Referenced properties/methods for EMC_VNXe_DiskExtent_PoolExtent_CompositeExtentBasedOnAssocLeaf	152
Table 158 - Referenced properties/methods for EMC_VNXe_DiskExtent_DiskRemainingExtent_BasedOnAssocLeaf	152
Table 159 - Referenced properties/methods for EMC_VNXe_PoolExtent_PoolRemainingExtent_BasedOnAssocLeaf	152
Table 160 - Referenced properties/methods for EMC_VNXe_PoolExtent_LogicalDisk_BasedOnAssocLeaf	153
Table 161 - Referenced properties/methods for EMC_VNXe_PoolExtent_StorageVolume_BasedOnAssocLeaf	153

Table 162 - Referenced properties/methods for EMC_VNXe_PoolExtent_SnapVolume_BasedOnAssocLeaf.....	153
Table 163 - Referenced properties/methods for EMC_VNXe_PoolExtent_PoolGapExtent_BasedOnAssocLeaf	153
Table 164 - Referenced properties/methods for EMC_VNXe_MappedStoragePool_PoolExtent_AssociatedComponentExtentAssocLeaf.....	153
Table 165 - Referenced properties/methods for EMC_VNXe_PrimordialPool_DiskExtent_AssociatedComponentExtentAssocLeaf	154
Table 166 - Referenced properties/methods for EMC_VNXe_MappedStoragePool_PoolRemainingExtent_AssociatedRemainingExtentAssocLeaf	154
Table 167 - Referenced properties/methods for EMC_VNXe_PrimordialPool_DiskRemainingExtent_AssociatedRemainingExtentAssocLeaf	154
Table 168 - Referenced properties/methods for EMC_VNXe_MappedStoragePool_PoolExtent_ConcreteComponentAssocLeaf	154
Table 169 - Referenced properties/methods for EMC_VNXe_MappedStoragePool_PoolRemainingExtent_ConcreteComponentAssocLeaf.....	154
Table 170 - Referenced properties/methods for EMC_VNXe_PrimordialPool_DiskExtent_ConcreteComponentAssocLeaf.....	155
Table 171 - Referenced properties/methods for EMC_VNXe_PrimordialPool_DiskRemainingExtent_ConcreteComponentAssocLeaf	155
Table 172 - Extrinsic methods on StorageCapabilities	157
Table 173 - Signature and parameters of EMC_VNXe_MappedStoragePoolCapabilitiesLeaf.GetSupportedStripeLength	157
Table 174 - Possible return code of EMC_VNXe_MappedStoragePoolCapabilitiesLeaf.GetSupportedStripeLengths	157
Table 175 - Signature and parameters of EMC_VNXe_MappedStoragePoolCapabilitiesLeaf.GetSupportedParityLayouts.....	158
Table 176 - Possible return code of EMC_VNXe_MappedStoragePoolCapabilitiesLeaf.GetSupportedParityLayouts.....	158
Table 177 - Signature and parameters of EMC_VNXe_StorageConfigurationServiceCapabilitiesLeaf.CreateSetting.....	158
Table 178 - Possible return code of EMC_VNXe_StorageConfigurationServiceCapabilitiesLeaf.CreateSetting.....	158
Table 179 - Signature and parameters of EMC_VNXe_PrimordialPoolCapabilitiesLeaf.GetSupportedStripeLengths.....	159
Table 180 - Possible return code of EMC_VNXe_PrimordialPoolCapabilitiesLeaf.GetSupportedStripeLengths.....	159
Table 181 - Signature and parameters of EMC_VNXe_PrimordialPoolCapabilitiesLeaf.GetSupportedParityLayouts	159
Table 182 - Possible return code of EMC_VNXe_PrimordialPoolCapabilitiesLeaf.GetSupportedParityLayouts	159
Table 183 - Signature and parameters of EMC_VNXe_StorageConfigurationServiceCapabilitiesLeaf.GetSupportedStripeLengths	160

Table 184 - Possible return code of EMC_VNXe_StorageConfigurationServiceCapabilitiesLeaf.GetSupportedStripeLengths	160
Table 185 - Signature and parameters of EMC_VNXe_StorageConfigurationServiceCapabilitiesLeaf.GetSupportedParityLayouts	160
Table 186 - Possible return code of EMC_VNXe_StorageConfigurationServiceCapabilitiesLeaf.GetSupportedParityLayouts	160
Table 187 - Extrinsic methods on StoragePool.....	161
Table 188 - Signature and parameters of EMC_VNXe_MappedStoragePoolLeaf.GetAvailableExtents .	161
Table 189 - Possible return code of EMC_VNXe_MappedStoragePoolLeaf.GetAvailableExtents	161
Table 190 – Signature and parameters of EMC_VNXe_MappedStoragePoolLeaf.GetSupportedSizeRange.....	161
Table 191 - Possible return code of EMC_VNXe_MappedStoragePoolLeaf.GetSupportedSizeRange ..	162
Table 192 - Signature and parameters of EMC_VNXe_PrimalPoolLeaf.GetAvailableExtents	162
Table 193 - Possible return code of EMC_VNXe_PrimalPoolLeaf.GetAvailableExtents	163
Table 194- Extrinsic methods on StorageConfiguration	163
Table 195 - Signature and parameters of EMC_VNXe_StorageConfigurationServiceLeaf.CreateOrModifyElementFromStoragePool	163
Table 196 - Possible return code of EMC_VNXe_StorageConfigurationServiceLeaf.CreateOrModifyElementFromStoragePool	165
Table 197 - Signature and parameters of EMC_VNXe_StorageConfigurationServiceLeaf.ReturnToStoragePool	166
Table 198 - Possible return code of EMC_VNXe_StorageConfigurationServiceLeaf.ReturnToStoragePool	166
Table 199 - CIM Elements implemented on the VNXe for Block Services Package	170
Table 200 - Referenced properties/methods for EMC_VNXe_PrimalPoolLeaf	173
Table 201 - Referenced properties/methods for EMC_VNXe_MappedStoragePoolLeaf	173
Table 202 - Referenced properties/methods for EMC_VNXe_LogicalDiskLeaf	174
Table 203 - Referenced properties/methods for EMC_VNXe_StorageVolumeLeaf.....	175
Table 204 - Referenced properties/methods for EMC_VNXe_StorageConfigurationServiceLeaf	176
Table 205 - Referenced properties/methods for EMC_VNXe_PrimalPoolStorageConfigurationCapabilitiesLeaf	177
Table 206 - Referenced properties/methods for EMC_VNXe_MappedStoragePoolStorageConfigurationCapabilitiesLeaf	177
Table 207 - Referenced properties/methods for EMC_VNXe_StorageConfigurationCapabilitiesLeaf.....	178
Table 208 - Referenced properties/methods for EMC_VNXe_PrimalPoolCapabilitiesLeaf	178
Table 209 - Referenced properties/methods for EMC_VNXe_MappedStoragePoolCapabilitiesLeaf.....	179
Table 210 - Referenced properties/methods for EMC_VNXe_StorageConfigurationServiceCapabilitiesLeaf	179
Table 211 - Referenced properties/methods for EMC_VNXe_GeneratedSettingLeaf	180
Table 212 - Referenced properties/methods for EMC_VNXe_StorageSettingLeaf.....	181
Table 213 - Referenced properties/methods for EMC_VNXe_LogicalDiskSettingLeaf.....	182

Table 214 - Referenced properties/methods for EMC_VNXe_StorageVolumeSettingLeaf	182
Table 215 - Referenced properties/methods for EMC_VNXe_PrimordialPool_MappedStoragePool_AllocatedFromStoragePoolAssocLeaf	183
Table 216 - Referenced properties/methods for EMC_VNXe_MappedStoragePool_LogicalDisk_AllocatedFromStoragePoolAssocLeaf	183
Table 217 - Referenced properties/methods for EMC_VNXe_MappedStoragePool_StorageVolume_AllocatedFromStoragePoolAssocLeaf.....	183
Table 218 - Referenced properties/methods for EMC_VNXe_StorageConfigurationService_StorageConfigurationCapabilities_ElementCapabilitiesAssocL eaf	184
Table 219 - Referenced properties/methods for EMC_VNXe_PrimordialPool_PrimordialPoolStorageConfigurationCapabilities_ElementCapabilitiesAssoc Leaf	184
Table 220 - Referenced properties/methods for EMC_VNXe_MappedStoragePool_MappedStoragePoolStorageConfigurationCapabilities_ElementCapabi litiesAssocLeaf	184
Table 221 - Referenced properties/methods for EMC_VNXe_StorageConfigurationService_StorageConfigurationServiceCapabilities_ElementCapabilities AssocLeaf.....	184
Table 222 - Referenced properties/methods for EMC_VNXe_PrimordialPool_PrimordialPoolCapabilities_ElementCapabilitiesAssocLeaf	185
Table 223 - Referenced properties/methods for EMC_VNXe_MappedStoragePool_MappedStoragePoolCapabilities_ElementCapabilitiesAssocLeaf ...	185
Table 224 - Referenced properties/methods for EMC_VNXe_LogicalDisk_LogicalDiskSetting_ElementSettingDataAssocLeaf	185
Table 225 - Referenced properties/methods for EMC_VNXe_StorageVolume_StorageVolumeSetting_ElementSettingDataAssocLeaf	185
Table 226 - Referenced properties/methods for EMC_VNXe_MappedStoragePoolCapabilities_StorageSetting_StorageSettingsAssociatedToCapabilities AssocLeaf.....	185
Table 227 - Referenced properties/methods for EMC_VNXe_MappedStoragePoolCapabilities_GeneratedSetting_StorageSettingsGeneratedFromCapabil itiesAssocLeaf	186
Table 228 - Referenced properties/methods for EMC_VNXe_StorageSystem_MappedStoragePool_HostedStoragePoolAssocLeaf.....	186
Table 229 - Referenced properties/methods for EMC_VNXe_StorageSystem_PrimordialPool_HostedStoragePoolAssocLeaf	186
Table 230 - Referenced properties/methods for EMC_VNXe_StorageSystem_LogicalDisk_SystemDeviceAssocLeaf	186
Table 231 - Referenced properties/methods for EMC_VNXe_StorageSystem_PoolGapExtent_SystemDeviceAssocLeaf.....	187
Table 232 - Referenced properties/methods for EMC_VNXe_StorageSystem_StorageVolume_SystemDeviceAssocLeaf	187
Table 233 - CIM Elements for Automated Storage Tiering.....	192
Table 234 - Referenced properties/methods for EMC_VNXe_TierServiceLeaf	193

Table 235 - Referenced properties/methods for EMC_VNXe_TierServiceCapabilitiesLeaf	193
Table 236 - Referenced properties/methods for EMC_VNXe_StorageTierLeaf.....	194
Table 237 - Referenced properties/methods for EMC_VNXe_TierDomainLeaf.....	194
Table 238 - Referenced properties/methods for EMC_VNXe_StorageTierExtentLeaf	195
Table 239 - Referenced properties/methods for EMC_VNXe_StorageSystem_TierService_HostedServiceAssocLeaf	196
Table 240 - Referenced properties/methods for EMC_VNXe_TierService_TierServiceCapabilities_ElementCapabilitiesAssocLeaf.....	196
Table 241 - Referenced properties/methods for EMC_VNXe_StorageSystem_TierDomain_SystemComponentAssocLeaf.....	196
Table 242 - Referenced properties/methods for EMC_VNXe_TierService_StorageTier_ServiceAffectsElementAssocLeaf	196
Table 243 - Referenced properties/methods for EMC_VNXe_TierService_TierDomain_ServiceAffectsElementAssocLeaf	197
Table 244 - Referenced properties/methods for EMC_VNXe_TierDomain_StorageTier_ConcreteDependencyAssocLeaf	197
Table 245 - Referenced properties/methods for EMC_VNXe_StorageTier_StorageTierExtent_MemberOfCollectionAssocLeaf.....	197
Table 246 - Referenced properties/methods for EMC_VNXe_StorageTier_StorageVolume_AssociatedElementTierAssocLeaf.....	197
Table 247 - Referenced properties/methods for EMC_VNXe_TierDomain_MappedStoragePool_AssociatedResourcePoolAssocLeaf.....	198
Table 248 - Extrinsic methods on StorageConfigurationService	200
Table 249 - Signature and parameters of EMC_VNXe_StorageConfigurationServiceLeaf.CreateReplica	201
Table 250 - Possible return code of EMC_VNXe_StorageConfigurationServiceLeaf.CreateReplica	202
Table 251 - Signature and parameters of EMC_VNXe_StorageConfigurationServiceLeaf.ModifySynchronization	203
Table 252 - Possible return code of EMC_VNXe_StorageConfigurationServiceLeaf.ModifySynchronization	203
Table 253 - Extrinsic methods on ReplicationService	203
Table 254 - CIM Elements for Copy Services Subprofile	205
Table 255 - Referenced properties/methods for EMC_VNXe_StorageConfigurationService_StorageConfigurationCapabilities_ElementCapabilitiesAssocL eaf	206
Table 256 - Referenced properties/methods for EMC_VNXe_StorageConfigurationService_SnapReplicationCapabilities_ElementCapabilitiesAssocLeaf	207
Table 257 - Referenced properties/methods for EMC_VNXe_ReplicationService_ReplicationServiceCapabilities_ElementCapabilitiesAssocLeaf	207
Table 258 - Referenced properties/methods for EMC_VNXe_StorageSystem_StorageConfigurationService_HostedServiceAssocLeaf	207

Table 259 - Referenced properties/methods for EMC_VNXe_StorageSystem_ReplicationService_HostedServiceAssocLeaf.....	207
Table 260 - Signature and parameters of EMC_VNXe_ReplicationServiceLeaf.CreateGroup.....	211
Table 261 - Possible return code of EMC_VNXe_ReplicationServiceLeaf.CreateGroup	212
Table 262 - Signature and Parameters of EMC_VNXe_ReplicationServiceLeaf.DeleteGroup.....	213
Table 263 - Possible return code of EMC_VNXe_ReplicationServiceLeaf.DeleteGroup	213
Table 264 - Signature and Parameters of EMC_VNXe_ReplicationServiceLeaf.AddMembers.....	214
Table 265 - Possible return code of EMC_VNXe_ReplicationServiceLeaf.AddMembers	214
Table 266 - Signature and Parameters of EMC_VNXe_ReplicationServiceLeaf.RemoveMembers.....	215
Table 267 - Possible return code of EMC_VNXe_ReplicationServiceLeaf.RemoveMembers	216
Table 268 - Signature and Parameters of EMC_VNXe_ReplicationServiceLeaf.GetPeerSystems	216
Table 269 - Possible return code of EMC_VNXe_ReplicationServiceLeaf.GetPeerSystems	217
Table 270 - Signature and Parameters of EMC_VNXe_ReplicationServiceLeaf.GetServiceAccessPoints	217
Table 271 - Possible return code of EMC_VNXe_ReplicationServiceLeaf.GetServiceAccessPoints.....	217
Table 272 - Signature and Parameters of EMC_VNXe_ReplicationServiceLeaf.GetReplicationRelationshipInstances.....	218
Table 273 - Possible return code of EMC_VNXe_ReplicationServiceLeaf.GetReplicationRelationshipInstances.....	218
Table 274 - Signature and Parameters of EMC_VNXe_ReplicationServiceLeaf.GetReplicationRelationships	219
Table 275 - Possible return code of EMC_VNXe_ReplicationServiceLeaf.GetReplicationRelationships	219
Table 276 - Signature and Parameters of EMC_VNXe_ReplicationServiceLeaf.GetAvailableTargetElements.....	220
Table 277 - Possible return code of EMC_VNXe_ReplicationServiceLeaf.GetAvailableTargetElements	220
Table 278 - Signature and Parameters of EMC_VNXe_ReplicationServiceLeaf.CreateElementReplica	221
Table 279 - Possible return code of EMC_VNXe_ReplicationServiceLeaf.CreateElementReplica	223
Table 280 - Signature and Parameters of EMC_VNXe_ReplicationServiceLeaf.CreateGroupReplica ...	223
Table 281 - Possible return code of EMC_VNXe_ReplicationServiceLeaf.CreateGroupReplica	225
Table 282 - Signature and Parameters of EMC_VNXe_ReplicationServiceLeaf.ModifyReplicaSynchronization	226
Table 283 - Possible return code of EMC_VNXe_ReplicationServiceLeaf.ModifyReplicaSynchronization	227
Table 284 - Signature and Parameters of EMC_VNXe_ReplicationServiceLeaf.CreateSynchronizationAspect	228
Table 285 - Possible return code of EMC_VNXe_ReplicationServiceLeaf.CreateSynchronizationAspect	229
Table 286 - Signature and Parameters of EMC_VNXe_ReplicationServiceLeaf.ModifySettingsDefineState	229
Table 287 - Possible return code of EMC_VNXe_ReplicationServiceLeaf.ModifySettingsDefineState ...	230

Table 288 - Signature and Parameters of EMC_VNXe_ReplicationServiceCapabilities.ConvertReplicationTypeToSyncType	231
Table 289 - Possible return code of EMC_VNXe_ReplicationServiceCapabilities.ConvertReplicationTypeToSyncType	231
Table 290 - Signature and Parameters of EMC_VNXe_ReplicationServiceCapabilities.ConvertSyncTypeToReplicationType	232
Table 291 - Possible return code of EMC_VNXe_ReplicationServiceCapabilities.ConvertSyncTypeToReplicationType	232
Table 292 - Signature and Parameters of EMC_VNXe_ReplicationServiceCapabilities.GetDefaultConsistency	232
Table 293 - Possible return code of EMC_VNXe_ReplicationServiceCapabilities.GetDefaultConsistency	233
Table 294 - Signature and Parameters of EMC_VNXe_ReplicationServiceCapabilities.GetDefaultGroupPersistency	233
Table 295 - Possible return code of EMC_VNXe_ReplicationServiceCapabilities.GetDefaultGroupPersistency	233
Table 296 - Signature and Parameters of EMC_VNXe_ReplicationServiceCapabilities.GetSupportedConsistency	233
Table 297 - Possible return code of EMC_VNXe_ReplicationServiceCapabilities.GetSupportedConsistency	234
Table 298 - Signature and Parameters of EMC_VNXe_ReplicationServiceCapabilities.GetSupportedCopyStates	234
Table 299 - Possible return code of EMC_VNXe_ReplicationServiceCapabilities.GetSupportedCopyStates	234
Table 300 - Signature and Parameters of EMC_VNXe_ReplicationServiceCapabilities.GetSupportedFeatures	235
Table 301 - Possible return code of EMC_VNXe_ReplicationServiceCapabilities.GetSupportedFeatures	235
Table 302 - Signature and Parameters of EMC_VNXe_ReplicationServiceCapabilities.GetSupportedGroupCopyStates	235
Table 303 - Possible return code of EMC_VNXe_ReplicationServiceCapabilities.GetSupportedGroupCopyStates	235
Table 304 - Signature and Parameters of EMC_VNXe_ReplicationServiceCapabilities.GetSupportedGroupFeatures	236
Table 305 - Possible return code of EMC_VNXe_ReplicationServiceCapabilities.GetSupportedGroupFeatures	236
Table 306 - Signature and Parameters of EMC_VNXe_ReplicationServiceCapabilities.GetSupportedOperations	236
Table 307 - Possible return code of EMC_VNXe_ReplicationServiceCapabilities.GetSupportedOperations	237
Table 308 - CIM Elements implemented in VNXe for Replication Services Profile	247
Table 309 - Referenced properties/methods for EMC_VNXe_ReplicationServiceLeaf	250
Table 310 - Referenced properties/methods for EMC_VNXe_ReplicationServiceCapabilitiesLeaf	250
Table 311 - Referenced properties/methods for EMC_VNXe_SnapReplicationCapabilitiesLeaf	251

Table 312 - Referenced properties/methods for EMC_VNXe_MirrorReplicationCapabilitiesLeaf	251
Table 313 - Referenced properties/methods for EMC_VNXe_SnapVolumeLeaf	252
Table 314 - Referenced properties/methods for EMC_VNXe_ReplicationGroupLeaf	253
Table 315 - Referenced properties/methods for EMC_VNXe_SnapGroupLeaf	253
Table 316 - Referenced properties/methods for EMC_VNXe_SynchronizationAspectForSourceLeaf	254
Table 317 - Referenced properties/methods for EMC_VNXe_SynchronizationAspectForSourceGroupLeaf	254
Table 318 - Referenced properties/methods for EMC_VNXe_StorageSystem_ReplicationService_HostedServiceAssocLeaf.....	254
Table 319 - Referenced properties/methods for EMC_VNXe_ReplicationService_ReplicationServiceCapabilities_ElementCapabilitiesAssocLeaf	255
Table 320 - Referenced properties/methods for EMC_VNXe_StorageSystem_ReplicationGroup_HostedCollectionAssocLeaf	255
Table 321 - Referenced properties/methods for EMC_VNXe_StorageSystem_SnapGroup_HostedCollectionAssocLeaf	255
Table 322 - Referenced properties/methods for EMC_VNXe_StorageVolume_SnapVolume_StorageSynchronizedAssocLeaf.....	255
Table 323 - Referenced properties/methods for EMC_VNXe_StorageVolume_StorageVolume_MirrorSynchronizedAssocLeaf.....	256
Table 324 - Referenced properties/methods for EMC_VNXe_StorageVolume_RemoteStorageVolume_MirrorSynchronizedAssocLeaf	257
Table 325 - Referenced properties/methods for EMC_VNXe_StorageVolume_SynchronizationAspect_SettingsDefineStateAssocLeaf	257
Table 326 - Referenced properties/methods for EMC_VNXe_RemoteStorageVolume_StorageVolume_MirrorSynchronizedAssocLeaf	258
Table 327 - Referenced properties/methods for EMC_VNXe_SnapVolume_SynchronizationAspect_SettingsDefineStateAssocLeaf	258
Table 324 - Referenced properties/methods for EMC_VNXe_ReplicationGroup_SnapGroup_GroupSynchronizedAssocLeaf	259
Table 329 - Referenced properties/methods for EMC_VNXe_ReplicationGroup_ReplicationGroup_GroupMirrorSynchronizedAssocLeaf	259
Table 330 - Referenced properties/methods for EMC_VNXe_ReplicationGroup_SynchronizationAspect_SettingsDefineStateAssocLeaf	260
Table 331 - Referenced properties/methods for EMC_VNXe_SnapGroup_SynchronizationAspect_SettingsDefineStateAssocLeaf.....	260
Table 326 - Referenced properties/methods for EMC_VNXe_ReplicationGroup_StorageVolume_OrderedMemberOfCollectionAssocLeaf	260
Table 333 - Referenced properties/methods for EMC_VNXe_ReplicationGroup_RemoteStorageVolume_OrderedMemberOfCollectionAssocLeaf	261
Table 327 - Referenced properties/methods for EMC_VNXe_SnapGroup_SnapVolume_OrderedMemberOfCollectionAssocLeaf	261
Table 312 - Signature and Parameters of EMC_VNXe_ControllerConfigurationServiceLeaf.ExposePaths	263

Table 313 - ExposePaths Use Cases and Input Parameter Constraints.....	264
Table 314 - Possible return code of EMC_VNXe_ControllerConfigurationServiceLeaf.ExposePaths.....	265
Table 315 - Signature and Parameters of EMC_VNXe_ControllerConfigurationServiceLeaf.ExposePaths	266
Table 316 - HidePaths Use Cases and Input Parameter Constraints	266
Table 317 - Possible return code of EMC_VNXe_ControllerConfigurationServiceLeaf.HidePaths	267
Table 318 - Signature and Parameters of EMC_VNXe_ControllerConfigurationServiceLeaf.ExposePaths	267
Table 319 - Possible return code of EMC_VNXe_ControllerConfigurationServiceLeaf.HidePaths	268
Table 320 - CIM Elements implemented in VNXe for Masking and Mapping Subprofile	272
Table 321 - Referenced properties/methods for EMC_VNXe_ControllerConfigurationServiceLeaf	274
Table 322 - Referenced properties/methods for EMC_VNXe_PrivilegeManagementServiceLeaf	274
Table 323 - Referenced properties/methods for EMC_VNXe_StorageHardwareIDManagementServiceLeaf.....	275
Table 324 - Referenced properties/methods for EMC_VNXe_ProtocolControllerMaskingCapabilitiesLeaf	275
Table 325 - Referenced properties/methods for EMC_VNXe_LunMaskingProtocolControllerLeaf	276
Table 326 - Referenced properties/methods for EMC_VNXe_StorageHardwareIDLeaf	276
Table 327 - Referenced properties/methods for EMC_VNXe_AuthorizedPrivilegeLeaf	277
Table 328 - Referenced properties/methods for EMC_VNXe_SystemSpecificCollectionLeaf.....	277
Table 329 - Referenced properties/methods for EMC_VNXe_StorageClientSettingDataLeaf.....	277
Table 330 - Referenced properties/methods for EMC_VNXe_StorageSystem_ControllerConfigurationService_HostedServiceAssocLeaf	277
Table 331 - Referenced properties/methods for EMC_VNXe_StorageSystem_PrivilegeManagementService_HostedServiceAssocLeaf	278
Table 332 - Referenced properties/methods for EMC_VNXe_StorageSystem_StorageHardwareIDManagementService_HostedServiceAssocLeaf	278
Table 333 - Referenced properties/methods for EMC_VNXe_StorageSystem_ProtocolControllerMaskingCapabilities_ElementCapabilitiesAssocLeaf ..	278
Table 334 - Referenced properties/methods for EMC_VNXe_StorageSystem_SystemSpecificCollection_HostedCollectionAssocLeaf	278
Table 335 - Referenced properties/methods for EMC_VNXe_StorageSystem_StorageClientSettingData_ElementSettingDataAssocLeaf	279
Table 336 - Referenced properties/methods for EMC_VNXe_SystemSpecificCollection_StorageHardwareID_MemberOfCollectionAssocLeaf	279
Table 337 - Referenced properties/methods for EMC_VNXe_LunMaskingProtocolController_AuthorizedPrivilege_AuthorizedTargetAssocLeaf	279
Table 338 - Referenced properties/methods for EMC_VNXe_LunMaskingProtocolController_StorageVolume_ProtocolControllerForUnitAssocLeaf.....	279
Table 339 - Referenced properties/methods for EMC_VNXe_StorageHardwareID_LunMaskingProtocolController_AssociatedPrivilegeAssocLeaf.....	280

Table 340 - Referenced properties/methods for EMC_VNXe_StorageHardwareID_AuthorizedPrivilege_AuthorizedSubjectAssocLeaf	280
Table 341 - Referenced properties/methods for EMC_VNXe_StorageHardwareID_StorageClientSettingData_ElementSettingDataAssocLeaf	280
Table 342 - Referenced properties/methods for EMC_VNXe_iSCSIProtocolEndpoint_LunMaskingProtocolController_SAPAvailableForElementAssocLeaf	280
Table 343 - Referenced properties/methods for EMC_VNXe_FCSCSIProtocolEndpoint_LunMaskingProtocolController_SAPAvailableForElementAssocLeaf	281
Table 344 - Referenced properties/methods for EMC_VNXe_ControllerConfigurationService_LunMaskingProtocolController_ConcreteDependencyAssocLeaf	281
Table 345 - Referenced properties/methods for EMC_VNXe_PrivilegeManagementService_AuthorizedPrivilege_ConcreteDependencyAssocLeaf.....	281
Table 346 - Referenced properties/methods for EMC_VNXe_StorageHardwareIDManagementService_StorageHardwareID_ConcreteDependencyAssocLeaf	281
Table 347 - Referenced properties/methods for EMC_VNXe_StorageHardwareIDManagementService_SystemSpecificCollection_ConcreteDependencyAssocLeaf	282
Table 348 - CIM Elements for NAS Head Profile	286
Table 349 - CIM Elements for NAS Network Port Profile	290
Table 350 - Referenced properties/methods for EMC_VNXe_IPProtocolEndpointForNASLeaf.....	291
Table 351 - Referenced properties/methods for EMC_VNXe_TCIPProtocolEndpointForNASLeaf	292
Table 352 - Referenced properties/methods for EMC_VNXe_CIFSProtocolEndpointLeaf	292
Table 353 - Referenced properties/methods for EMC_VNXe_NFSProtocolEndpointLeaf	292
Table 354 - Referenced properties/methods for EMC_VNXe_NetworkVLANLeaf	293
Table 355 - Referenced properties/methods for EMC_VNXe_IPProtocolEndpoint_IPInterfaceSettingData_ElementSettingDataAssocLeaf	293
Table 356 - Referenced properties/methods for EMC_VNXe_NetworkVLAN_IPProtocolEndpoint_MemberOfCollectionAssocLeaf	293
Table 357 - Referenced properties/methods for EMC_VNXe_TCIPProtocolEndpointForNAS_CIFSProtocolEndpoint_BindsToAssocLeaf	294
Table 358 - Referenced properties/methods for EMC_VNXe_TCIPProtocolEndpointForNAS_NFSProtocolEndpoint_BindsToAssocLeaf	294
Table 359 - Referenced properties/methods for EMC_VNXe_IPProtocolEndpointForNAS_TCIPProtocolEndpointForNAS_BindsToAssocLeaf	294
Table 360 - Referenced properties/methods for EMC_VNXe_EthernetPort_CIFSProtocolEndpoint_DeviceSAPImplementationAssocLeaf	294
Table 361 - Referenced properties/methods for EMC_VNXe_EthernetPort_NFSProtocolEndpoint_DeviceSAPImplementationAssocLeaf	294
Table 362 - Referenced properties/methods for EMC_VNXe_EthernetPort_IPProtocolEndpointForNAS_DeviceSAPImplementationAssocLeaf	295

Table 363 - Referenced properties/methods for EMC_VNXe_CIFSServer_CIFSProtocolEndpoint_HostedAccessPointAssocLeaf	295
Table 364 - Referenced properties/methods for EMC_VNXe_NFSServer_NFSProtocolEndpoint_HostedAccessPointAssocLeaf	295
Table 365 - Referenced properties/methods for EMC_VNXe_CIFSServer_TCPProtocolEndpoint_HostedAccessPointAssocLeaf	295
Table 366 - Referenced properties/methods for EMC_VNXe_NFSServer_TCPProtocolEndpoint_HostedAccessPointAssocLeaf	295
Table 367 - Referenced properties/methods for EMC_VNXe_CIFSServer_IPProtocolEndpoint_HostedAccessPointAssocLeaf	296
Table 368 - Referenced properties/methods for EMC_VNXe_NFSServer_IPProtocolEndpoint_HostedAccessPointAssocLeaf	296
Table 369 - Default setting data instances used in EMC_VNXe_FileServerCapabilities.CreateGoalSetting	299
Table 370 - Signature and parameters of EMC_VNXe_FileServerCapabilitiesLeaf.CreateGoalSettings	300
Table 371 - Possible return code of EMC_VNXe_FileServerCapabilitiesLeaf.CreateGoalSettings	300
Table 372 - Signature and Parameters of EMC_VNXe_FileServerConfigurationServiceLeaf.CreateFileServer	302
Table 373 - Possible return code of EMC_VNXe_FileServerConfigurationServiceLeaf.CreateFileServer	303
Table 374 - Signature and Parameters of EMC_VNXe_FileServerConfigurationServiceLeaf.ModifyFileServer	304
Table 375 - Possible return code of EMC_VNXe_FileServerConfigurationServiceLeaf.ModifyFileServer	305
Table 376 - Signature and Parameters of EMC_VNXe_FileServerConfigurationServiceLeaf.DeleteFileServer	305
Table 377 - Possible return code of EMC_VNXe_FileServerConfigurationServiceLeaf.DeleteFileServer	306
Table 378 - Signature and Parameters of EMC_VNXe_FileServerConfigurationServiceLeaf.AddIPInterface	306
Table 379 - Possible return code of EMC_VNXe_FileServerConfigurationServiceLeaf.AddIPInterface .	307
Table 380 - Signature and Parameters of EMC_VNXe_FileServerConfigurationServiceLeaf.ModifyIPInterface	308
Table 381 - Possible return code of EMC_VNXe_FileServerConfigurationServiceLeaf.ModifyIPInterface	309
Table 382 - Signature and Parameters of EMC_VNXe_FileServerConfigurationServiceLeaf.DeleteIPInterface	309
Table 383 - Possible return code of EMC_VNXe_FileServerConfigurationServiceLeaf.DeleteIPInterface	310
Table 384 - CIM Elements for File Server Manipulation Subprofile	316
Table 385 - Referenced properties/methods for EMC_VNXe_FileServerConfigurationServiceLeaf	318
Table 386 - Referenced properties/methods for EMC_VNXe_FileServerConfigurationCapabilitiesLeaf .	319
Table 387 - Referenced properties/methods for EMC_VNXe_FileServerCapabilitiesLeaf	319

Table 388 - Referenced properties/methods for EMC_VNXe_CIFSServerLeaf.....	320
Table 389 - Referenced properties/methods for EMC_VNXe_NFSServerLeaf.....	320
Table 390 - Referenced properties/methods for EMC_VNXe_FileServerSettingsLeaf.....	321
Table 391 - Referenced properties/methods for EMC_VNXe_IPInterfaceSettingDataLeaf.....	321
Table 392 - Referenced properties/methods for EMC_VNXe_DNSSettingDataLeaf.....	321
Table 393 - Referenced properties/methods for EMC_VNXe_NISSettingDataLeaf.....	322
Table 394 - Referenced properties/methods for EMC_VNXe_CIFSSettingDataLeaf.....	322
Table 395 - Referenced properties/methods for EMC_VNXe_NFSSettingDataLeaf.....	322
Table 396 - Referenced properties/methods for EMC_VNXe_StorageSystem_FileServerConfigurationService_HostedServiceAssocLeaf.....	323
Table 397 - Referenced properties/methods for EMC_VNXe_FileServerConfigurationService_FileServerCapabilities_ElementCapabilitiesAssocLeaf...	323
Table 398 - Referenced properties/methods for EMC_VNXe_FileServerConfigurationService_FileServerConfigurationCapabilities_ElementCapabilitiesAs socLeaf.....	323
Table 399 - Referenced properties/methods for EMC_VNXe_CIFSServer_FileServerSettings_SettingsDefineStateAssocLeaf.....	323
Table 400 - Referenced properties/methods for EMC_VNXe_NFSServer_FileServerSettings_SettingsDefineStateAssocLeaf.....	323
Table 401 - Referenced properties/methods for EMC_VNXe_FileServerSettings_CIFSSettingData_ConcreteComponentAssocLeaf.....	324
Table 402 - Referenced properties/methods for EMC_VNXe_FileServerSettings_DNSSettingData_ConcreteComponentAssocLeaf.....	324
Table 403 - Referenced properties/methods for EMC_VNXe_FileServerSettings_IPInterfaceSettingData_ConcreteComponentAssocLeaf.....	324
Table 404 - Referenced properties/methods for EMC_VNXe_FileServerSettings_NFSSettingData_ConcreteComponentAssocLeaf.....	324
Table 405 - Referenced properties/methods for EMC_VNXe_FileServerSettings_NISSettingData_ConcreteComponentAssocLeaf.....	324
Table 406 - Referenced properties/methods for EMC_VNXe_FileServerCapabilities_FileServerSettings_SettingsDefineCapabilitiesAssocLeaf.....	325
Table 407 - Referenced properties/methods for EMC_VNXe_FileServerCapabilities_CIFSSettingData_SettingsDefineCapabilitiesAssocLeaf.....	325
Table 408 - Referenced properties/methods for EMC_VNXe_FileServerCapabilities_DNSSettingData_SettingsDefineCapabilitiesAssocLeaf.....	325
Table 409 - Referenced properties/methods for EMC_VNXe_FileServerCapabilities_IPInterfaceSettingData_SettingsDefineCapabilitiesAssocLeaf.....	325
Table 410 - Referenced properties/methods for EMC_VNXe_FileServerCapabilities_NFSSettingData_SettingsDefineCapabilitiesAssocLeaf.....	326
Table 411 - Referenced properties/methods for EMC_VNXe_FileServerCapabilities_NISSettingData_SettingsDefineCapabilitiesAssocLeaf.....	326
Table 412 - CIM Elements for File Storage Profile.....	328

Table 413 - Referenced properties/methods for EMC_VNXe_LogicalDisk_UxfsLocalFileSystem_ResidesOnExtentAssocLeaf	328
Table 414 - CIM Elements for File System Profile	331
Table 415 - Referenced properties/methods for EMC_VNXe_UxfsLocalFileSystemLeaf	332
Table 416 - Referenced properties/methods for EMC_VNXe_FileSystemSettingLeaf	333
Table 417 - Referenced properties/methods for EMC_VNXe_LocallyAccessibleFileSystemSettingLeaf	334
Table 418 - Referenced properties/methods for EMC_VNXe_UxfsLocalFileSystem_LocallyAccessibleFileSystemSetting_ElementSettingDataAssocLeaf	334
Table 419 - Referenced properties/methods for EMC_VNXe_UxfsLocalFileSystem_FileSystemSetting_ElementSettingDataAssocLeaf	334
Table 420 - Referenced properties/methods for EMC_VNXe_StorageSystem_UxfsLocalFileSystem_HostedFileSystemAssocLeaf	334
Table 421 - Referenced properties/methods for EMC_VNXe_CIFSServer_UxfsLocalFileSystem_LocalAccessAvailableAssocLeaf	335
Table 422 - Referenced properties/methods for EMC_VNXe_NFSServer_UxfsLocalFileSystem_LocalAccessAvailableAssocLeaf	335
Table 423 - Referenced properties/methods for EMC_VNXe_CIFSServer_LocallyAccessibleFileSystemSetting_HostedDependencyAssocLeaf	335
Table 424 - Referenced properties/methods for EMC_VNXe_NFSServer_LocallyAccessibleFileSystemSetting_HostedDependencyAssocLeaf	335
Table 425 - Signature and Parameters of EMC_VNXe_UxfsFileSystemCapabilitiesLeaf.CreateGoalSettings	338
Table 426 - Possible return code of EMC_VNXe_UxfsFileSystemCapabilitiesLeaf.CreateGoalSettings	338
Table 427 - Signature and Parameters of EMC_VNXe_LocallyAccessibleFileSystemCapabilitiesLeaf.CreateGoalSettings	339
Table 428 - Possible return code of EMC_VNXe_LocallyAccessibleFileSystemCapabilitiesLeaf.CreateGoalSettings	339
Table 429 - Signature and Parameters of EMC_VNXe_FileSystemConfigurationServiceLeaf.SNIA_CreateFileSystem	340
Table 430 - Possible return code of EMC_VNXe_FileSystemConfigurationServiceLeaf.SNIA_CreateFileSystem	342
Table 431 - Signature and Parameters of EMC_VNXe_FileSystemConfigurationServiceLeaf.SNIA_ModifyFileSystem	342
Table 432 - Possible return code of EMC_VNXe_FileSystemConfigurationServiceLeaf.SNIA_ModifyFileSystem	344
Table 433 - Signature and Parameters of EMC_VNXe_FileSystemConfigurationServiceLeaf.DeleteFileSystem	344
Table 434 - Possible return code of EMC_VNXe_FileSystemConfigurationServiceLeaf.DeleteFileSystem	345
Table 435 - CIM Elements for File System Manipulation Subprofile	348
Table 436 - Referenced properties/methods for EMC_VNXe_FileSystemConfigurationServiceLeaf	349
Table 437 - Referenced properties/methods for EMC_VNXe_FileSystemConfigurationCapabilitiesLeaf	349

Table 438 - Referenced properties/methods for EMC_VNXe_UxfsFileSystemCapabilitiesLeaf.....	350
Table 439 - Referenced properties/methods for EMC_VNXe_LocallyAccessibleFileSystemCapabilitiesLeaf.....	350
Table 440 - Referenced properties/methods for EMC_VNXe_StorageSystem_FileSystemConfigurationService_HostedServiceAssocLeaf	350
Table 441 - Referenced properties/methods for EMC_VNXe_CIFSServer_LocallyAccessibleFileSystemCapabilities_HostedDependencyAssocLeaf	350
Table 442 - Referenced properties/methods for EMC_VNXe_NFSServer_LocallyAccessibleFileSystemCapabilities_HostedDependencyAssocLeaf	351
Table 443 - Referenced properties/methods for EMC_VNXe_FileSystemConfigurationService_FileSystemCapabilities_ElementCapabilitiesAssocLeaf	351
Table 444 - Referenced properties/methods for EMC_VNXe_FileSystemConfigurationService_FileSystemConfigurationCapabilities_ElementCapabilities AssocLeaf.....	351
Table 445 - Referenced properties/methods for EMC_VNXe_FileSystemConfigurationService_LocallyAccessibleFileSystemCapabilities_ElementCapabili tiesAssocLeaf	351
Table 446 - Referenced properties/methods for EMC_VNXe_FileSystemCapabilities_FileSystemSetting_SettingsDefineCapabilitiesAssocLeaf	352
Table 447 - Referenced properties/methods for EMC_VNXe_LocallyAccessibleFileSystemCapabilities_LocallyAccessibleFileSystemSetting_SettingsDefi neCapabilitiesAssocLeaf	352
Table 448 - CIM Elements for File Export Profile.....	354
Table 449 - Referenced properties/methods for EMC_VNXe_CIFSShareLeaf.....	355
Table 450 - Referenced properties/methods for EMC_VNXe_NFSShareLeaf.....	355
Table 451 - Referenced properties/methods for EMC_VNXe_CIFSShareSettingLeaf	355
Table 452 - Referenced properties/methods for EMC_VNXe_NFSShareSettingLeaf	356
Table 453 - Referenced properties/methods for EMC_VNXe_CIFSServer_CIFSShare_HostedShareAssocLeaf	356
Table 454 - Referenced properties/methods for EMC_VNXe_NFSServer_NFSShare_HostedShareAssocLeaf	357
Table 455 - Referenced properties/methods for EMC_VNXe_CIFSShare_CIFSShareSetting_ElementSettingDataAssocLeaf.....	357
Table 456 - Referenced properties/methods for EMC_VNXe_NFSShare_NFSShareSetting_ElementSettingDataAssocLeaf.....	357
Table 457 - Referenced properties/methods for EMC_VNXe_CIFSProtocolEndpoint_CIFSShare_SAPAvailableForElementAssocLeaf	357
Table 458 - Referenced properties/methods for EMC_VNXe_NFSProtocolEndpoint_NFSShare_SAPAvailableForElementAssocLeaf	357
Table 459 - Referenced properties/methods for EMC_VNXe_UxfsLocalFileSystem_CIFSShare_SharedElementAssocLeaf	358
Table 460 - Referenced properties/methods for EMC_VNXe_UxfsLocalFileSystem_NFSShare_SharedElementAssocLeaf	358

Table 461 - Signature and Parameters of EMC_VNXe_NFSShareCapabilitiesLeaf.CreateGoalSettings	360
Table 462 - Possible return code of EMC_VNXe_NFSShareCapabilitiesLeaf.CreateGoalSettings	360
Table 463 - Signature and Parameters of EMC_VNXe_CIFSShareCapabilitiesLeaf.CreateGoalSettings	361
Table 464 - Possible return code of EMC_VNXe_CIFSShareCapabilitiesLeaf.CreateGoalSettings	361
Table 465 - Signature and Parameters of EMC_VNXe_FileExportServiceLeaf.SNIA_CreateExportedShare	362
Table 466 - Possible return code of EMC_VNXe_FileExportServiceLeaf.SNIA_CreateExportedShare..	363
Table 467 - Signature and Parameters of EMC_VNXe_FileExportServiceLeaf.SNIA_ModifyExportedShare	364
Table 468 - Possible return code of EMC_VNXe_FileExportServiceLeaf.SNIA_ModifyExportedShare..	365
Table 469 - Signature and Parameters of EMC_VNXe_FileExportServiceLeaf.ReleaseExportedShare	365
Table 470 - Possible return code of EMC_VNXe_FileExportServiceLeaf.ReleaseExportedShare	366
Table 471 - CIM Elements for File Export Manipulation Subprofile	369
Table 472 - Referenced properties/methods for EMC_VNXe_FileExportServiceLeaf	370
Table 473 - Referenced properties/methods for EMC_VNXe_FileExportCapabilitiesLeaf	371
Table 474 - Referenced properties/methods for EMC_VNXe_CIFSShareCapabilitiesLeaf	371
Table 475 - Referenced properties/methods for EMC_VNXe_NFSShareCapabilitiesLeaf	371
Table 476 - Referenced properties/methods for EMC_VNXe_CIFSServer_FileExportService_HostedServiceAssocLeaf	372
Table 477 - Referenced properties/methods for EMC_VNXe_NFSServer_FileExportService_HostedServiceAssocLeaf	372
Table 478 - Referenced properties/methods for EMC_VNXe_FileExportService_FileExportCapabilities_ElementCapabilitiesAssocLeaf	372
Table 479 - Referenced properties/methods for EMC_VNXe_FileExportService_CIFSShareCapabilities_ElementCapabilitiesAssocLeaf	372
Table 480 - Referenced properties/methods for EMC_VNXe_FileExportService_NFSShareCapabilities_ElementCapabilitiesAssocLeaf	372
Table 481 - Referenced properties/methods for EMC_VNXe_FileExportService_CIFSShare_ServiceAffectsElementAssocLeaf	373
Table 482 - Referenced properties/methods for EMC_VNXe_FileExportService_NFSShare_ServiceAffectsElementAssocLeaf	373
Table 483 - Referenced properties/methods for EMC_VNXe_CIFSShareCapabilities_CIFSShareSetting_SettingsDefineCapabilitiesAssocLeaf	373
Table 484 - Referenced properties/methods for EMC_VNXe_NFSShareCapabilities_NFSShareSetting_SettingsDefineCapabilitiesAssocLeaf	373
Table 485 - Signature and parameters of EMC_VNXe_AccountManagementServiceLeaf.CreateUserContact.....	376
Table 486 - Possible return code of EMC_VNXe_AccountManagementServiceLeaf.CreateUserContact	376

Table 487 - Signature and parameters of EMC_VNXe_AccountManagementServiceLeaf.GetUserContact	377
Table 488 - Possible return code of EMC_VNXe_AccountManagementServiceLeaf.GetUserContact ...	377
Table 489 - Signature and parameters of EMC_VNXe_FileExportServiceLeaf.AssignPrivilegeOnExportedShare	379
Table 490 - Possible return code of EMC_VNXe_FileExportServiceLeaf.AssignPrivilegeOnExportedShare	379
Table 491 - Description of properties in the AssociatedPrivilege	381
Table 492 - Mapping between Activities and Privileges	381
Table 493 - CIM Elements implemented in VNXe for Simple Identity Management Profile	384
Table 494 - Referenced properties/methods for EMC_VNXe_AccountManagementServiceLeaf	385
Table 495 - Referenced properties/methods for EMC_VNXe_UserContactLeaf	385
Table 496 - Referenced properties/methods for EMC_VNXe_IdentityLeaf	385
Table 497 - Referenced properties/methods for EMC_VNXe_StorageSystem_AccountManagementService_HostedServiceAssocLeaf	386
Table 498 - Referenced properties/methods for EMC_VNXe_AccountManagementService_Identity_ServiceAffectsElementAssocLeaf	386
Table 499 - Referenced properties/methods for EMC_VNXe_UserContact_Identity_AssignedIdentityAssocLeaf	386
Table 500 - Referenced properties/methods for EMC_VNXe_CIFSShare_Identity_AssociatedPrivilegeAssocLeaf	386
Table 501 - Array Command for Pool Information Display	387

Introduction

Audience

This document provides users with information about the SMI-S interface in the VNXe storage system and how to use it to manage the VNXe.

The proper audience includes, but is not limited to VNXe file storage system client developers.

VNXe basic requirements

Users of this document should be familiar with CIM, WBEM, CIM/XML, SNIA and SMI-S.

Format of Profile Chapter

Each profile chapter contains the following sections:

- **Overview** - A simple overview and introduction about the captioned profile.
- **Class diagram** - UML diagrams based on implementation level.
- **Profile methods** – Extrinsic and Intrinsic methods supported by the profile.
- **Client considerations** - Vital information, key design principles, system characteristics, programming tips, and use cases.

Flowcharts and/or working flow descriptions are included in each use case to demonstrate the usage of functions specified in the profile to perform information retrieval and system management.

- **CIM Element** - Description of implemented classes and associations.


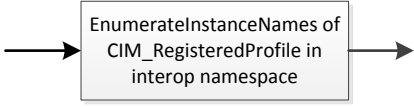
Definitions and conventions

The prefix CIM_ of class name is omitted if there is no confusion.

The following terminology and operations are frequently used in the flowcharts and descriptions of use cases.

- **Enumerate**

Invoke the intrinsic method [EnumerateInstanceNames](#) or [EnumerateInstances](#) to get all Instance-Names (model path) or Instances of a CIM Class in the target namespace.

Flowchart element	Process description
	EnumerateInstances of <Class-Name> in <Namespace-Name> namespace
	EnumerateInstanceNames of <Class-Name> in <Namespace-Name> namespace

- **Traverse**

Invoke the intrinsic method *AssociatorNames* or *Associators* to get all the Instance-Names or Instances associated to a particular source Instance via a given Association Class.

Flowchart element	Process description
	<p>Traverse <Association-Class-Name> to <Result-Class-Name>(<Result-Role>)</p> <p>Example: Traverse <i>CIM_NamespaceInManager</i> (Association Class) from an instance of <i>CIM_Service</i> (Source Instance represents <i>ObjectManager</i>) to <i>CIM_Namespace</i> (Result Class) to get all namespaces managed by CIM ObjectManager.</p>

- **Reference**

Invoke the intrinsic method *ReferenceNames* or *References* to get all the Instance-Names or Instances of an Association Class that refers to a particular source Instance.

Flowchart element	Process description
	<p>Reference <Association-Class-Name> (<Role>)</p> <p>Example: Reference <i>CIM_NamespaceInManager</i> (Association Class) from an instance of <i>CIM_Service</i> (Source Instance that represents <i>ObjectManager</i>) to get all instances of <i>CIM_NamespaceInManager</i> that refer to the ObjectManager.</p>

- **Invoke method**

Invoke an intrinsic method or an extrinsic method defined in a CIM Class.

Flowchart element	Process description
	<p>Invoke <Intrinsic-Method> on <Class-Name></p>
	<p>Invoke <Class-Name>.<Method-Name></p>

Recommendations

For optimum performance:

- Invoke *EnumerateInstanceNames*, *AssociatorNames*, and *ReferenceNames* instead of *EnumerateInstances*, *Associators* and *References* if non-key properties are not needed.
- Invoke *Associatorss* or *AssociatorNames* to retrieve instances of result object class instead of invoking *EnumerateInstances* or *EnumerateInstanceNames* on the resulting object class directly.
- Invoke *References* or *ReferenceNames* to retrieve instances of association class instead of invoking *EnumerateInstances* or *EnumerateInstanceNames* on the association class directly.

References

1. Distributed Management Task Force (DMTF): <http://www.dmtf.org>
2. Common Information Model (CIM): <http://www.dmtf.org/standards/cim/>
3. Web Based Enterprise Management (WBEM) Specification: <http://www.dmtf.org/standards/wbem/>
4. Storage Management Initiative (SMI): <http://www.snia.org/smi/home>
5. SMI Specification: http://www.snia.org/tech_activities/standards/curr_standards/smi
6. Service Location Protocol: <http://www.ietf.org/rfc/rfc2165.txt>
7. SNIA Conformance Testing Program (CTP): <http://www.snia.org/ctp/>
8. Simple Identity Management Profile (DSP1034):
http://www.dmtf.org/sites/default/files/standards/documents/DSP1034_1.0.1.pdf

Server Profile

Overview

A CIM Server supports CIM-XML or other WBEM protocols. 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 (CIMOM) based on the communication mechanisms that it supports. The namespace model of the Server Profile describes the namespaces managed by the object manager and the type information contained within the namespace. The main information provided in the namespace part of the model is the namespace itself, and its association to the ObjectManager. The InteropNamespace refers to the first namespace found in the InteropSchemaNamespace attribute of the SLP Template.

A Server is modeled as a System with a HostedService association to an ObjectManager. The ObjectManager is subclassified from Service. Implementations support an ElementConformsToProfile association that references the RegisteredProfile for the Server Profile and the ObjectManager (rather than CIM_System which is common in the other profiles).

It is mandatory that all namespaces supported by the Server be identified (as the Namespace class) and associated to the ObjectManager via the NamespaceInManager association.

The communication protocol is CIM-XML. This class is associated to the ObjectManager via the CommMechanismForManager association.

The Profile Registration Profile describes the set of classes, associations, and profiles supported by the ObjectManager. The Profile Registration Profile is required by the Server Profile.

NOTE: For more details, refer to [Clause 40: Server Profile in Storage Management Technical Specification, Part 2 Common Profiles, Version 1.6.0, Revision 4](#).

Class diagram

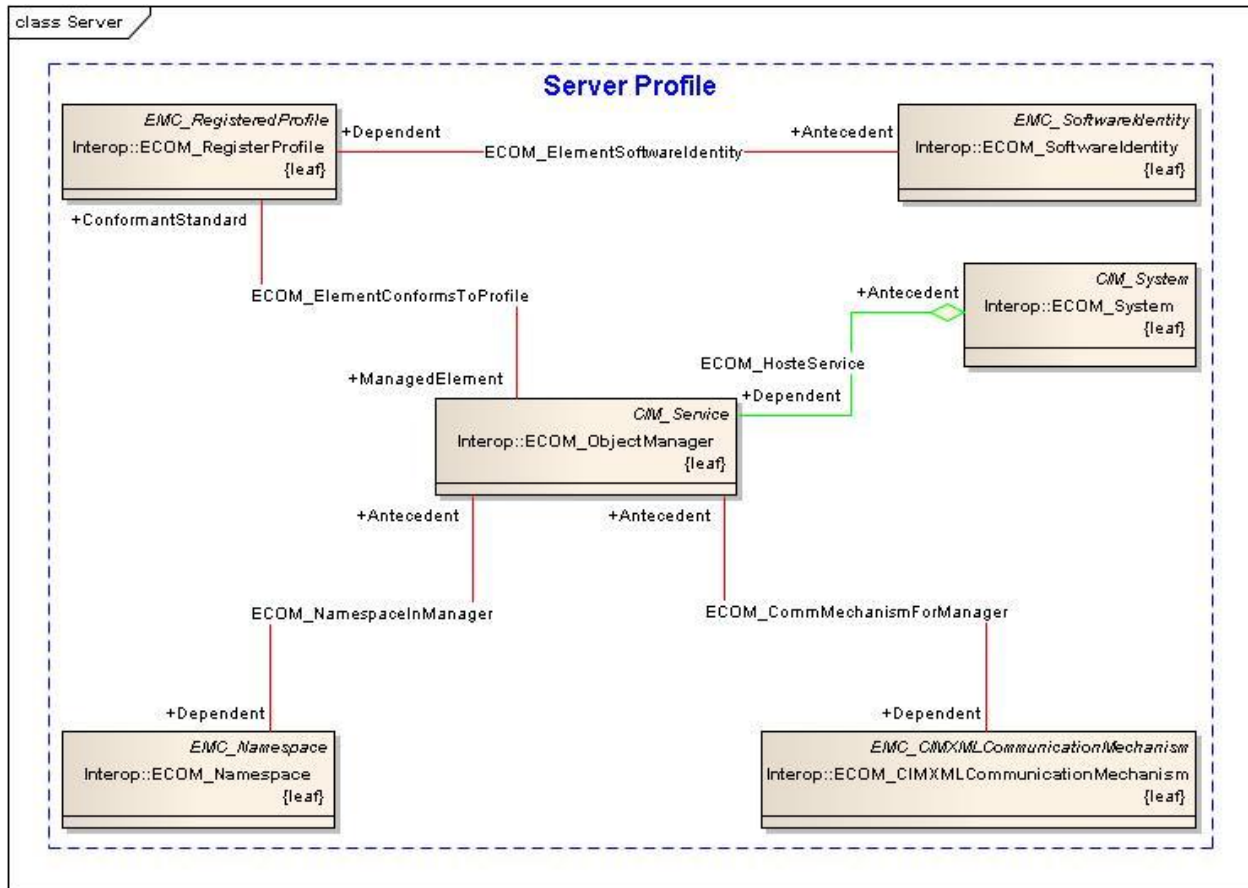


Figure 1 - Server Profile class diagram

Method of the Profile

This profile does not include any extrinsic methods. Use this profile to discover information about the CIM Server, but you cannot change the CIM Server settings and properties.

Client considerations

Model specifications

- Server Profile Version
Implemented in *ECOM_RegisteredProfile(Server).RegisteredVersion*.
- ECOM Version
Implemented in *ECOM_SoftwareIdentity.VersionString* associated with *ECOM_RegisteredProfile(Server)* via *ECOM_ElementSoftwareIdentity*.
- CIM Server Hostname
Implemented in *ECOM_System.Name*.
- Managed Namespace

Implemented in *ECOM_Namespace.Name* associated with *ECOM_ObjectManager* via *ECOM_NamespaceInManager*.

- Communication Mechanism
 VNXe SMI-S Provider supports the CIM-XML protocol only.

Use case: View Server Profile implementation for EMC CIM server

This use case describes how to view the Server Profile implementation for an EMC CIM Server, including discovering ECOM (EMC CIM Object Manager) and getting *SoftwareIdentity*, *CIMServer*, *ManagedNamespaces*, and *CommunicationProtocol*.

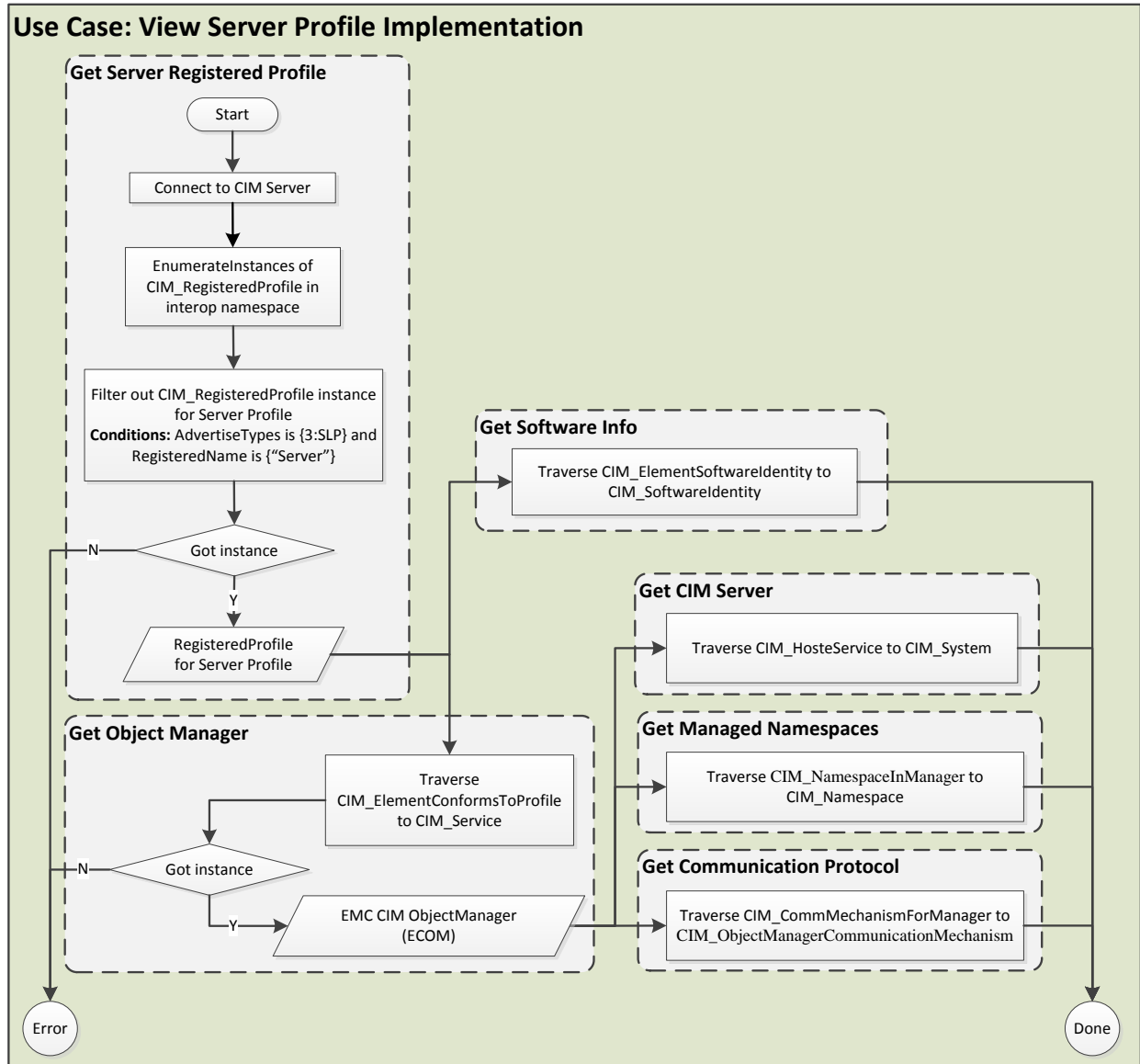


Figure 2 - Flowchart of viewing Server Profile implementation

CIM Element

The implemented classes and associations related to Server Profile on the VNXe storage system are described as follows:

Table 1 - CIM Elements for Server Profile

CIM Class	Implemented Class	Description
CIM_CIMXMLCommunicationMechanism	ECOM_CIMXMLCommunicationMechanism	CIMXML communication protocols supported by ECOM.
CIM_CommMechanismForManager	ECOM_CommMechanismForManager	Associates the communication protocols to the ObjectManager.
CIM_ElementConformsToProfile	ECOM_ElementConformsToProfile	Associates the ObjectManager to the registered profile for Server Profile.
CIM_HostedService	ECOM_HostedService	Associates the ObjectManager to the EMC CIM Server that is hosting the ObjectManager.
CIM_Namespace	ECOM_Namespace	Namespaces managed by ECOM.
CIM_NamespaceInManager	ECOM_NamespaceInManager	Associates the Namespaces to the ObjectManager which manages the Namespaces.
CIM_ObjectManager	ECOM_ObjectManager	EMC CIM Object Manager (ECOM).
CIM_ObjectManagerCommunicationMechanism	ECOM_OtherCommunicationMechanism	Other communication protocols supported by ECOM.
CIM_RegisteredProfile	ECOM_RegisteredProfile (Server)	Registered profile for Server Profile.
CIM_System	ECOM_System	EMC CIM Server.

ECOM_ObjectManager

Table 2 - Referenced properties/methods for ECOM_ObjectManager

CIM property	Description/notes
SystemCreationClassName	CreationClassName of the scoping system.
SystemName	The name of the scoping system.
CreationClassName	The name of the class used to create this instance.
Name	A string that uniquely identifies the ObjectManager.
Caption	The short textual description of the object.
Description	The description of the CIM Server for human interface.

CIM property	Description/notes
ElementName	The name of the CIM Server.
EnabledDefault	Default or startup configuration for the EnabledState of the ObjectManager.
EnabledState	Enabled and disabled states of the ObjectManager.
GatherStatisticalData	Controls the gathering of statistical data made accessible through the CIM_CIMOMStatisticalData objects.
OperationalStatus	Current operational status of the ObjectManager.
OtherEnabledState	Describes the enabled or disabled state of the element when the EnabledState property is set to {1: Other}.
PrimaryOwnerContact	Contact information of the primary owner of this computer system.
PrimaryOwnerName	The name of the primary owner of this computer system.
RequestedState	The last requested or desired state for the ObjectManager.
Started	Whether the Service has been started (TRUE) or stopped (FALSE).
StartMode	Start mode.
StatusDescriptions	Strings describing the various OperationalStatus array value.

ECOM_RegisteredProfile

Table 3 - Referenced properties/methods for ECOM_RegisteredProfile

CIM property	Description/notes
InstanceID	The unique identifier for the RegisteredProfile.
AdvertiseTypeDescription	A free-form string providing additional information related to the AdvertiseType.
AdvertiseType	Advertisement for the profile information.
Caption	The short textual description of the object.
Description	The textual description of the object.
ElementName	A use-friendly name of the object.
OtherRegisteredOrganization	A free-form string providing additional description of the organization when {1: Other} is specified for the RegisteredOrganization.
RegisterName	The name of the registered profile.
RegisteredOrganization	The organization that defines the profile.

CIM property	Description/notes
RegisteredVersion	Version of the profile.

ECOM_System

Table 4 - Referenced properties/methods for ECOM_System

CIM property	Description/notes
CreationClassName	The name of the class used to create this instance.
Name	A unique identifier for the CIM Server.
Caption	The short textual description of the object.
Description	The description of the CIM Server for human interface.
ElementName	The name of the CIM Server.
EnabledDefault	Default or startup configuration for the EnabledState of the CIM Server.
EnabledState	Enabled and disabled states of the CIM Server.
NameFormat	The format of the Name property.
OperationalStatus	Current operational status of the ObjectManager.
OtherEnabledState	A string that describes the enabled or disabled state of the element when the EnabledState property is set to {1: Other}.
PrimaryOwnerContact	Contact information of the primary owner of this computer system.
PrimaryOwnerName	The name of the primary owner of this computer system.
RequestedState	The last requested or desired state for the ObjectManager.
Roles	An array of string that specifies the administrator-defined roles the CIM Server plays in the managed environment.
StatusDescriptions	Strings describing the various OperationalStatus array value.

ECOM_Namespace

Table 5 - Referenced properties/methods for ECOM_Namespace

CIM property	Description/notes
SystemCreationClassName	CreationClassName of the scoping system.
SystemName	The name of the scoping system.
ObjectManagerCreationClassName	CreationClassName of the scoping ObjectManager.
ObjectMananerName	The name of the scoping ObjectManager.

CIM property	Description/notes
CreationClassName	The name of the class used to create this instance.
Name	A string that uniquely identifies the Namespace within the ObjectManager.
Caption	The short textual description of the object.
ClassType	Schema of the objects in the Namespace.
ClassTypeVersion	Version of the objects in the Namespace.
Description	The textual description of the object.
DescriptionOfClassType	Detail for the object hierarchy of the namespace.
ElementName	A user-friendly name.

ECOM_CIMXMLCommunicationMechanism

Table 6 - Referenced properties/methods for ECOM_CIMXMLCommunicationMechanism

CIM property	Description/notes
SystemCreationClassName	CreationClassName of the scoping system.
SystemName	The name of the scoping system.
CreationClassName	The name of the class used to create this instance.
Name	A string that uniquely identifies the access point.
AdvertiseTypeDescription	A free-form string providing additional information related to the AdvertiseType.
AdvertiseType	Advertisement for the access point.
AuthenticationMechanismDescriptions	Description of the supported mechanism.
AuthenticationMechanismsSupported	Types of authentication mechanism supported by the ObjectManager.
CIMValidated	Whether the CIM Server is strictly validating or not.
CIMXMLProtocolVersion	CIM-XML protocol version supported by the ObjectManager.
CommunicationMechanism	The only valid mechanism for this subclass is CIM-XML.
Caption	The short textual description of the object.
Description	The description of the CIM Server for human interface.
ElementName	The name of the CIM Server.
EnabledDefault	Default or startup configuration for the EnabledState of the

CIM property	Description/notes
	ObjectManager.
EnabledState	Enabled and disabled states of the ObjectManager.
FunctionalProfileDescriptions	Descriptions of the operations supported by the ObjectManager using this protocol.
FunctionalProfilesSupport	Types of operations supported by the ObjectManager using this protocol.
MultipleOperationsSupported	Whether the ObjectManager supports multiple operation requests.
OperationalStatus	Current operational status of the ObjectManager.
OtherCommunicationMechanismDescription	A free-form string providing additional description of the mechanism when {1: Other} is specified for the CommunicationMechanism.
OtherEnabledState	A string that describes the enabled or disabled state of the element when the EnabledState property is set to {1: Other}.
RequestedState	The last requested or desired state for the ObjectManager.
StatusDescriptions	Strings describing the various OperationalStatus array value.
Version	The CIM-XML version supported by the ObjectManager.

ECOM_ElementConformsToProfile

Table 7 - Referenced properties/methods for ECOM_ElementConformsToProfile

CIM property	Description/notes
ConformantStandard	Reference of ECOM_RegisteredProfile
ManagedElement	Reference of CIM_ManagedElement

ECOM_HostedService

Table 8 - Referenced properties/methods for ECOM_HostedService

CIM property	Description/notes
Antecedent	Reference of ECOM_System
Dependent	Reference of CIM_Service

ECOM_CommMechanismForManager

Table 9 - Referenced properties/methods for ECOM_CommMechanismForManager

CIM property	Description/notes
Antecedent	Reference of ECOM_ObjectManager

CIM property	Description/notes
Dependent	Reference of CIM_ObjectManagerCommunicationMechanism

ECOM_NamespaceInManager

Table 10 - Referenced properties/methods for ECOM_NamespaceInManager

CIM property	Description/notes
Antecedent	Reference of ECOM_ObjectManager
Dependent	Reference of ECOM_Namespace

Profile Registration Profile

Overview

The Profile Registration Profile models the profiles registered in the object manager and the associations between registration classes and the domain classes that implement the profile.

In DMTF profiles, the term 'component profile' is similar to 'subprofile' in SMI-S 1.0.x and 1.1.x; and the term 'autonomous profile' is similar to 'profile' in SMI-S 1.0.x and 1.1.x. SNIA implementations may use the SNIA 1.0.x/1.1.x approach with the RegisteredSubProfile and SubProfileRequiresProfile subclasses, or the DMTF approach with RegisteredProfile for component profiles and ReferencedProfile.

SMI-S clients should use the superclasses (RegisteredProfile and ReferencedProfile) in CIM operations to assure that implementations conforming to either SMI-S or DMTF profiles are discovered. ReferencedProfile associates with two instances of RegisteredProfile. The DMTF Profile Registration Profile describes how to use the Antecedent and Dependent references when one profile includes another in its supported/referenced profile list.

Instances of RegisteredProfile, RegisteredSubProfile, SubProfileRequiresProfile, and ReferencedProfile are in the Interop namespace. The ManagedElement is in the implementation namespace. RegisteredProfile instances are required by all SMI-S profiles, including those instances named as Subprofiles or Packages.

NOTE: For more details, refer to [Clause 41: Profile Registration Profile in Storage Management Technical Specification, Part 2 Common Profiles, Version 1.5.0, Revision 6](#).

Class diagram

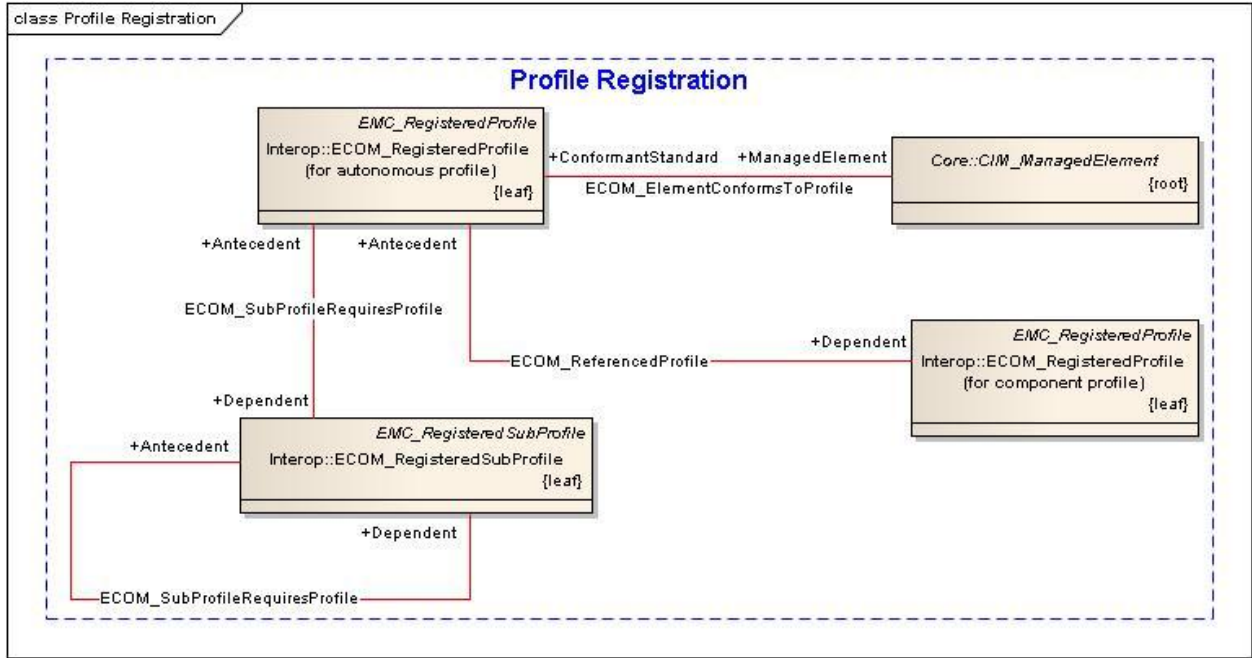


Figure 3 - Profile Registration Profile class diagram

Method of the Profile

This profile does not include any extrinsic methods. Use this profile to discover profile registration information in the Object Manager, but you cannot change profile registration information.

Client consideration

Model specification

Version of SMI-S Profile

Each *RegisteredProfile* instance that represents a profile from SMI-S version 1.2.0 or later (represented by *ECOM_RegisteredSubProfile* or *ECOM_RegisteredProfile*) is associated to a *RegisteredProfile* instance that holds the SMI-S version number (represented by *ECOM_RegisteredProfile*). The version number (*RegisteredVersion*) of SMI-S profiles may or may not be the same as the version number of the SMI-S Registered Profile.

Note: For further information about how the client determines the version of an SMI-S profile, refer to [Use Case: Determine the SNIA Version of a SMI-S Profile](#).

Use case: Find all Profiles on a CIM Server

This use case describes how to find all profiles on a CIM server.

Invoking *EnumerateInstances* on *CIM_RegisteredProfile*, to get all profiles registered on a specified CIM Server.

Use case: Identify the ManagedElement defined by a Profile

This use case describes how to identify ManagedElement defined by a profile.

From *CIM_RegisteredProfile*, traverse *CIM_ElementConformsToProfile* to *CIM_ManagedElement* to get all managed elements defined by the profile.

Use case: Enumerate autonomous profiles supported by a given CIM server

This use case describes how to enumerate autonomous profiles supported by the CIM server.

A *CIM_RegisteredProfile* is an autonomous profile if the Profile is not referenced by any other profiles.

Taking the current *CIM_RegisteredProfile* as a *Dependent*, traverse *CIM_ReferencedProfile* to *CIM_RegisteredProfile* (role is *Antecedent*). If there is no result, the current profile is autonomous.

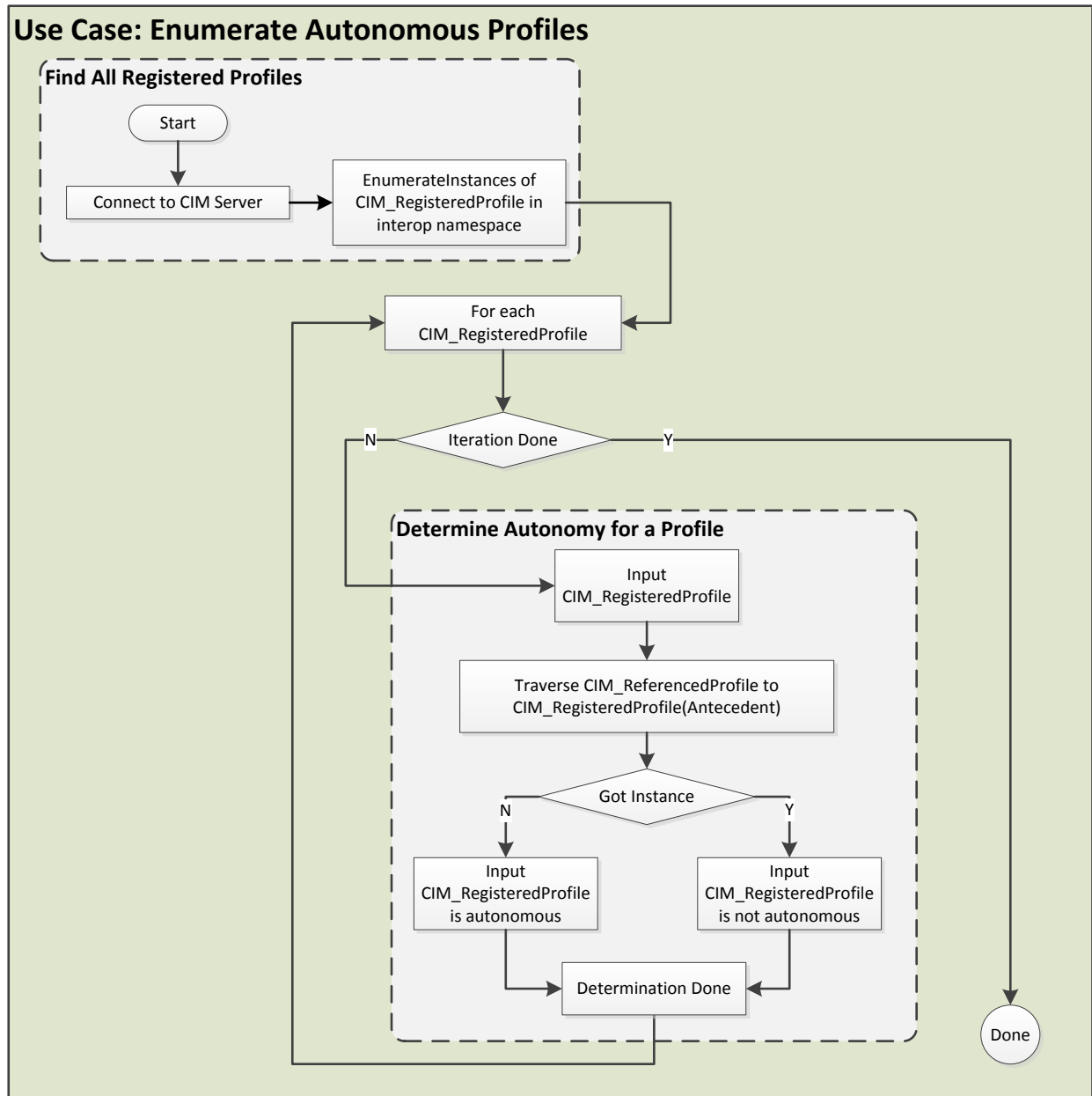


Figure 4 - Flowchart of enumerating autonomous profiles

Use case: Determine the SNIA Version of a SMI-S Profile

This use case describes how to get the SNIA version of an SMI-S profile.

If the current profile is not associated with another *CIM_RegisteredProfile* (Role is *ConformantStandard*) via *CIM_ElementConformsToProfile*, use the *RegisteredVersion* of current profile as the final version; otherwise, use the *RegisteredVersion* of the resulting *CIM_RegisteredProfile* (*ConformantStandard*) as the final version.

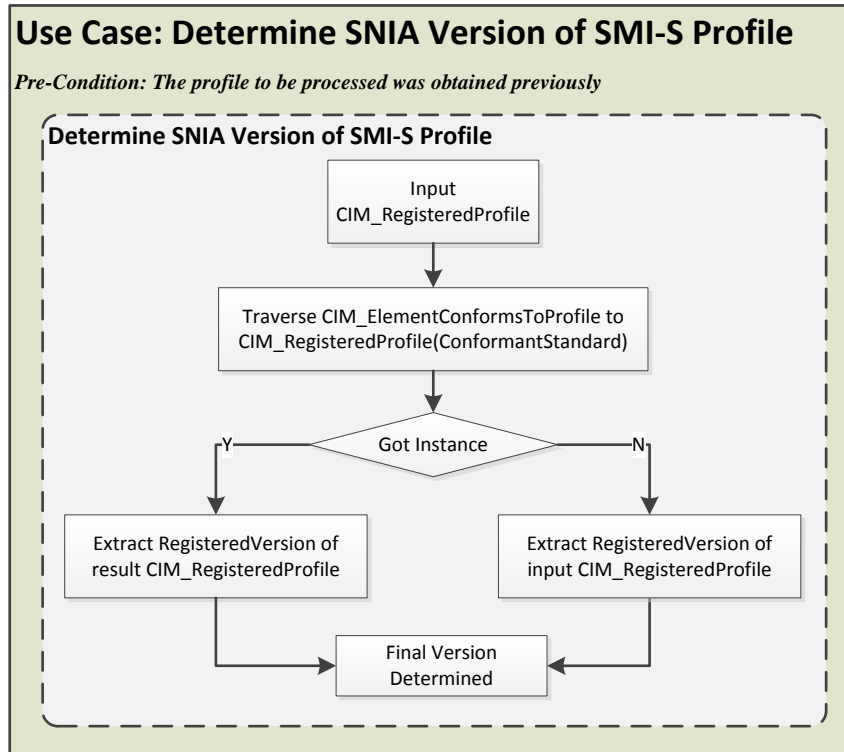


Figure 5 - Flowchart of determining the SNIA version of an SMI-S Profile

CIM Elements

The implemented classes and associations related to Profile Registration Profile on the VNXe storage system are described as follows:

Table 11 - CIM elements for Profile Registration profile

CIM Class	Implemented Class	Description
CIM_ElementConformsToProfile	ECOM_ElementConformsToProfile	Associates the RegisteredProfile for SMI-S autonomous profiles to scoping managed elements.
CIM_ReferencedProfile	ECOM_ReferencedProfile	Associates to a profile and its supported profiles.
CIM_RegisteredProfile	ECOM_RegisteredProfile	Registered profiles in ECOM including autonomous profile and SMI-S Registered Profile.

CIM Class	Implemented Class	Description
CIM_RegisteredSubProfile	ECOM_RegisteredSubProfile	SMI-S Subprofile from version 1.2.0 or later.
CIM_SubProfileRequiresProfile	ECOM_SubProfileRequiresProfile	Specialization of ReferencedProfile referencing a SubProfile.

ECOM_RegisteredProfile

For further information, refer to [ECOM_RegisteredProfile](#) in Server Profile.

ECOM_RegisteredSubProfile

Table 12 - Referenced properties/methods for ECOM_RegisteredSubProfile

CIM property	Description/notes
InstanceID	The unique identifier for the RegisteredProfile.
AdvertiseTypeDescription	A free-form string providing additional information related to the AdvertiseType.
AdvertiseType	Advertisement for the profile information.
Caption	The short textual description of the object.
Description	The textual description of the object.
ElementName	A user-friendly name for the object.
OtherRegisteredOrganization	A free-form string providing additional description of the organization when {1: Other} is specified for the RegisteredOrganization.
RegisterName	The name of the registered profile.
RegisteredOrganization	Organization that defines the profile.
RegisteredVersion	Version of the profile.

ECOM_ReferencedProfile

Table 13 - Referenced properties/methods for ECOM_ReferencedProfile

CIM property	Description/notes
Antecedent	Reference of ECOM_RegisteredProfile
Dependent	Reference of ECOM_RegisteredProfile

ECOM_SubProfileRequiresProfile

Table 14 - Referenced properties/methods for ECOM_SubProfileRequiresProfile

CIM property	Description/notes
--------------	-------------------

CIM property	Description/notes
Antecedent	Reference of ECOM_RegisteredProfile
Dependent	Reference of ECOM_RegisteredSubProfile

[ECOM_ElementConformsToProfile](#)

For further information, refer to [ECOM_ElementConformsToProfile](#) in Server Profile

Multiple Computer System Subprofile

Overview

The Multiple Computer System Subprofile allows a client to retrieve information about the redundancy configuration and the capabilities of the VNXe system.

This subprofile models multiple systems that cooperate to present a “virtual” computer system with additional capabilities or redundancy. This virtual aggregate system is sometimes referred to as a cluster.

NOTE: For more details, refer to [Clause 30: Multiple Computer System Subprofile in Storage Management Technical Specification, Part 2 Common Profiles, Version 1.6.0, Revision 4.](#)

Class diagram

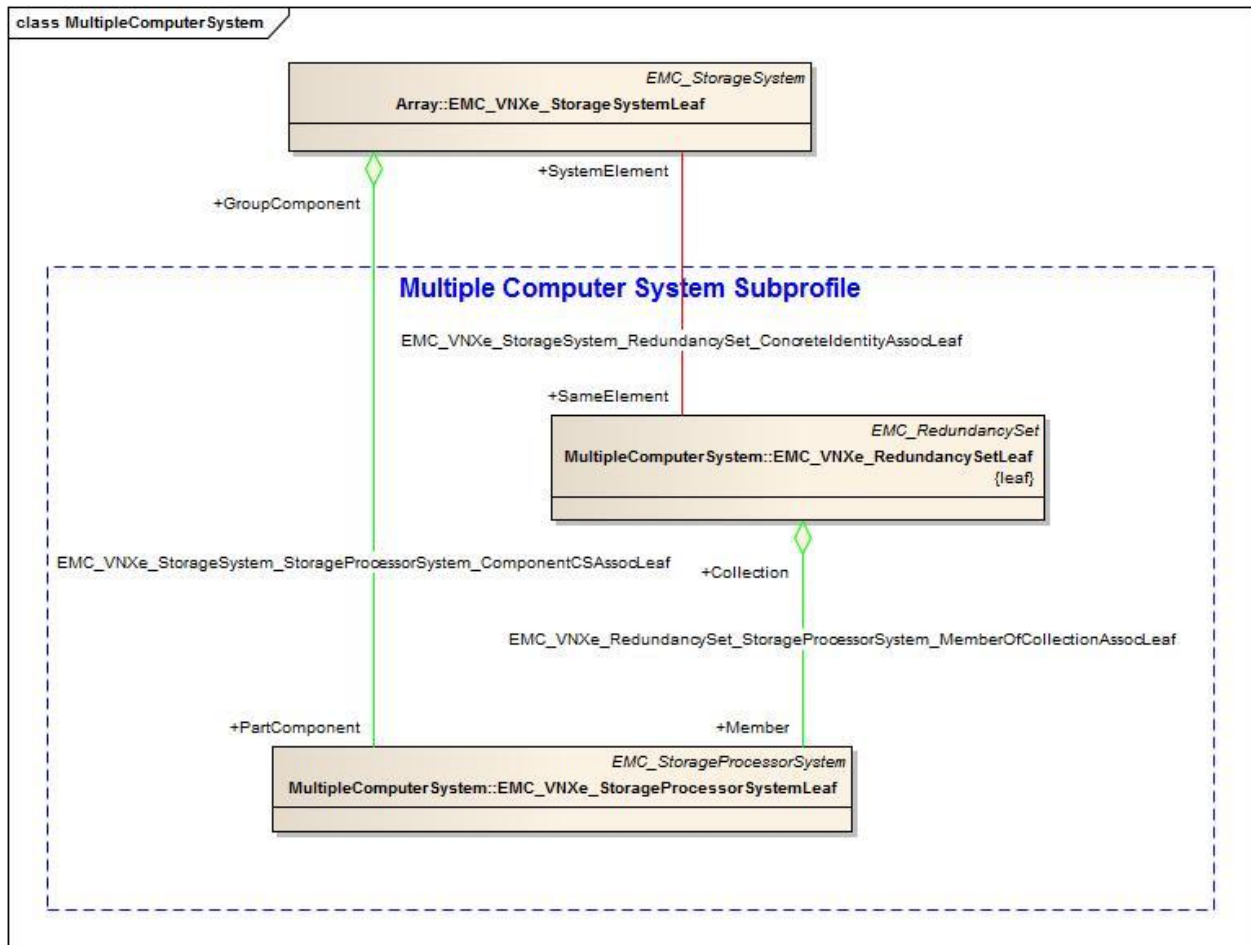


Figure 6 - Multiple Computer System Subprofile class diagram

Methods of the Profile

This subprofile does not include any extrinsic methods. Use this subprofile to discover information about the topology of computer systems, but you cannot change the topology.

Client considerations

Model Specification

- Type of RedundancySet
Implemented in *RedundancySet.TypeOfSet*. For VNXe, the value of *TypeOfSet* is {4: Sparing}.
- Redundancy Status
Implemented in *RedundancySet.RedundancyStatus*. Value is calculated from HealthState of both SPs (represented by *EMC_VNXe_StorageProcessorSystemLeaf*).

Table 15 - Value map of RedundancyStatus

SP A HealthState	SP B HealthState	Redundancystatus
Exist	Not exist	Redundancy_Lost
Unknown	OK OK_BUT Unknown Minor_failure Major_failure Critical_failure Non_recoverable_failure	Unknown
Unknown	Degraded_Warning	Redundancy_Lost
OK OK_BUT	Unknown	Unknown
OK OK_BUT	OK OK_BUT	Fully_Redundant
OK OK_BUT	Degraded_Warning Minor_failure Major_failure Critical_failure Non_recoverable_failure	Redundancy_Lost
Degraded_Warning	OK OK_BUT Unknown Degraded_Warning Minor_failure	Redundancy_Lost

SP A HealthState	SP B HealthState	Redundancystatus
	Major_failure Critical_failure Non_recoverable_failure	
Minor_failure Major_failure Critical_failure Non_recoverable_failure	Unknown	Unknown
Minor_failure Major_failure Critical_failure Non_recoverable_failure	OK OK_BUT Degraded_Warning	Redundancy_Lost
Minor_failure Major_failure Critical_failure Non_recoverable_failure	Minor_failure Major_failure Critical_failure Non_recoverable_failure	Overall_Failure

Use case: Find top-level ComputerSystem

This use case describes how to find the top-level ComputerSystem.

In VNXe, the Top-level ComputerSystem is both the NAS Head ComputerSystem and the Array ComputerSystem. For further information, refer to [Use Case: Discover Block Server \(Array ComputerSystem\)](#) in Array Profile.

Use case: Check system redundancy

This use case describes how to check VNXe system redundancy.

1. From Top-level ComputerSystem, traverse *CIM_ConcreteIdentity* to *CIM_RedundancySet* to view redundancy status.
2. From the resulting instance of *CIM_RedundancySet*, traverse *CIM_MemberOfCollection* to *CIM_ComputerSystem* to get systems contributing to the redundancy set.

Use case: Find top-level ComputerSystem for any LogicalDevice

This use case describes how to find the Top-level ComputerSystem from a given LogicalDevice.

1. From *LogicalDevice* (including *LogicalDisk*, *StorageVolume*, *PrimordialStorageExtent*, *EthernetPort* and so on), traverse *CIM_SystemDevice* to *CIM_ComputerSystem* to get the scoping system.
2. Taking the scoping system as *PartComponent*, traverse *CIM_ComponentCS* to *CIM_ComputerSystem* (*GroupComponent*). If there is no result, the scoping system is already the top-level. Otherwise, the result

is the top-level since all component systems must be associated with a top-level system via *CIM_ComponentCS*.

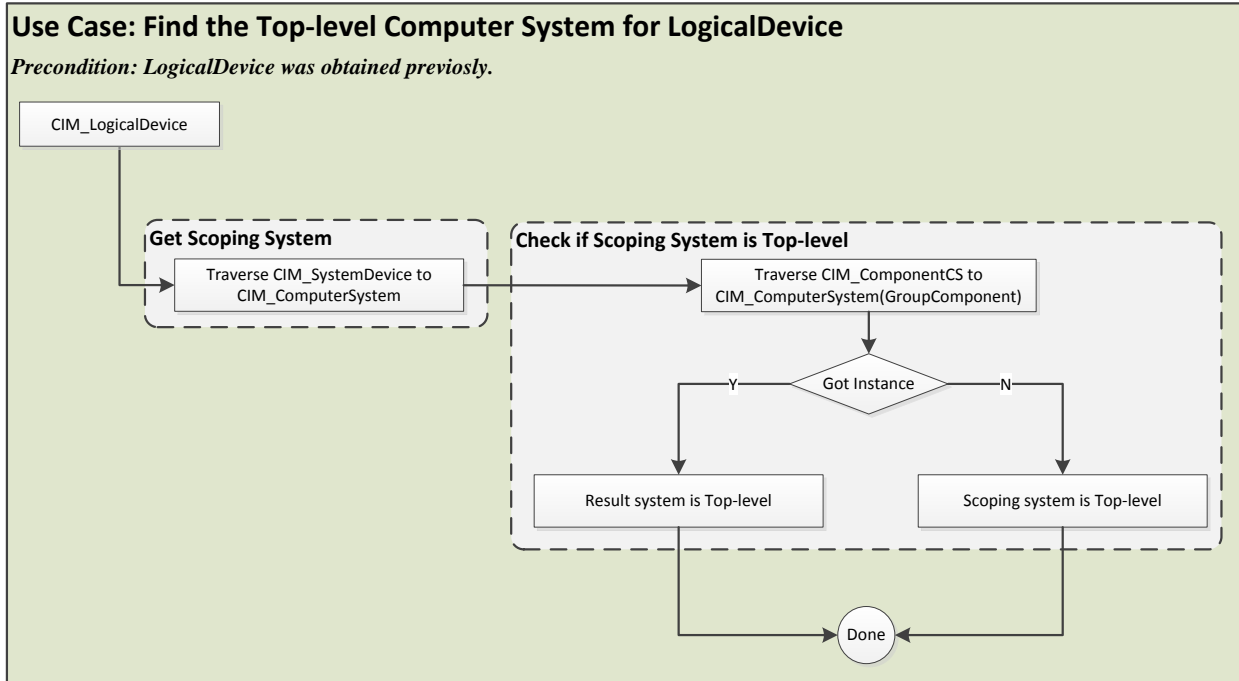


Figure 7 - Flowchart of find top-level ComputerSystem for any LogicalDevice

CIM Elements

The implemented classes and associations related to Multiple Computer System Subprofile in VNXe Storage System are described as follows:

Table 16 - CIM Elements for Multiple Computer System Subprofile

CIM Class	Implemented Class	Description
CIM_ComponentCS	EMC_VNXe_StorageSystem_StorageProcessorSystem_ComponentCSAssocLeaf	Represents the association between the storage system and storage processor.
CIM_ComputerSystem (Component)	EMC_VNXe_StorageProcessorSystemLeaf	Represents the storage processor (SP A/SP B).
CIM_ComputerSystem (System)	EMC_VNXe_StorageSystemLeaf	Represents the top-level ComputerSystem.
CIM_ConcretelIdentity	EMC_VNXe_StorageSystem_RedundancySet_ConcretelIdentityAssocLeaf	Represents the association between the storage system and redundancy set.
CIM_MemberOfCollection	EMC_VNXe_RedundancySet_StorageProcessorSystem_MemberOfCollectionAssocLeaf	Represents the association between the redundancy set and SP.

CIM_RedundancySet	EMC_VNXe_RedundancySetLeaf	Represents the system redundancy set.
-------------------	----------------------------	---------------------------------------

EMC_VNXe_StorageSystemLeaf

Table 17 - Referenced properties/methods for EMC_VNXe_StorageSystemLeaf

CIM property	Description/notes
CreationClassName	Set to EMC_VNXe_StorageSystemLeaf.
Name	Unique identifier for the array. Set to <SystemName>
ElementName	User friendly name.
OtherIdentifyingInfo	Additional data, beyond System Name information, that could be used to identify a ComputerSystem. Not supported.
IdentifyingDescriptions	An array of strings providing explanations and details behind the entries in the OtherIdentifying Info array.
OperationalStatus	Overall status of the array.
NameFormat	Format for Name property.
Dedicated	Indicates that this computer system is dedicated to operation as a storage array. Set to [15: Block Server, 24: NAS Head]
PrimaryStatus	One of the four statuses: Unknown, OK, Degraded, Errorby calculating from the OperationalStatus.
HealthState	Indicates the current health status of the ComputerSystem.
StatusDescriptions	Strings describing the various OperationalStatus array values.
EnabledState	Set to 5: Not Applicable
RequestedState	Set to 12: Not Applicable

EMC_VNXe_StorageProcessorSystemLeaf

Table 18 - Referenced properties/methods for EMC_VNXe_StorageProcessorSystemLeaf

CIM property	Description/notes
CreationClassName	Set to EMC_VNXe_StorageProcessorSystemLeaf.
Name	Unique identifier for the array.
ElementName	User friendly name.
OtherIdentifyingInfo	Additional data, beyond System Name information, that could be used to identify a ComputerSystem.

CIM property	Description/notes
	Not supported.
IdentifyingDescriptions	An array of strings providing explanations and details behind the entries in the OtherIdentifying Info array.
OperationalStatus	Overall status of the array.
NameFormat	Format for Name property.
Dedicated	Indicates that this computer system is dedicated to operation as a storage array. Set to [2: Other]
PrimaryStatus	A high level status value.
StatusDescriptions	Strings describing the various OperationalStatus array values.
TransitioningToState	Set to 12: Not Applicable
EnabledState	Set to 5: Not Applicable
RequestedState	Set to 12: Not Applicable

EMC_VNXe_RedundancySetLeaf

Table 19 - Referenced properties/methods for EMC_VNXe_RedundancySetLeaf

CIM property	Description/notes
InstanceID	Unique identifier.
ElementName	User friendly name.
RedundancyStatus	Provides information on the state of the RedundancyGroup. Unknown, Fully Redundant, Degraded Redundancy, Redundancy Lost, Overall Failure
TypeOfSet	Provides information on the type of redundancy. Set to [4: Sparing].
LoadBalanceAlgorithm	The current load balance algorithm. Set to 2: No Load Balancing.

EMC_VNXe_StorageSystem_StorageProcessorSystem_ComponentCSAssocLeaf

Table 20 - Referenced properties/methods for EMC_VNXe_StorageSystem_StorageProcessorSystem_ComponentCSAssocLeaf

CIM property	Description/notes
GroupComponent	Reference of the Top-Level ComputerSystem.

CIM property	Description/notes
PartComponent	Reference of the contained StorageProcessorSystem.

EMC_VNXe_StorageSystem_RedundancySet_ConcretIdentityAssocLeaf

**Table 21 - Referenced properties/methods for
EMC_VNXe_StorageSystem_RedundancySet_ConcretIdentityAssocLeaf**

CIM property	Description/notes
SameElement	Reference of the EMC_VNXe_RedundancySetLeaf.
SystemElement	Reference of the Top-Level ComputerSystem.

EMC_VNXe_RedundancySet_StorageProcessorSystem_MemberOfCollectionAssocLeaf

**Table 22 - Referenced properties/methods for
EMC_VNXe_RedundancySet_StorageProcessorSystem_MemberOfCollectionAssocLeaf**

CIM property	Description/notes
Collection	Reference of the EMC_VNXe_RedundancySetLeaf.
Member	Reference of the EMC_VNXe_StorageProcessorSystemLeaf.

Software Subprofile

Overview

The Software Profile models software or firmware installed in a computer system. Information about the installed software is provided by the SoftwareIdentity class. This is linked to the system using an InstalledSoftwareIdentity association.

Firmware is modeled as SoftwareIdentity. InstalledSoftwareIdentity is used for firmware associated with a System.

NOTE: For more details, refer to [Clause 38: Software Subprofile in Storage Management Technical Specification, Part 2 Common Profiles, Version 1.6.0, Revision 4](#).

Class diagram

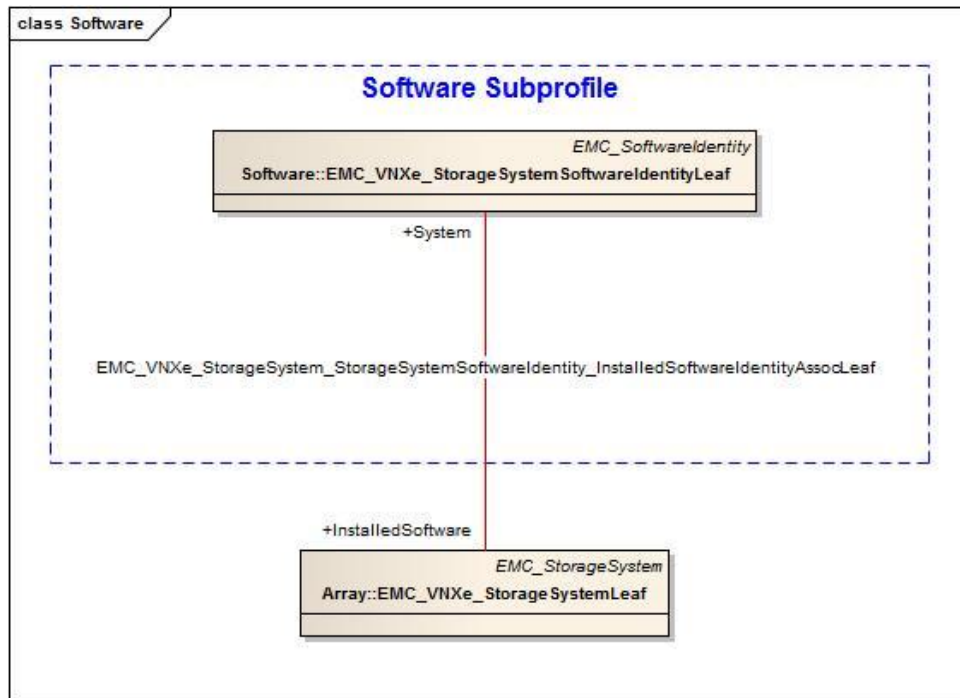


Figure 8 - Software Subprofile class diagram

Method of the Profile

This subprofile does not include any extrinsic methods.

Client considerations

N/A

CIM Element

The implemented classes and associations related to Software Subprofile in VNXe Storage System are described as follows:

Table 23 - CIM Elements for Software Subprofile

CIM Class	Implemented Class	Description
CIM_InstalledSoftwareIdentity	EMC_VNXe_StorageSystem_StorageSystemSoftwareIdentity_InstalledSoftwareIdentityAssocLeaf	Associates the SoftwareIdentity with the corresponding ComputerSystem.
CIM_SoftwareIdentity	EMC_VNXe_StorageSystemSoftwareIdentityLeaf	Software information about VNXe Storage System.

EMC_VNXe_StorageSystemSoftwareIdentityLeaf

Table 24 - Referenced properties/methods for EMC_VNXe_StorageSystemSoftwareIdentityLeaf

CIM property	Description/notes
InstanceID	Unique identifier for this instance.
VersionString	Version of StorageSystem software.
Manufacturer	The information of Manufacturer. Set to EMC Corporation.
MajorVersion	Major version of StorageSystem software.
MinorVersion	Minor version of StorageSystem software.
IsEntity	Set to FALSE
IsLargeBuildNumber	Set to FALSE

EMC_VNXe_StorageSystem_StorageSystemSoftwareIdentity_InstalledSoftwareIdentityAssocLeaf

Table 25 - Referenced properties/methods for EMC_VNXe_StorageSystem_StorageSystemSoftwareIdentity_InstalledSoftwareIdentityAssocLeaf

CIM property	Description/notes
System	Reference of EMC_VNXe_StorageSystemLeaf
InstalledSoftware	Reference of EMC_VNXe_StorageSystemSoftwareIdentityLeaf

Physical Package Package

Overview

The Physical Package Package models information about a storage system's physical package, and optionally about internal subpackages.

A System is 'realized' using a SystemPackaging association to a PhysicalPackage (or a subclass such as Chassis). The physical containment model is built using Container associations and subclasses (such as PackageInChassis).

Physical elements are described as products using the Product class and ProductPhysicalComponent associations. The Product instances are built into a hierarchy using the ProductParentChild association. The Product class holds information such as vendor name, serial number, and version.

NOTE: For more details, refer to [Clause 31: Physical Package Package in Storage Management Technical Specification, Part 2 Common Profiles, Version 1.6.0, Revision 4.](#)

Class diagram

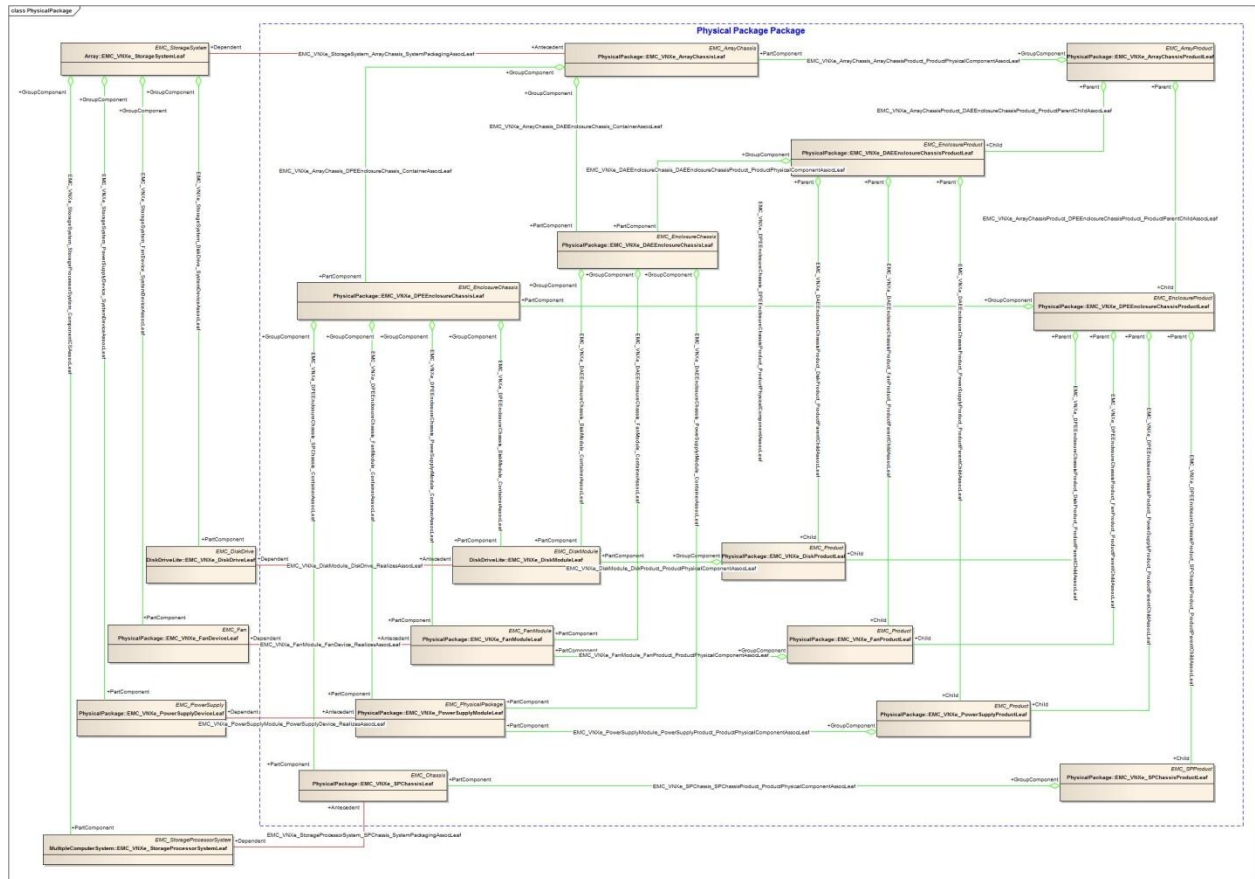


Figure 9 - Physical Package Package class diagram

Methods of the Profile

This subprofile does not include any extrinsic methods.

Client considerations

Model Specification

- VNXe system product information
System product version: *EMC_VNXe_ArrayChassisProductLeaf.Version*.
System serial number: *EMC_VNXe_ArrayChassisProductLeaf.IdentifyingNumber*.
System vendor: *EMC_VNXe_ArrayChassisProductLeaf.Vendor*.
- Storage processor (SP) Product Information
SP product version: *EMC_VNXe_SPChassisProductLeaf.Version*.
SP serial number: *EMC_VNXe_SPChassisProductLeaf.IdentifyingNumber*.
SP vendor: *EMC_VNXe_SPChassisProductLeaf.Vendor*.
- Disk processor enclosure (DPE) Product Information
DPE product version: *EMC_VNXe_DPEEnclosureChassisProductLeaf.Version*.
DPE serial number: *EMC_VNXe_DPEEnclosureChassisProductLeaf.IdentifyingNumber*.
DPE Vendor: *EMC_VNXe_DPEEnclosureChassisProductLeaf.Vendor*.
- Disk -array enclosure (DAE) Product Information
DAE product version: *EMC_VNXe_DAEEnclosureChassisProductLeaf.Version*.
DAE serial number: *EMC_VNXe_DAEEnclosureChassisProductLeaf.IdentifyingNumber*.
DAE vendor: *EMC_VNXe_DAEEnclosureChassisProductLeaf.Vendor*.
- Disk product information
Disk product version: *EMC_VNXe_DiskProductLeaf.Version*.
Disk serial number: *EMC_VNXe_DiskProductLeaf.IdentifyingNumber*.
Disk vendor: *EMC_VNXe_DiskProductLeaf.Vendor*.
- Fan product information
Fan product version: *EMC_VNXe_FanProductLeaf.Version*.
Fan serial number: *EMC_VNXe_FanProductLeaf.IdentifyingNumber*.
Fan vendor: *EMC_VNXe_FanProductLeaf.Vendor*.
- Power supply (PS) product information
PS product version: *EMC_VNXe_PowerSupplyProductLeaf.Version*.
PS serial number: *EMC_VNXe_PowerSupplyProductLeaf.IdentifyingNumber*.
PS vendor: *EMC_VNXe_PowerSupplyProductLeaf.Vendor*.

Use case: Get product information for a ComputerSystem

This use case describes how to get product information for a ComputerSystem (including Top-level systems and subcomponent systems):

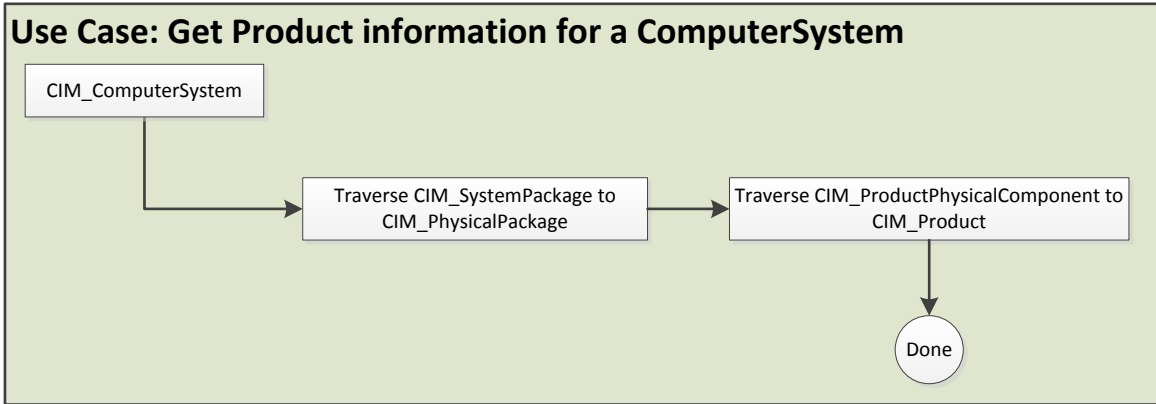


Figure 10 - Flowchart of getting product information for a ComputerSystem

Use case: Find asset information for subcomponents

This use case describes how to find asset information for subcomponents.

There are certain subcomponents of a device that a client may want to locate such as disks in an array, or power supply devices. To locate the asset information of these subcomponents, follow the *ProductParentChild* association from the system *Product* to lower-level *Products*.

CIM Elements

The implemented classes and associations related to the Physical Package Package for VNXe storage systems are described as follows:

Table 26 - CIM Elements for Physical Package Package

CIM Class	Implemented Class	Description
CIM_Container	EMC_VNXe_ArrayChassis_DAEEnclosureChassis_ContainerAssocLeaf	Represents the association between VNXe Array Chassis and Dis Array Enclosure (DAE).
	EMC_VNXe_ArrayChassis_DPEEnclosureChassis_ContainerAssocLeaf	Represents the association between VNXe Array Chassis and Disk Processor Enclosure (DPE).
	EMC_VNXe_DAEEnclosureChassis_DiskModule_ContainerAssocLeaf	Represents the association between DAE and Disk drive.
	EMC_VNXe_DAEEnclosureChassis_FanModule_ContainerAssocLeaf	Represents the association between DAE and Fan.
	EMC_VNXe_DAEEnclosureChassis_PowerSupplyModule_ContainerAssocLeaf	Represents the association between DAE and Power Supply.
	EMC_VNXe_DPEEnclosureChassis_DiskModule_ContainerAssocLeaf	Represents the association between DPE and Disk drive.
	EMC_VNXe_DPEEnclosureChassis_FanModule_ContainerAssocLeaf	Represents the association between DPE and Fan.

CIM Class	Implemented Class	Description
	EMC_VNXe_DPEEnclosureChassis_PowerSupplyModule_ContainerAssocLeaf	Represents the association between DPE and Power Supply.
	EMC_VNXe_DPEEnclosureChassis_SPChassis_ContainerAssocLeaf	Represents the association between DPE and Storage Processor Chassis (SP).
CIM_PhysicalPackage (System)	EMC_VNXe_ArrayChassisLeaf	Represents the Physical Package of Array System.
CIM_PhysicalPackage (Component)	EMC_VNXe_DAEEnclosureChassisLeaf	Represents the Physical Package of DAE.
	EMC_VNXe_DPEEnclosureChassisLeaf	Represents the Physical Package of DPE.
	EMC_VNXe_SPChassisLeaf	Represents the Physical Package of SP.
	EMC_VNXe_DiskModuleLeaf	Represents the Physical Package of Disk Drive.
	EMC_VNXe_FanModuleLeaf	Represents the Physical Package of Fan.
	EMC_VNXe_PowerSupplyModuleLeaf	Represents the Physical Package of Power Supply.
CIM_Product (System)	EMC_VNXe_ArrayChassisProductLeaf	Represents the Product information of Array System.
CIM_Product (Component)	EMC_VNXe_DAEEnclosureChassisProductLeaf	Represents the Product information of DAE.
	EMC_VNXe_DPEEnclosureChassisProductLeaf	Represents the Product information of DPE.
	EMC_VNXe_SPChassisProductLeaf	Represents the Product information of SP.
	EMC_VNXe_DiskProductLeaf	Represents the Product information of Disk Drive.
	EMC_VNXe_FanProductLeaf	Represents the Product information of Fan.
	EMC_VNXe_PowerSupplyProductLeaf	Represents the Product information of Power Supply.
CIM_ProductParentChild	EMC_VNXe_ArrayChassisProduct_DAEEnclosureChassisProduct_ProductParentChildAssocLeaf	Represents the association between Array Product information and DAE Product information.

CIM Class	Implemented Class	Description
	EMC_VNXe_ArrayChassisProduct_DPEEnclosureChassisProduct_ProductParentChildAssocLeaf	Represents the association between Array Product information and DPE Product information.
	EMC_VNXe_DAEEnclosureChassisProduct_DiskProduct_ProductParentChildAssocLeaf	Represents the association between DAE Product information and Disk Product information.
	EMC_VNXe_DAEEnclosureChassisProduct_FanProduct_ProductParentChildAssocLeaf	Represents the association between DAE Product information and Fan Product information.
	EMC_VNXe_DAEEnclosureChassisProduct_PowerSupplyProduct_ProductParentChildAssocLeaf	Represents the association between DAE Product information and Power Supply Product information.
	EMC_VNXe_DPEEnclosureChassisProduct_DiskProduct_ProductParentChildAssocLeaf	Represents the association between DPE Product information and Disk Product information.
	EMC_VNXe_DPEEnclosureChassisProduct_FanProduct_ProductParentChildAssocLeaf	Represents the association between DPE Product information and Fan Product information.
	EMC_VNXe_DPEEnclosureChassisProduct_PowerSupplyProduct_ProductParentChildAssocLeaf	Represents the association between DPE Product information and Power Supply Product information.
	EMC_VNXe_DPEEnclosureChassisProduct_SPChassisProduct_ProductParentChildAssocLeaf	Represents the association between DPE Product information and SP Product information.
CIM_ProductPhysicalComponent (System)	EMC_VNXe_ArrayChassis_ArrayChassisProduct_ProductPhysicalComponentAssocLeaf	Represents the association between the Array Product information and Array Physical Package.
CIM_ProductPhysicalComponent (Componet)	EMC_VNXe_DAEEnclosureChassis_DAEEnclosureChassisProduct_ProductPhysicalComponentAssocLeaf	Represents the association between DAE Product information and DAE Physical Package.
	EMC_VNXe_DPEEnclosureChassis_DPEEnclosureChassisProduct_ProductPhysicalComponentAssocLeaf	Represents the association between DPE Product information and DPE Physical Package.
	EMC_VNXe_SPChassis_SPChassisProduct_ProductPhysicalComponentAssocLeaf	Represents the association between SP Product information and SP Physical Package.
	EMC_VNXe_DiskModule_DiskProduct_ProductPhysicalComponentAssocLeaf	Represents the association between Disk Product information and Disk Physical Package.
	EMC_VNXe_FanModule_FanProduct_ProductPhysicalComponentAssocLeaf	Represents the association between Fan Product information and Fan Physical Package.

CIM Class	Implemented Class	Description
	EMC_VNXe_PowerSupplyModule_PowerSupplyProduct_ProductPhysicalComponentAssocLeaf	Represents the association between Power Supply Product information and Power Supply Physical Package.
CIM_SystemPackaging (System)	EMC_VNXe_StorageSystem_ArrayChassis_SystemPackagingAssocLeaf	Represents the association between Top-level Computer System and Array Physical Package.
CIM_SystemPackaging (Component)	EMC_VNXe_StorageProcessorSystem_SPChassis_SystemPackagingAssocLeaf	Represents the association between SP and SP Physical Package.

EMC_VNXe_ArrayChassisLeaf

Table 27 - Referenced properties/methods for EMC_VNXe_ArrayChassisLeaf

CIM property	Description/notes
CreationClassName	Set to EMC_VNXe_ArrayChassisLeaf
Tag	System name of VNXe system
Model	Model of VNXe system
SerialNumber	Serial number of VNXe system.
Manufacturer	Set to EMC Corporation
RemovalConditions	Set to 2: Not Applicable

EMC_VNXe_DAEEnclosureChassisLeaf

Table 28 - Referenced properties/methods for EMC_VNXe_DAEEnclosureChassisLeaf

CIM property	Description/notes
CreationClassName	Set to EMC_VNXe_DAEEnclosureChassisLeaf
Tag	Set to DAE
Model	Model of DAE
ElementName	Set to DAE
Name	Set to 'DAE'
PartNumber	Part number
SerialNumber	Serial number of DAE
Manufacturer	Vendor of DAE

CIM property	Description/notes
RemovalConditions	Set to 2: Not Applicable

EMC_VNXe_DPEEnclosureChassisLeaf

Table 29 - Referenced properties/methods for EMC_VNXe_DPEEnclosureChassisLeaf

CIM property	Description/notes
CreationClassName	Set to EMC_VNXe_DPEEnclosureChassisLeaf
Tag	Set to DPE
Model	Model of DPE
ElementName	Set to 'DPE
Name	Set to DPE
PartNumber	Part number
SerialNumber	Serial number of DPE
Manufacturer	Vendor of DPE
RemovalConditions	Set to 2: Not Applicable

EMC_VNXe_SPChassisLeaf

Table 30 - Referenced properties/methods for EMC_VNXe_SPChassisLeaf

CIM property	Description/notes
CreationClassName	Set to EMC_VNXe_SPChassisLeaf
Tag	Set to SP A or SP B
Model	Set to Storage Processor
SerialNumber	Serial number of SP
Manufacturer	Vendor of SP
RemovalConditions	Set to 2: Not Applicable

EMC_VNXe_DiskModuleLeaf

Table 31 - Referenced properties/methods for EMC_VNXe_DiskModuleLeaf

CIM property	Description/notes
CreationClassName	Set to EMC_VNXe_DiskModuleLeaf

CIM property	Description/notes
Tag	Uniquely identifiers of disk module
Model	Model
SerialNumber	Serial number
PartNumber	Part number
Manufacturer	Vendor
RemovalConditions	Set to 2: Not Applicable

EMC_VNXe_FanModuleLeaf

Table 32 - Referenced properties/methods for EMC_VNXe_FanModuleLeaf

CIM property	Description/notes
CreationClassName	Set to EMC_VNXe_FanModuleLeaf
Tag	Uniquely identifiers of fan module
Model	Model
SerialNumber	Serial number
PartNumber	Part number
Manufacturer	Vendor
RemovalConditions	Set to 2: Not Applicable

EMC_VNXe_PowerSupplyModuleLeaf

Table 33 - Referenced properties/methods for EMC_VNXe_PowerSupplyModuleLeaf

CIM property	Description/notes
CreationClassName	Set to EMC_VNXe_PowerSupplyModuleLeaf
Tag	Uniquely identifiers of power supply module
Model	Model
SerialNumber	Serial number
PartNumber	Part number
Manufacturer	Vendor
RemovalConditions	Set to 2: Not Applicable

EMC_VNXe_ArrayChassisProductLeaf

Table 34 - Referenced properties/methods for EMC_VNXe_ArrayChassisProductLeaf

CIM property	Description/notes
Name	The name of the product
Vendor	Set to EMC Corporation
Version	The version of the product
IdentifyingNumber	Set as serial number

EMC_VNXe_DAEEnclosureChassisProductLeaf

Table 35 - Referenced properties/methods for EMC_VNXe_DAEEnclosureChassisProductLeaf

CIM property	Description/notes
Name	The name of the product
Vendor	Set as EMC Corporation
Version	The version of the product
IdentifyingNumber	Set as serial number
ElementName	Set as Name

EMC_VNXe_DPEEnclosureChassisProductLeaf

Table 36 - Referenced properties/methods for EMC_VNXe_DPEEnclosureChassisProductLeaf

CIM property	Description/notes
Name	The name of the product
Vendor	Set as EMC Corporation
Version	The version of the product
IdentifyingNumber	Set as serial number
ElementName	Set as Name

EMC_VNXe_SPChassisProductLeaf

Table 37 - Referenced properties/methods for EMC_VNXe_SPChassisProductLeaf

CIM property	Description/notes
Name	The name of the product
Vendor	Set as EMC Corporation.

CIM property	Description/notes
Version	The version of the product.
IdentifyingNumber	Set as serial number.
ElementName	Set as Name.

EMC_VNXe_DiskProductLeaf

Table 38 - Referenced properties/methods for EMC_VNXe_DiskProductLeaf

CIM property	Description/notes
Name	The name of the product
Vendor	Vendor
Version	The version of the product
IdentifyingNumber	Set as serial number
ElementName	Set as Name

EMC_VNXe_FanProductLeaf

Table 39 - Referenced properties/methods for EMC_VNXe_FanProductLeaf

CIM property	Description/notes
Name	The name of the product
Vendor	Vendor
Version	The version of the product
IdentifyingNumber	Set as serial number
ElementName	Set as Name

EMC_VNXe_PowerSupplyProductLeaf

Table 40 - Referenced properties/methods for EMC_VNXe_PowerSupplyProductLeaf

CIM property	Description/notes
Name	The name of the product
Vendor	Vendor
Version	The version of the product
IdentifyingNumber	Set as serial number

CIM property	Description/notes
ElementName	Set as Name

EMC_VNXe_StorageSystem_ArrayChassis_SystemPackagingAssocLeaf

**Table 41 - Referenced properties/methods for
EMC_VNXe_StorageSystem_ArrayChassis_SystemPackagingAssocLeaf**

CIM property	Description/notes
Antecedent	Reference of ENC_VNXe_ArraySystemLeaf
Dependent	Reference of ENC_VNXe_ArrayChassisLeaf

EMC_VNXe_StorageProcessorSystem_SPChassis_SystemPackagingAssocLeaf

**Table 42 - Referenced properties/methods for
EMC_VNXe_StorageProcessorSystem_SPChassis_SystemPackagingAssocLeaf**

CIM property	Description/notes
Antecedent	Reference of ENC_VNXe_StorageProcessorSystemLeaf
Dependent	Reference of ENC_VNXe_SPChassisLeaf

EMC_VNXe_ArrayChassis_DAEEnclosureChassis_ContainerAssocLeaf

**Table 43 - Referenced properties/methods for
EMC_VNXe_ArrayChassis_DAEEnclosureChassis_ContainerAssocLeaf**

CIM property	Description/notes
GroupComponent	Reference of ENC_VNXe_ArrayChassisLeaf
PartComponent	Reference of ENC_VNXe_DAEEnclosureChassis Leaf

EMC_VNXe_ArrayChassis_DPEEnclosureChassis_ContainerAssocLeaf

**Table 44 - Referenced properties/methods for
EMC_VNXe_ArrayChassis_DPEEnclosureChassis_ContainerAssocLeaf**

CIM property	Description/notes
GroupComponent	Reference of ENC_VNXe_ArrayChassisLeaf
PartComponent	Reference of ENC_VNXe_DPEEnclosureChassisLeaf

EMC_VNXe_DAEEnclosureChassis_DiskModule_ContainerAssocLeaf

**Table 45 - Referenced properties/methods for
EMC_VNXe_DAEEnclosureChassis_DiskModule_ContainerAssocLeaf**

CIM property	Description/notes
GroupComponent	Reference of ENC_VNXe_DAEEnclosureChassisLeaf
PartComponent	Reference of ENC_VNXe_DiskModuleLeaf

EMC_VNXe_DAEEnclosureChassis_FanModule_ContainerAssocLeaf

**Table 46 - Referenced properties/methods for
EMC_VNXe_DAEEnclosureChassis_FanModule_ContainerAssocLeaf**

CIM property	Description/notes
GroupComponent	Reference of ENC_VNXe_DAEEnclosureChassisLeaf
PartComponent	Reference of ENC_VNXe_FanModuleLeaf

EMC_VNXe_DAEEnclosureChassis_PowerSupplyModule_ContainerAssocLeaf

**Table 47 - Referenced properties/methods for
EMC_VNXe_DAEEnclosureChassis_PowerSupplyModule_ContainerAssocLeaf**

CIM property	Description/notes
GroupComponent	Reference of ENC_VNXe_DAEEnclosureChassisLeaf
PartComponent	Reference of ENC_VNXe_PowerSupplyModuleLeaf

EMC_VNXe_DPEEnclosureChassis_DiskModule_ContainerAssocLeaf

**Table 48 - Referenced properties/methods for
EMC_VNXe_DPEEnclosureChassis_DiskModule_ContainerAssocLeaf**

CIM property	Description/notes
GroupComponent	Reference of ENC_VNXe_DPEEnclosureChassisLeaf
PartComponent	Reference of ENC_VNXe_DiskModuleLeaf

EMC_VNXe_DPEEnclosureChassis_FanModule_ContainerAssocLeaf

**Table 49 - Referenced properties/methods for
EMC_VNXe_DPEEnclosureChassis_FanModule_ContainerAssocLeaf**

CIM property	Description/notes
GroupComponent	Reference of ENC_VNXe_DPEEnclosureChassisLeaf
PartComponent	Reference of ENC_VNXe_FanModuleLeaf

EMC_VNXe_DPEEnclosureChassis_PowerSupplyModule_ContainerAssocLeaf

**Table 50 - Referenced properties/methods for
EMC_VNXe_DPEEnclosureChassis_PowerSupplyModule_ContainerAssocLeaf**

CIM property	Description/notes
GroupComponent	Reference of ENC_VNXe_DPEEnclosureChassisLeaf
PartComponent	Reference of ENC_VNXe_PowerSupplyModuleLeaf

EMC_VNXe_DPEEnclosureChassis_SPChassis_ContainerAssocLeaf

**Table 51 - Referenced properties/methods for
EMC_VNXe_DPEEnclosureChassis_SPChassis_ContainerAssocLeaf**

CIM property	Description/notes
GroupComponent	Reference of ENC_VNXe_DPEEnclosureChassisLeaf
PartComponent	Reference of ENC_VNXe_SPChassisLeaf

EMC_VNXe_ArrayChassis_ArrayChassisProduct_ProductPhysicalComponentAssocLeaf

**Table 52 - Referenced properties/methods for
EMC_VNXe_ArrayChassis_ArrayChassisProduct_ProductPhysicalComponentAssocLeaf**

CIM property	Description/notes
GroupComponent	Reference of ENC_VNXe_ArrayChassisProductLeaf
PartComponent	Reference of ENC_VNXe_ArrayChassisLeaf

EMC_VNXe_DAEEnclosureChassis_DAEEnclosureChassisProduct_ProductPhysicalComponentAssocLeaf

**Table 53 - Referenced properties/methods for
EMC_VNXe_DAEEnclosureChassis_DAEEnclosureChassisProduct_ProductPhysicalComponentAssocLeaf**

CIM property	Description/notes
GroupComponent	Reference of ENC_VNXe_DAEEnclosureChassisProductLeaf
PartComponent	Reference of ENC_VNXe_DAEEnclosureChassisLeaf

EMC_VNXe_DPEEnclosureChassis_DPEEnclosureChassisProduct_ProductPhysicalComponentAssocLeaf

**Table 54 - Referenced properties/methods for
EMC_VNXe_DPEEnclosureChassis_DPEEnclosureChassisProduct_ProductPhysicalComponentAssocLeaf**

CIM property	Description/notes
GroupComponent	Reference of ENC_VNXe_DPEEnclosureChassisProductLeaf
PartComponent	Reference of ENC_VNXe_DPEEnclosureChassisLeaf

EMC_VNXe_SPChassis_SPChassisProduct_ProductPhysicalComponentAssocLeaf

**Table 55 - Referenced properties/methods for
EMC_VNXe_SPChassis_SPChassisProduct_ProductPhysicalComponentAssocLeaf**

CIM property	Description/notes
GroupComponent	Reference of ENC_VNXe_SPChassisProductLeaf
PartComponent	Reference of ENC_VNXe_SPChassisLeaf

EMC_VNXe_DiskModule_DiskProduct_ProductPhysicalComponentAssocLeaf

**Table 56 - Referenced properties/methods for
EMC_VNXe_DiskModule_DiskProduct_ProductPhysicalComponentAssocLeaf**

CIM property	Description/notes
GroupComponent	Reference of ENC_VNXe_DiskProductLeaf
PartComponent	Reference of ENC_VNXe_DiskModuleLeaf

EMC_VNXe_FanModule_FanProduct_ProductPhysicalComponentAssocLeaf

**Table 57 - Referenced properties/methods for
EMC_VNXe_FanModule_FanProduct_ProductPhysicalComponentAssocLeaf**

CIM property	Description/notes
GroupComponent	Reference of ENC_VNXe_FanProductLeaf
PartComponent	Reference of ENC_VNXe_FanModuleLeaf

EMC_VNXe_PowerSupplyModule_PowerSupplyProduct_ProductPhysicalComponentAssocLeaf

**Table 58 - Referenced properties/methods for
EMC_VNXe_PowerSupplyModule_PowerSupplyProduct_ProductPhysicalComponentAssocLeaf**

CIM property	Description/notes
GroupComponent	Reference of ENC_VNXe_PowerSupplyProductLeaf
PartComponent	Reference of ENC_VNXe_PowerSupplyModuleLeaf

EMC_VNXe_ArrayChassisProduct_DAEEnclosureChassisProduct_ProductParentChildAssocLeaf

**Table 59 - Referenced properties/methods for
EMC_VNXe_ArrayChassisProduct_DAEEnclosureChassisProduct_ProductParentChildAssocLeaf**

CIM property	Description/notes
GroupComponent	Reference of ENC_VNXe_ArrayChassisProductLeaf

CIM property	Description/notes
PartComponent	Reference of ENC_VNXe_DAEEnclosureChassisProductLeaf

EMC_VNXe_ArrayChassisProduct_DPEEnclosureChassisProduct_ProductParentChildAssocLeaf

Table 60 - Referenced properties/methods for EMC_VNXe_ArrayChassisProduct_DPEEnclosureChassisProduct_ProductParentChildAssocLeaf

CIM property	Description/notes
GroupComponent	Reference of ENC_VNXe_ArrayChassisProductLeaf
PartComponent	Reference of ENC_VNXe_DPEEnclosureChassisProductLeaf

EMC_VNXe_DAEEnclosureChassisProduct_DiskProduct_ProductParentChildAssocLeaf

Table 61 - Referenced properties/methods for EMC_VNXe_DAEEnclosureChassisProduct_DiskProduct_ProductParentChildAssocLeaf

CIM property	Description/notes
GroupComponent	Reference of ENC_VNXe_DAEEnclosureChassisProductLeaf
PartComponent	Reference of ENC_VNXe_DiskProductLeaf

EMC_VNXe_DAEEnclosureChassisProduct_FanProduct_ProductParentChildAssocLeaf

Table 62 - Referenced properties/methods for EMC_VNXe_DAEEnclosureChassisProduct_FanProduct_ProductParentChildAssocLeaf

CIM property	Description/notes
GroupComponent	Reference of ENC_VNXe_DAEEnclosureChassisProductLeaf
PartComponent	Reference of ENC_VNXe_FanProductLeaf

EMC_VNXe_DAEEnclosureChassisProduct_PowerSupplyProduct_ProductParentChildAssocLeaf

Table 63 - Referenced properties/methods for EMC_VNXe_DAEEnclosureChassisProduct_PowerSupplyProduct_ProductParentChildAssocLeaf

CIM property	Description/notes
GroupComponent	Reference of ENC_VNXe_DAEEnclosureChassisProductLeaf
PartComponent	Reference of ENC_VNXe_PowerSupplyProductLeaf

EMC_VNXe_DPEEnclosureChassisProduct_DiskProduct_ProductParentChildAssocLeaf

**Table 64 - Referenced properties/methods for
EMC_VNXe_DPEEnclosureChassisProduct_DiskProduct_ProductParentChildAssocLeaf**

CIM property	Description/notes
GroupComponent	Reference of ENC_VNXe_DPEEnclosureChassisProductLeaf
PartComponent	Reference of ENC_VNXe_DiskProductLeaf

EMC_VNXe_DPEEnclosureChassisProduct_FanProduct_ProductParentChildAssocLeaf

**Table 65 - Referenced properties/methods for
EMC_VNXe_DPEEnclosureChassisProduct_FanProduct_ProductParentChildAssocLeaf**

CIM property	Description/notes
GroupComponent	Reference of ENC_VNXe_DPEEnclosureChassisProductLeaf
PartComponent	Reference of ENC_VNXe_FanProductLeaf

EMC_VNXe_DPEEnclosureChassisProduct_PowerSupplyProduct_ProductParentChildAssocLeaf

**Table 66 - Referenced properties/methods for
EMC_VNXe_DPEEnclosureChassisProduct_PowerSupplyProduct_ProductParentChildAssocLeaf**

CIM property	Description/notes
GroupComponent	Reference of ENC_VNXe_DPEEnclosureChassisProductLeaf
PartComponent	Reference of ENC_VNXe_PowerSupplyProductLeaf

EMC_VNXe_DPEEnclosureChassisProduct_SPChassisProduct_ProductParentChildAssocLeaf

**Table 67 - Referenced properties/methods for
EMC_VNXe_DPEEnclosureChassisProduct_SPChassisProduct_ProductParentChildAssocLeaf**

CIM property	Description/notes
GroupComponent	Reference of ENC_VNXe_DPEEnclosureChassisProductLeaf
PartComponent	Reference of ENC_VNXe_SPChassisProductLeaf

Access Points Subprofile

Overview

The Access Points Subprofile provides addresses of remote access points for management services. This is modeled using a RemoteServiceAccessPoint linked to the managed system with a HostedAccessPoint association.

NOTE: For more details, refer to *Clause 23: Access Points Subprofile in Storage Management Technical Specification, Part 2 Common Profiles, Version 1.6.0, Revision 4.*

Class diagram

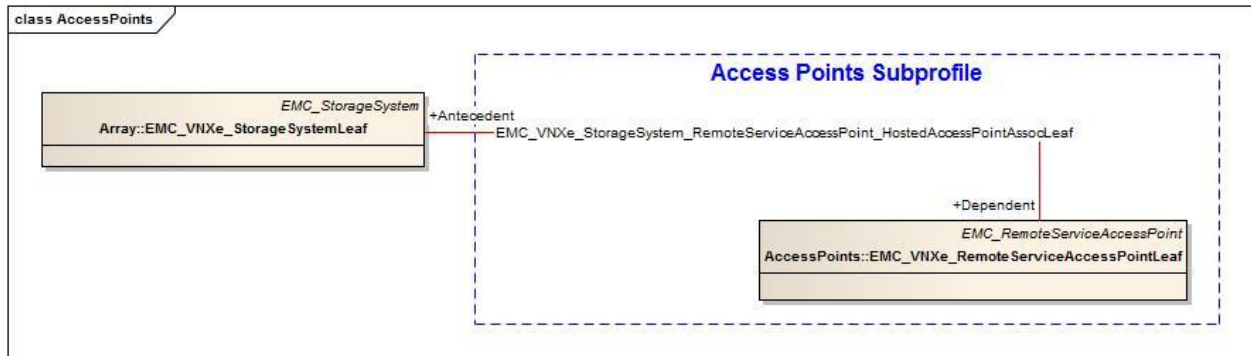


Figure 11 - Access Points Subprofile class diagram

Method of the Profile

This subprofile does not include any extrinsic methods.

Client considerations

N/A

CIM Element

The implemented classes and associations related to Access Points Subprofile in VNXe Storage System are described as follows:

Table 68 - CIM Elements for Access Points Subprofile

CIM Class	Implemented Class	Description
CIM_HostedAccessPoint	EMC_VNXe_StorageSystem_RemoteAccessPoint_HostedAccessPointLeaf	Associates the RemoteAccessPoint with the Hosting System.
CIM_RemoteServiceAccessPoint	EMC_VNXe_RemoteServiceAccessPointLeaf	Represents the remote service access point for the storage system.

EMC_VNXe_RemoteServiceAccessPointLeaf

Table 69 - Referenced properties/methods for EMC_VNXe_RemoteServiceAccessPointLeaf

CIM property	Description/notes
SystemCreationClassName	Set to EMC_VNXe_StorageSystemLeaf
SystemName	Set as system name of VNXe
CreationClassName	Set to EMC_VNXe_RemoteServiceAccessPointLeaf
Name	Unique ID of the access point
ElementName	Set to Remote Service Access Point
InfoFormat	Set to 3: IPv4 Address or 4: IPv6 Address' according to address type
AccessInfo	Set as IP address

EMC_VNXe_StorageSystem_RemoteAccessPoint_HostedAccessPointLeaf

**Table 70 - Referenced properties/methods for
EMC_VNXe_StorageSystem_RemoteAccessPoint_HostedAccessPointLeaf**

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_StorageSystemLeaf, the Hosting System
Dependent	Reference of EMC_VNXe_RemoteAccessPointLeaf hosted by system

FC Target Ports Subprofile

Overview

The FC Target Port Subprofile models the Fibre Channel-specific aspects of a VNXe Storage System. For Fibre Channel ports, the concrete subclass of *LogicalPort* is FCPort. FCPort is always associated 1-1 with a SCSIProtocolEndpoint instance.

NOTE: For more details, refer to *Clause 8: FC Target Port Profile in Storage Management Technical Specification, Part 2 Common Profiles, Version 1.6.0, Revision 4.*

Class diagram

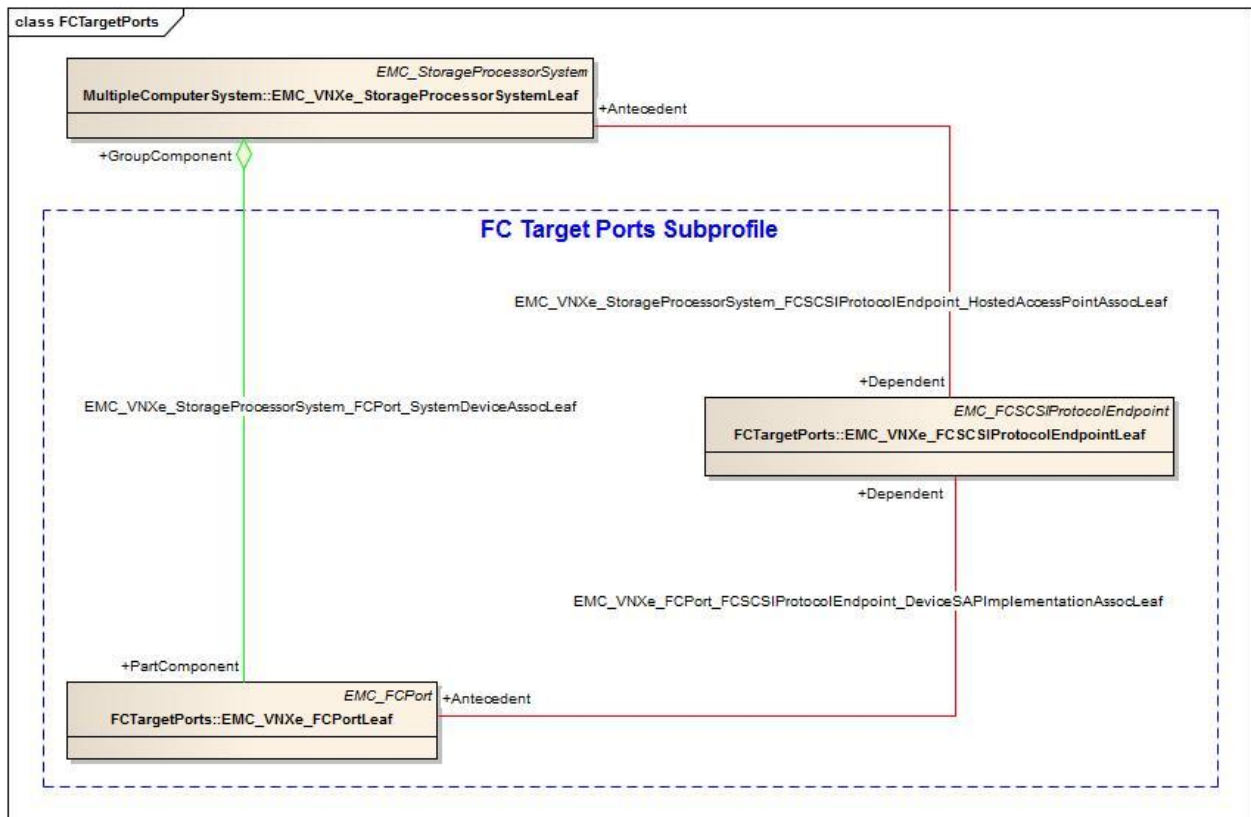


Figure 12 - FC Target Ports Subprofile class diagram

Methods of the Profile

This subprofile does not include any extrinsic methods.

Client considerations

N/A

CIM Elements

The implemented classes and associations related to FC Target Port Subprofile in VNXe Storage System are described as follows:

Table 71 - CIM Elements for FC Target Port Subprofile

CIM Class	Implemented Class	Description
CIM_DeviceSAPImplementation	EMC_VNXe_FCPort_FCSCSIProtocolEndpoint_DeviceSAPImplementationAssocLeaf	Associates FCPort and SCSIProtocolEndpoint.
CIM_FCPort	EMC_VNXe_FCPortLeaf	Represents the Fiber Channel Target Port.
CIM_HostedAccessPoint	EMC_VNXe_StorageProcessorSystem_FCSCSIProtocolEndpoint_HostedAccessPointAssocLeaf	Associates ComputerSystem to SCSIProtocolEndpoint.
CIM_SCSIProtocolEndpoint	EMC_VNXe_FCSCSIProtocolEndpointLeaf	Represents management characteristics related to the SCSI command set.
CIM_SystemDevice (Port)	EMC_VNXe_StorageProcessorSystem_FCPort_SystemDeviceAssocLeaf	Associates controller ComputerSystem to FCPort.

EMC_VNXe_FCPortLeaf

Table 72 - Referenced properties/methods for EMC_VNXe_FCPortLeaf

CIM property	Description/notes
SystemCreationClassName	Set to EMC_VNXe_StorageProcessorSystemLeaf
SystemName	SP system name
CreationClassName	Set to EMC_VNXe_FCPortLeaf
DeviceID	Device ID of the FC port
HealthState	Current health state of the port
LinkTechnology	Set to 4: FC
Name	Name of the port
ElementName	Same as Name
NetworkAddresses	Addresses assigned to the port
OperationalStatus	Operational status
PermanentAddresses	WWN (World Wide Name)
PortNumber	Port number
PortType	Set to 10: N

CIM property	Description/notes
UsageRestriction	Set to 2: Front_end_only
MaxSpeed	The maximum bandwidth of the port (bits/second)
Speed	Link speed

EMC_VNXe_FCSCSIProtocolEndpointLeaf

Table 73 - Referenced properties/methods for EMC_VNXe_FCSCSIProtocolEndpointLeaf

CIM property	Description/notes
SystemCreationClassName	Set to EMC_VNXe_StorageProcessorSystemLeaf
SystemName	SP system name
CreationClassName	Set to EMC_VNXe_FCSCSIProtocolEndpointLeaf
Name	WWN
Role	Set to 3: Target
ProtocolIFType	Set to 56: Fibre Channel
ConnectionType	Set to 2: Fibre Channel
OtherConnectionType	Empty
OtherTypeDescription	Set to SCSI

EMC_VNXe_StorageProcessorSystem_FCPort_SystemDeviceAssocLeaf

Table 74 - Referenced properties/methods for EMC_VNXe_StorageProcessorSystem_FCPort_SystemDeviceAssocLeaf

CIM property	Description/notes
GroupComponent	Reference of the SP ComputerSystem.
PartComponent	Reference of the FC Port hosted by the SP ComputerSystem.

EMC_VNXe_StorageProcessorSystem_FCSCSIProtocolEndpoint_HostedAccessPointAssocLeaf

Table 75 - Referenced properties/methods for EMC_VNXe_StorageProcessorSystem_FCSCSIProtocolEndpoint_HostedAccessPointAssocLeaf

CIM property	Description/notes
Antecedent	Reference of the SP ComputerSystem.
Dependent	Reference of the FC SCSI protocol endpoint hosted by the SP ComputerSystem.

EMC_VNXe_FCPort_FCSCSIProtocolEndpoint_DeviceSAPImplementationAssocLeaf

**Table 76 - Referenced properties/methods for
EMC_VNXe_FCPort_FCSCSIProtocolEndpoint_DeviceSAPImplementationAssocLeaf**

CIM property	Description/notes
Antecedent	Reference of the FC SCSI protocol endpoint implemented on the FC Port.
Dependent	Reference of the FC Port.

iSCSI Target Ports Subprofile

Overview

The iSCSI Target Ports Subprofile describes the iSCSI-specific aspects of a target device.

In the VNXe SMI-S Provider, it is a read-only implementation. None of the extrinsic methods of iSCSI Target Ports Subprofile are supported since the implementation does not provide any instances of *CIM_iSCSIConfigurationService*. No iSCSI node manipulation is provided.

NOTE: For more details, refer to *Clause 9: iSCSI Target Port Subprofile in Storage Management Technical Specification, Part 2 Common Profiles, Version 1.6.0, Revision 4.*

Class diagram

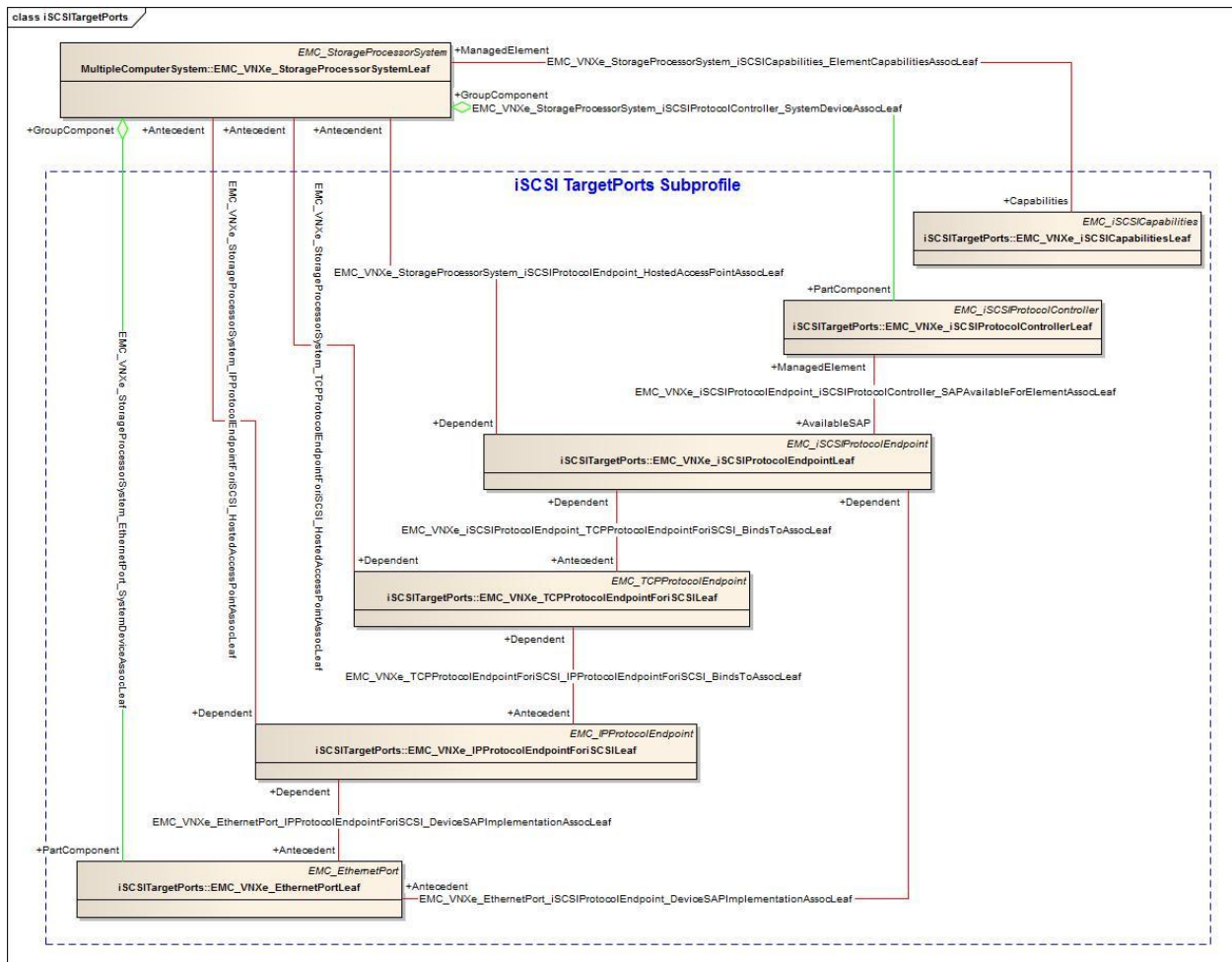


Figure 13 - iSCSI Target Ports Subprofile class diagram

Methods of the Profile

This subprofile does not include any extrinsic methods.

Client considerations

Model Specification

iSCSI node/port manipulation

There is no configuration service for iSCSI target ports in the VNXe SMI-S Provider. When IP/VLAN is configured, the iSCSI view is stable and remains unchanged.

Use case: Discover the iSCSI target port capabilities

This use case describes how to discover capabilities of an iSCSI target port.

From the *ComputerSystem* representing the storage processor (the NetworkEntity), traverse *CIM_ElementCapabilities* to *CIM_iSCSICapabilities* to obtain the capabilities of iSCSI Target Port.

Use case: Identify the iSCSI nodes in a target system (NetworkEntity).

This use case describes how to identify all iSCSI nodes in a given NetworkEntity.

From the *ComputerSystem* representing the NetworkEntity, traverse *CIM_SystemDevice* to *CIM_SCSIProtocolController* to obtain all iSCSI nodes.

Use case: Identify the iSCSI ports on a given iSCSI node.

This use case describes how to identify all iSCSI Ports on a given iSCSI Node.

From the *CIM_SCSIProtocolController* representing the iSCSI node, traverse *CIM_SAPAvailableForElement* to *CIM_iSCSIProtocolEndpoint* to obtain all iSCSI ports on the node.

CIM Elements

The implemented classes and associations related to iSCSI Target Port Subprofile in VNXe Storage System are described as follows:

Table 77 - CIM Elements for iSCSI Target Port

CIM Class	Implemented Class	Description
CIM_BindsTo (iSCSI to TCP)	EMC_VNXe_iSCSIProtocolEndpoint_TCPProtocolEndpointForiSCSI_BindsToAssocLeaf	Represents the association between an iSCSI port and a TCP protocol endpoint.
CIM_BindsTo (TCP to IP)	EMC_VNXe_TCPProtocolEndpointForiSCSI_IPProtocolEndpointForiSCSI_BindsToAssocLeaf	Represents the association between TCP protocol endpoint and an IP protocol endpoint.
CIM_DeviceSAPImplementation (EthernetPort to IP)	EMC_VNXe_EthernetPort_IPProtocolEndpointForiSCSI_DeviceSAPImplementationAssocLeaf	Represents the association between the Ethernet port and IP protocol endpoint.
CIM_DeviceSAPImplementation (EthernetPort to iSCSI)	EMC_VNXe_EthernetPort_iSCSIProtocolEndpoint_DeviceSAPImplementationAssocLeaf	Represents the association between an Ethernet port and an iSCSI port.
CIM_ElementCapabilities	EMC_VNXe_StorageProcessorSystem_iSCSICapabilities_ElementCapabilitiesAssocLeaf	Represents the association between the computer system and the capabilities for an iSCSI entity.

CIM_EthernetPort	EMC_VNXe_EthernetPortLeaf	Represents the logical aspects of the physical port.
CIM_HostedAccessPoint (NetworkEntity to IP)	EMC_VNXe_StorageProcessorSystem_IPProtocolEndpointForiSCSI_HostedAccessPointAssocLeaf	Represents the association between the computer system and an IP protocol endpoint.
CIM_HostedAccessPoint (NetworkEntity to iSCSI)	EMC_VNXe_StorageProcessorSystem_iSCSIProtocolEndpoint_HostedAccessPointAssocLeaf	Represents the association between the computer system and an iSCSI port.
CIM_HostedAccessPoint (NetworkEntity to TCP)	EMC_VNXe_StorageProcessorSystem_TCPProtocolEndpointForiSCSI_HostedAccessPointAssocLeaf	Represents the association between the computer system and a TCP protocol endpoint.
CIM_IPProtocolEndpoint	EMC_VNXe_IPProtocolEndpointForiSCSILeaf	Represents a ProtocolEndpoint that is dedicated to running IP.
CIM_iSCSICapabilities	EMC_VNXe_iSCSICapabilitiesLeaf	Represents the capabilities for an iSCSI NetworkEntity.
CIM_iSCSIProtocolEndpoint	EMC_VNXe_iSCSIProtocolEndpointLeaf	Represents an iSCSI port, which is a subset of network portals (TCP+IP ProtocolEndpoints) in a Portal Group by an iSCSI node.
CIM_SAPAvailableForElement	EMC_VNXe_iSCSIProtocolEndpoint_iSCSIProtocolController_SAPAvailableForElementAssocLeaf	Represents the association between an iSCSI port and an iSCSI node.
CIM_SCSIProtocolController	EMC_VNXe_iSCSIProtocolControllerLeaf	Represents an iSCSI node.
CIM_SystemDevice (NetworkEntity to EthernetPort)	EMC_VNXe_StorageProcessorSystem_EthernetPort_SystemDeviceAssocLeaf	Represents the association between the computer system and an Ethernet port.
CIM_SystemDevice (NetworkEntity to SCSIProtocolController)	EMC_VNXe_StorageProcessorSystem_iSCSIProtocolController_SystemDeviceAssocLeaf	Represents the association between the computer system and an iSCSI node.
CIM_TCPProtocolEndpoint	EMC_VNXe_TCPProtocolEndpointForiSCSILeaf	Represents a protocol endpoint that is dedicated to running TCP.

EMC_VNXe_iSCSICapabilitiesLeaf

Table 78 - Referenced properties/methods for EMC_VNXe_iSCSICapabilitiesLeaf

CIM property	Description/notes
InstanceID	Set as user friendly name of the scoping ComputerSystem (SP)
ElementName	Human readable description of the instance
MinimumSpecificationVersionSupported	Set to 20

CIM property	Description/notes
MaximumSpecificationVersionSupported	Set to 20
AuthenticationMethodsSupported	Set to [4: CHAP]
SupportedFeatures	Set to [0: None]

EMC_VNXe_iSCSIProtocolControllerLeaf

Table 79 - Referenced properties/methods for EMC_VNXe_iSCSIProtocolControllerLeaf

CIM property	Description/notes
SystemCreationClassName	The CreationClassName of the scoping system (SP). Set to EMC_VNXe_StorageProcessorSystemLeaf.
SystemName	The System Name of the scoping system
CreationClassName	Set to "EMC_VNXe_iSCSIProtocolControllerLeaf"
DeviceID	A unique name for the LogicalDevice
Name	iSCSI Name
NameFormat	Set to [3: iSCSI Name]
ElementName	Alias of the node

EMC_VNXe_EthernetPortLeaf

Table 80 - Referenced properties/methods for EMC_VNXe_EthernetPortLeaf

CIM property/method	Value/implementation
SystemCreationClassName	The CreationClassName of the scoping system (SP). Set to EMC_VNXe_StorageProcessorSystemLeaf.
SystemName	The System Name of the scoping system
CreationClassName	Set to EMC_VNXe_EthernetPortLeaf
DeviceID	A unique name for the port
HealthState	Current health of the port
PermanentAddress	Set to 000000000000
Name	Name of the Port
ElementName	Same as Name
NetworkAddresses	Network addresses

CIM property/method	Value/implementation
OperationalStatus	Current operational status of the Port
PortNumber	Set as Port ID
PortType	NS_CMP_OSLS_API:: CIM_LogicalPort::PortType_Unknown
UsageRestriction	Set to 2: Front-end only
MaxSpeed	Maximum bandwidth of the port (bits/second)
Speed	Current bandwidth of the port (bits/second)
LinkTechnology	Set to 2: Ethernet
FullDuplex	Set to TRUE

EMC_VNXe_iSCSIProtocolEndpointLeaf

Table 81 - Referenced properties/methods for EMC_VNXe_iSCSIProtocolEndpointLeaf

CIM property/method	Value/implementation
SystemCreationClassName	The CreationClassName of the scoping system (SP). Set to EMC_VNXe_StorageProcessorSystemLeaf
SystemName	The System Name of the scoping system
CreationClassName	Set to EMC_VNXe_iSCSIProtocolEndpointLeaf
Name	Name of the protocol endpoint
OperationalStatus	Current operational status of the ProtocolEndpoint
ProtocolIFType	Set to 1: Other
ConnectionType	Set to 7: iSCSI
Identifier	Set as Portal Group Tag
Role	Set to 3: Target
OtherTypeDescription	Set to SCSI
NameFormat	Set to <iSCSI node name> + 't,' + <TPGT>
Description	Set to MIB.IETF IF-MIB.ifDescr

EMC_VNXe_TCPProtocolEndpointForiSCSILeaf

Table 82 - Referenced properties/methods for EMC_VNXe_TCPProtocolEndpointForiSCSILeaf

CIM property/method	Value/implementation
---------------------	----------------------

CIM property/method	Value/implementation
SystemCreationClassName	The CreationClassName of the scoping system (SP). Set to EMC_VNXe_StorageProcessorSystemLeaf
SystemName	The System Name of the scoping system
CreationClassName	Set to EMC_VNXe_TCPProtocolEndpointLeaf
Name	Name of the protocol endpoint
OperationalStatus	Current operational status of the ProtocolEndpoint
ProtocolIFType	Set to 4111: TCP
PortNumber	Set to 3260
NameFormat	Set to ISCSI
Description	Set to TCP Protocol Endpoint

EMC_VNXe_IPProtocolEndpointForiSCSILeaf

Table 83 - Referenced properties/methods for EMC_VNXe_IPProtocolEndpointForiSCSILeaf

CIM property/method	Value/implementation
SystemCreationClassName	The CreationClassName of the scoping system (SP). Set to EMC_VNXe_StorageProcessorSystemLeaf
SystemName	The System Name of the scoping system
CreationClassName	Set to EMC_VNXe_IPProtocolEndpointLeaf
Name	Name of the protocol endpoint
OperationalStatus	Current operational status of the ProtocolEndpoint
HealthState	Current health status of the ProtocolEndpoint
ProtocolIFType	Set to 4098: IPv4/v6
IPv4Address	IPv4 address
IPv6Address	IPv6 address
SubnetMask	Net mask
NameFormat	Set to EMC_UIS_ISCSINetworkPortal.FriendlyID

EMC_VNXe_StorageProcessorSystem_iSCSICapabilities_ElementCapabilitiesAssocLeaf

Table 84 - Referenced properties/methods for EMC_VNXe_StorageProcessorSystem_iSCSICapabilities_ElementCapabilitiesAssocLeaf

CIM property	Description/notes
ManagedElement	Reference of the scoping system (SP)
Capabilities	Reference of EMC_VNXe_iSCSICapabilitiesLeaf

EMC_VNXe_StorageProcessorSystem_IPProtocolEndpointForiSCSI_HostedAccessPointAssocLeaf

Table 85 - Referenced properties/methods for EMC_VNXe_StorageProcessorSystem_IPProtocolEndpointForiSCSI_HostedAccessPointAssocLeaf

CIM property	Description/notes
Antecedent	Reference of the scoping system (SP)
Dependent	Reference of EMC_VNXe_IPProtocolEndpointForiSCSI Leaf

EMC_VNXe_StorageProcessorSystem_TCPProtocolEndpointForiSCSI_HostedAccessPointAssocLeaf

Table 86 - Referenced properties/methods for EMC_VNXe_StorageProcessorSystem_TCPProtocolEndpointForiSCSI_HostedAccessPointAssocLeaf

CIM property	Description/notes
Antecedent	Reference of the scoping system (SP)
Dependent	Reference of EMC_VNXe_TCPProtocolEndpointForiSCSI Leaf

EMC_VNXe_StorageProcessorSystem_iSCSIProtocolEndpoint_HostedAccessPointAssocLeaf

Table 87 - Referenced properties/methods for EMC_VNXe_StorageProcessorSystem_iSCSIProtocolEndpoint_HostedAccessPointAssocLeaf

CIM property	Description/notes
Antecedent	Reference of the scoping system (SP)
Dependent	Reference of EMC_VNXe_iSCSIProtocolEndpointLeaf

EMC_VNXe_iSCSIProtocolEndpoint_iSCSIProtocolController_SAPAvailableForElementAssocLeaf

Table 88 - Referenced properties/methods for EMC_VNXe_iSCSIProtocolEndpoint_iSCSIProtocolController_SAPAvailableForElementAssocLeaf

CIM property	Description/notes
--------------	-------------------

CIM property	Description/notes
AvailableSAP	Reference of EMC_VNXe_iSCSIProtocolEndpointLeaf
ManagedElement	Reference of EMC_VNXe_iSCSIProtocolControllerLeaf

EMC_VNXe_EthernetPort_IPProtocolEndpointForiSCSI_DeviceSAPImplementationAssocLeaf

Table 89 - Referenced properties/methods for EMC_VNXe_EthernetPort_IPProtocolEndpointForiSCSI_DeviceSAPImplementationAssocLeaf

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_EthernetPortLeaf
Dependent	Reference of EMC_VNXe_IPProtocolEndpointLeaf

EMC_VNXe_EthernetPort_iSCSIProtocolEndpoint_DeviceSAPImplementationAssocLeaf

Table 90 - Referenced properties/methods for EMC_VNXe_EthernetPort_iSCSIProtocolEndpoint_DeviceSAPImplementationAssocLeaf

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_EthernetPortLeaf
Dependent	Reference of EMC_VNXe_iSCSIProtocolEndpointLeaf

EMC_VNXe_iSCSIProtocolEndpoint_TCPProtocolEndpointForiSCSI_BindsToAssocLeaf

Table 91 - Referenced properties/methods for EMC_VNXe_iSCSIProtocolEndpoint_TCPProtocolEndpointForiSCSI_BindsToAssocLeaf

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_TCPProtocolEndpointLeaf
Dependent	Reference of EMC_VNXe_iSCSIProtocolEndpointLeaf

EMC_VNXe_TCPProtocolEndpointForiSCSI_IPProtocolEndpointForiSCSI_BindsToAssocLeaf

Table 92 - Referenced properties/methods for EMC_VNXe_TCPProtocolEndpointForiSCSI_IPProtocolEndpointForiSCSI_BindsToAssocLeaf

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_IPProtocolEndpointLeaf
Dependent	Reference of EMC_VNXe_TCPProtocolEndpointLeaf

Fan Profile

Overview

The Fan Profile describes the properties and methods of fans in a managed system.

NOTE: For more details, refer to [Clause 33: Fan Profile in Storage Management Technical Specification, Part 2 Common Profiles, Version 1.5.0, Revision 6](#).

Class diagram

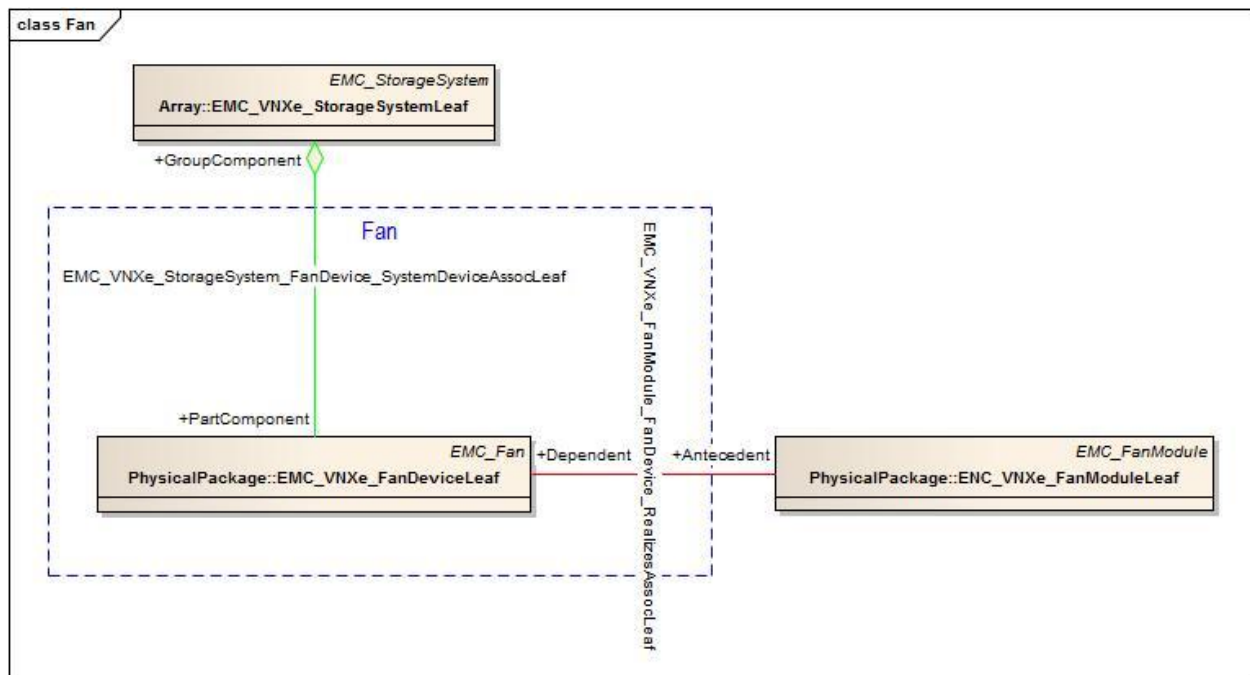


Figure 14 - Fan Profile class diagram

Method of the Profile

This subprofile does not include any extrinsic methods.

Client considerations

Model Specification

In VNXe, the Fan Profile is implemented as a read-only profile. There is no method for fan manipulation.

Use case: Discover fan devices

This use case describes how to discover fan devices.

From the `ComputerSystem` representing the storage system (the Top-Level Computer System), traverse `CIM_SystemDevice` to `CIM_Fan` to obtain all fan devices.

CIM Element

Table 93 - CIM Elements for Fan

CIM Class	Implemented Class	Description
CIM_Fan	EMC_VNXe_FanDeviceLeaf	The physical fan device.
CIM_Realizes	EMC_VNXe_FanModule_FanDevice_RealizesAssocLeaf	Association between fan device and its physical package.
CIM_SystemDevice	EMC_VNXe_StorageSystem_FanDevice_SystemDeviceAssocLeaf	Association between fan devices and the scoping system.

EMC_VNXe_FanDevice

Table 94 - Referenced properties/methods for EMC_VNXe_FanDeviceLeaf

CIM property	Description/notes
SystemCreationClassName	The CreationClassName of the scoping system Set to EMC_VNXe_StorageSystemLeaf
SystemName	The System Name of the scoping system
CreationClassName	Set to EMC_VNXe_FanDeviceLeaf
DeviceID	Unique name for the Fan
ElementName	User friendly name of the Fan
OperationalStatus	Current operational status of the Fan
HealthState	Current health state of the Fan
EnabledState	Enabled and disabled states of the Fan
RequestedState	Last requested or desired state of the Fan

EMC_VNXe_FanModule_FanDevice_RealizesAssocLeaf

Table 95 - Referenced properties/methods for EMC_VNXe_FanModule_FanDevice_RealizesAssocLeaf

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_FanModuleLeaf
Dependent	Reference of EMC_VNXe_FanDeviceLeaf

EMC_VNXe_StorageSystem_FanDevice_SystemDeviceAssocLeaf

**Table 96 - Referenced properties/methods for
EMC_VNXe_StorageSystem_FanDevice_SystemDeviceAssocLeaf**

CIM property	Description/notes
GroupComponent	Reference of EMC_VNXe_StorageSystemLeaf
PartComponent	Reference of EMC_VNXe_FanDeviceLeaf

Power Supply Profile

Overview

The Power Supply Profile describes the properties and methods of power supplies in a managed system.

NOTE: For more details, refer to *Clause 33: Power Supply Profile in Storage Management Technical Specification, Part 2 Common Profiles, Version 1.5.0, Revision 6.*

Class diagram

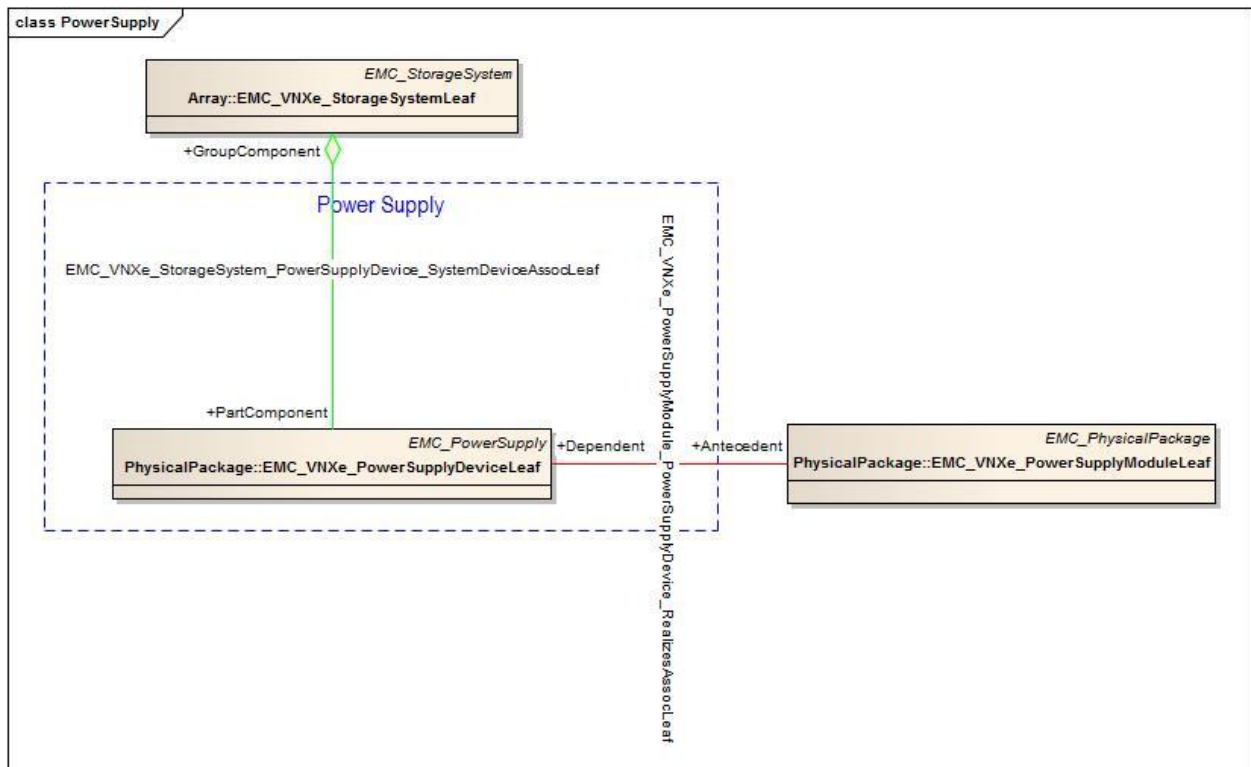


Figure 15 - Power Supply Profile class diagram

Method of the Profile

This subprofile does not include any extrinsic methods.

Client considerations

Model Specification

In VNXe, the Power Supply Profile is implemented as a read-only profile. There is no method for power supply manipulation.

Use case: Discover power supplies

This use case describes how to discover power supplies.

From the *ComputerSystem* representing the storage system (the Top-Level Computer System), traverse *CIM_SystemDevice* to *CIM_PowerSupply* to obtain all power supplies.

CIM Element

Table 97 - CIM Elements for Power Supply

CIM Class	Implemented Class	Description
CIM_PowerSupply	EMC_VNXe_PowerSupplyDeviceLeaf	The physical power supply device.
CIM_Realizes	EMC_VNXe_PowerSupplyModule_PowerSupplyDevice_RealizesAssocLeaf	Association between power supply device and its physical package.
CIM_SystemDevice	EMC_VNXe_StorageSystem_PowerSupplyDevice_SystemDeviceAssocLeaf	Association between power supply devices and the scoping system.

EMC_VNXe_PowerSupplyDevice

Table 98 - Referenced properties/methods for EMC_VNXe_PowerSupplyDeviceLeaf

CIM property	Description/notes
SystemCreationClassName	The CreationClassName of the scoping system Set to EMC_VNXe_StorageSystemLeaf
SystemName	The System Name of the scoping system
CreationClassName	Set to EMC_VNXe_PowerSupplyDeviceLeaf
DeviceID	Unique name for the Power Supply
ElementName	User friendly name of the Power Supply
OperationalStatus	Current operational status of the Power Supply
HealthState	Current health state of the Power Supply
EnabledState	Enabled and disabled states of the Power Supply
RequestedState	Last requested or desired state of the Power Supply

EMC_VNXe_PowerSupplyModule_PowerSupplyDevice_RealizesAssocLeaf

Table 99 - Referenced properties/methods for EMC_VNXe_PowerSupplyModule_PowerSupplyDevice_RealizesAssocLeaf

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_PowerSupplyModuleLeaf

CIM property	Description/notes
Dependent	Reference of EMC_VNXe_PowerSupplyDeviceLeaf

EMC_VNXe_StorageSystem_PowerSupplyDevice_SystemDeviceAssocLeaf

**Table 100 - Referenced properties/methods for
EMC_VNXe_StorageSystem_PowerSupplyDevice_SystemDeviceAssocLeaf**

CIM property	Description/notes
GroupComponent	Reference of EMC_VNXe_StorageSystemLeaf
PartComponent	Reference of EMC_VNXe_PowerSupplyDeviceLeaf

Health Package

Overview

Failures and abnormal occurrences are common when monitoring, controlling, and configuring devices and applications. An SMI-S client must be prepared to trap unexpected situations, and take appropriate action. This package defines the general mechanisms used in the expression of health in SMI-S. This package does not define the particular ways to report the health state.

NOTE: For VNXe, only *ComputerSystem.OperationalStatus* and *LogicalDevice.HealthState* are implemented to provide health state of components.

NOTE: For more details, refer to [Clause 25: Health Package in Storage Management Technical Specification, Part 2 Common Profiles, Version 1.5.0, Revision 6](#).

Class diagram

N/A

Method of the Profile

N/A

Client considerations

N/A

CIM Element

N/A

Indication Subprofile

Overview

The Indication Profile is a component profile of the Server Profile. It may also be a component profile of any other profile, such as the Array Profile. The Indication Profile allows a client to be notified when a particular event occurs.

NOTE: For more details, refer to [Clause 42: Indication Profile in Storage Management Technical Specification, Part 2 Common Profiles, Version 1.5.0, Revision 6.](#)

Class diagram

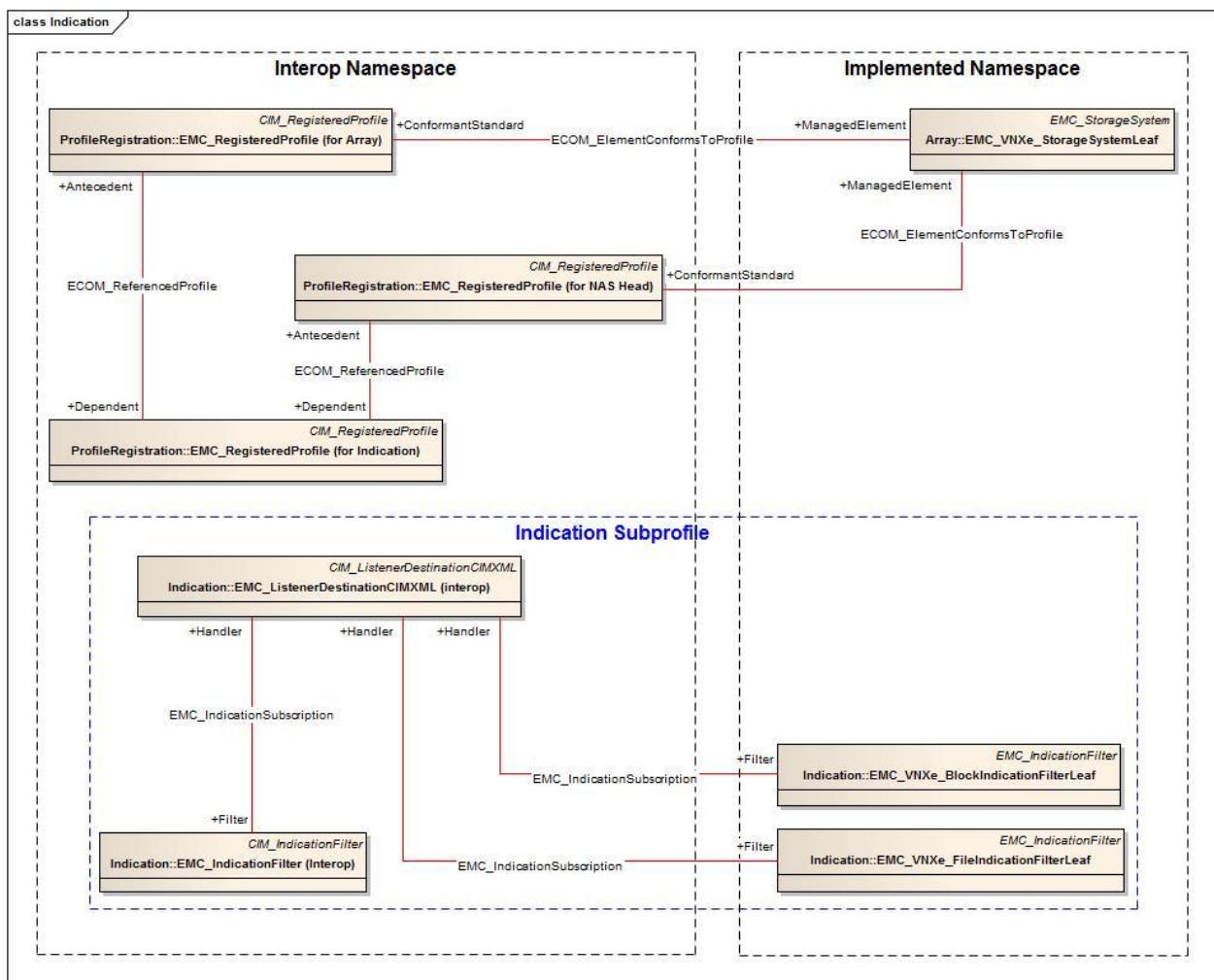


Figure 16 - Indication Subprofile class diagram

Method of the Profile

This profile does not include any extrinsic methods.

This profile supports instance creation, deletion and modification of indication filters, handlers, and subscriptions.

Table 101 - Supported Intrinsic Methods of Indication Profile

Class	CreateInstance	DeleteInstance	ModifyInstance
CIM_IndicationFilter	Y	Y	N
CIM_ListenerDestinationCIMXML	Y	Y	N
CIM_IndicationSubscription	Y	Y	N

Client considerations

Model specification

- Indication filter and handler

NOTE: Use the pre-defined filter for indication subscription. The behavior of a client-defined filter is not guaranteed.

[[X-Ref to table]] shows pre-defined indication filters supported by the VNXe SMI-S provider.

Table 102 - Supported pre-defined Indication filters

Indication filter name	Query
indCreateComputerSystem	SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_ComputerSystem
indModifyComputerSystem	SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_ComputerSystem AND SourceInstance.CIM_ComputerSystem::OperationalStatus <> PreviousInstance.CIM_ComputerSystem::OperationalStatus
indDeleteComputerSystem	SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_ComputerSystem
indModifyRedundancySet	SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_RedundancySet AND SourceInstance.CIM_RedundancySet::RedundancyStatus <> PreviousInstance.CIM_RedundancySet::RedundancyStatus
indCreateStoragePool	SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_StoragePool
indModifyStoragePool	SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_StoragePool AND SourceInstance.CIM_StoragePool::TotalManagedSpace <> PreviousInstance.CIM_StoragePool::TotalManagedSpace
indDeleteStoragePool	SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_StoragePool
indCreateStorageVolume	SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_StorageVolume

Indication filter name	Query
indModifyStorageVolume	SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_StorageVolume AND SourceInstance.CIM_StorageVolume::OperationalStatus <> PreviousInstance.CIM_StorageVolume::OperationalStatus
indDeleteStorageVolume	SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_StorageVolume
indCreateLogicalDisk	SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_LogicalDisk
indModifyLogicalDisk	SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_LogicalDisk AND SourceInstance.CIM_LogicalDisk::OperationalStatus <> PreviousInstance.CIM_LogicalDisk::OperationalStatus
indDeleteLogicalDisk	SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_LogicalDisk
indCreateStorageSynchronized	SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_StorageSynchronized
indDeleteStorageSynchronized	SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_StorageSynchronized
indCreateDiskDrive	SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_DiskDrive
indDeleteDiskDrive	SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_DiskDrive
indCreateAssociatedPrivilege	SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_AssociatedPrivilege
indDeleteAssociatedPrivilege	SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_AssociatedPrivilege
indCreateAuthorizedSubject	SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_AuthorizedSubject
indDeleteAuthorizedSubject	SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_AuthorizedSubject
indCreateProtocolController	SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_ProtocolController
indDeleteProtocolController	SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_ProtocolController
indCreateProtocolControllerForUnit	SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_ProtocolControllerForUnit
indModifyProtocolControllerForUnit	SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_ProtocolControllerForUnit

Indication filter name	Query
indDeleteProtocolControllerForUnit	SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_ProtocolControllerForUnit
indCreateFCPort	SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_FCPort
indDeleteFCPort	SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_FCPort
indCreateISCSIProtocolEndpoint	SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_iSCSIProtocolEndpoint
indDeleteISCSIProtocolEndpoint	SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_iSCSIProtocolEndpoint
indCreateSCSIProtocolController	SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_SCSIProtocolController
indDeleteSCSIProtocolController	SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_SCSIProtocolController
indModifyLogicalDevice	SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_LogicalDevice AND SourceInstance.CIM_LogicalDevice::HealthState <> PreviousInstance.CIM_LogicalDevice::HealthState
indCreateConcreteJob	SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_ConcreteJob
indModifyConcreteJob_PercentComplete	SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_ConcreteJob AND SourceInstance.CIM_ConcreteJob::PercentComplete <> PreviousInstance.CIM_ConcreteJob::PercentComplete
indModifyConcreteJob_OperationalStatus2	SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_ConcreteJob AND ANY SourceInstance.CIM_ConcreteJob::OperationalStatus[*] = 17 AND ANY SourceInstance.CIM_ConcreteJob::OperationalStatus[*] = 2
indModifyConcreteJob_OperationalStatus6	SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_ConcreteJob AND ANY SourceInstance.CIM_ConcreteJob::OperationalStatus[*] = 17 AND ANY SourceInstance.CIM_ConcreteJob::OperationalStatus[*] = 6
indModifyConcreteJob_JobState	SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_ConcreteJob AND SourceInstance.CIM_ConcreteJob::JobState <> PreviousInstance.CIM_ConcreteJob::JobState
indCreateFileServer	SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_Computer_System AND ANY SourceInstance.CIM_ComputerSystem::Dedicated[*] = 16
indDeleteFileServer	SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_Computer_System AND ANY SourceInstance.CIM_ComputerSystem::Dedicated[*] = 16

Indication filter name	Query
indCreateLocalFileSystem	SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA SNIA_LocalFileSystem
indModifyLocalFileSystem	SELECT * FROM CIM_InstModification WHERE SourceInstance ISA SNIA_LocalFileSystem
indCreateFileShare	SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_FileShare
indModifyFileShare	SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_FileShare
indDeleteFileShare	SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_FileShare

- PreviousInstance Support
PreviousInstance of InstModification is supported by the SMI-S Provider on VNXe.
- Indication Support
The SMI-S Provider on VNXe supports both alert indications and life cycle indications on the following CIM objects.
[[X-Ref to table]] shows Alert Indications supported in VNXe SMI-S Provider.

Table 103 - Supported alert Indications

AlertIndication	Description
POOL_CAP_WARN	Pool's or volume's capacity in use is near available limit.
POOL_CAP_CRITICAL	Pool's or volume's capacity in use exceeds available limit.

Table below shows Life Cycle indications supported in VNXe SMI-S Provider.

Table 104 - Supported life cycle Indications

CIM Object	Indication type	SMI-S Indication filters (pre-defined)
CIM_ComputerSystem	New	SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_ComputerSystem
	New	SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_Computer_System AND ANY SourceInstance.CIM_ComputerSystem::Dedicated[*] = 16
	Modify	SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_ComputerSystem AND SourceInstance.CIM_ComputerSystem::OperationalStatus <> PreviousInstance.CIM_ComputerSystem::OperationalStatus
	Delete	SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_ComputerSystem

CIM Object	Indication type	SMI-S Indication filters (pre-defined)
	Delete	SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_ComputerSystem AND ANY SourceInstance.CIM_ComputerSystem::Dedicated[*] = 16
CIM_RedundancySet	Modify	SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_RedundancySet AND SourceInstance.CIM_RedundancySet::RedundancyStatus <> PreviousInstance.CIM_RedundancySet::RedundancyStatus
CIM_StoragePool (Concrete Storage Pool)	New	SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_StoragePool
	Modify	SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_StoragePool AND SourceInstance.CIM_StoragePool::TotalManagedSpace <> PreviousInstance.CIM_StoragePool::TotalManagedSpace
	Delete	SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_StoragePool
CIM_StorageVolume	New	SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_StorageVolume
	Modify	SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_StorageVolume AND SourceInstance.CIM_StorageVolume::OperationalStatus <> PreviousInstance.CIM_StorageVolume::OperationalStatus
	Delete	SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_StorageVolume
CIM_LogicalDisk	New	SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_LogicalDisk
	Modify	SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_LogicalDisk AND SourceInstance.CIM_LogicalDisk::OperationalStatus <> PreviousInstance.CIM_LogicalDisk::OperationalStatus
	Delete	SELECT * FROM CIM_InstDeletion WHERE

CIM Object	Indication type	SMI-S Indication filters (pre-defined)
		SourceInstance ISA CIM_LogicalDisk
CIM_StorageSynchronized	New	SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_StorageSynchronized
	Delete	SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_StorageSynchronized
CIM_DiskDrive	New	SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_DiskDrive
	Delete	SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_DiskDrive
CIM_AssociatedPrivilege	New	SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_AssociatedPrivilege
	Delete	SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_AssociatedPrivilege
CIM_AuthorizedSubject	New	SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_AuthorizedSubject
	Delete	SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_AuthorizedSubject
CIM_ProtocolController	New	SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_ProtocolController
	Delete	SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_ProtocolController
CIM_ProtocolControllerForUnit	New	SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_ProtocolControllerForUnit
	Modify	SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_ProtocolControllerForUnit
	Delete	SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_ProtocolControllerForUnit
CIM_FCPort	New	SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_FCPort
	Modify	SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_FCPort

CIM Object	Indication type	SMI-S Indication filters (pre-defined)
		AND SourceInstance.CIM_FCPort::OperationalStatus <> PreviousInstance.CIM_FCPort::OperationalStatus
	Delete	SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_FCPort
CIM_iSCSIProtocolEndpoint	New	SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_iSCSIProtocolEndpoint
	Delete	SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_iSCSIProtocolEndpoint
CIM_SCSIProtocolController	New	SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_SCSIProtocolController
	Delete	SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_SCSIProtocolController
CIM_LogicalDevice	Modify	SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_LogicalDevice AND SourceInstance.CIM_LogicalDevice::HealthState <> PreviousInstance.CIM_LogicalDevice::HealthState
CIM_Fan	New	SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_Fan
	Delete	SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_Fan
	Modify	SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_Fan
CIM_PowerSupply	New	SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_PowerSupply
	Delete	SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_PowerSupply
	Modify	SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_PowerSupply
CIM_ConcreteJob	Create	SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_ConcreteJob
	Modify	SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_ConcreteJob AND SourceInstance.CIM_ConcreteJob::PercentComplete <> PreviousInstance.CIM_ConcreteJob::PercentComplete

CIM Object	Indication type	SMI-S Indication filters (pre-defined)
	Modify	SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_ConcreteJob AND ANY SourceInstance.CIM_ConcreteJob::OperationalStatus[*] = 17 AND ANY SourceInstance.CIM_ConcreteJob::OperationalStatus[*] = 2
	Modify	SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_ConcreteJob AND ANY SourceInstance.CIM_ConcreteJob::OperationalStatus[*] = 17 AND ANY SourceInstance.CIM_ConcreteJob::OperationalStatus[*] = 6
	Modify	SELECT * FROM CIM_InstModification WHERE SourceInstance ISA CIM_ConcreteJob AND SourceInstance.CIM_ConcreteJob::JobState <> PreviousInstance.CIM_ConcreteJob::JobState
CIM_FileShare	New	SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA CIM_FileShare
	Delete	SELECT * FROM CIM_InstDeletion WHERE SourceInstance ISA CIM_FileShare
SNIA_LocalFileSystem	Create	SELECT * FROM CIM_InstCreation WHERE SourceInstance ISA SNIA_LocalFileSystem
	Modify	SELECT * FROM CIM_InstModification WHERE SourceInstance ISA SNIA_LocalFileSystem

Use case: CQL for VNXe profiles

CIM Server and ObjectManger (ECOM) on VNXe supports the stand of DMTF:CQL for indication query. For more details, refer to "[DMTF: CIM Query Language Specification\(DSP0202\)](#)".

Use case: Create IndicationFilter and ListenerDestination instances

This use case describes how to create a client-defined indication filter and handler.

Invoke the *CreateInstance* intrinsic method on *IndicationFilter* and *ListenerDestinationCIMXML* to create a client-defined filter and handler.

For further information, refer to [Use Case: Subscribe Indication using Pre-Defined Filters](#).

Use case: Create IndicationSubscription instances

This use case describes how to subscribe to Indications.

Invoke the *CreateInstance* intrinsic method on *IndicationSubscription*.

For further information, refer to [Use Case: Subscribe Indication using Pre-Defined Filters](#).

Use case: Subscribe to Indications using pre-defined filters

This use case describes how to:

- Obtain the pre-defined Indication filter (instance of *IndicationFilter*) according to the query.
- Create an Indication handler (the instance of *ListenerDestinationCIMXML*);
- Create the *IndicationSubscription* (if one does not exist already) with the filter and handler.

After subscribing, go to the URL in *ListenerDestinationCIMXML.Destination* to receive the specified indications filtered by the client-defined filter.

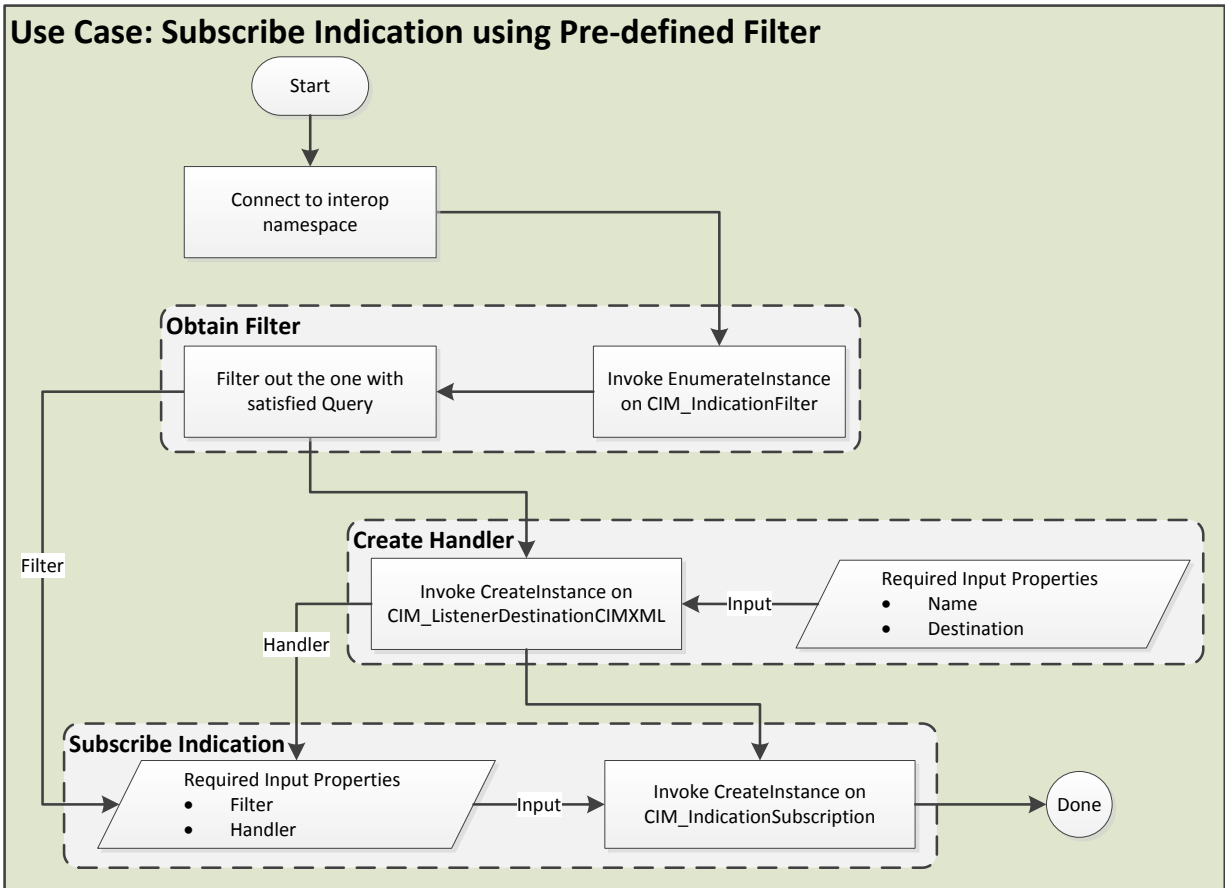


Figure 17 - Flowchart of Indication subscription

CIM Element

The implemented classes and associations related to the Indication Profile on the VNXe storage system are described as follows:

Table 105 - CIM Element in Indication Profile

CIM Class	Implemented Class	Description
CIM_AlertIndecation	CIM_AlertIndecation	Used to capture events on the VNXe Storage System.
CIM_InstCreation	CIM_InstCreation	An indication of the creation of a CIM instance.
CIM_InstDeletion	CIM_InstDeletion	An indication of the deletion of a CIM

CIM Class	Implemented Class	Description
		instance.
CIM_InstModification	CIM_InstModification	An indication of the modification of a CIM instance.
CIM_IndicationFilter	EMC_IndicationFilter (interop, pre-defined)	A system-defined filter that defines the value and the format of an indication filter string in the interop namespace.
CIM_IndicationFilter	EMC_IndicationFilter (interop, client-defined)	A client-defined filter that defines the value and the format of an indication filter string in the interop namespace.
CIM_IndicationFilter	EMC_VNXe_BlockIndicationFilter Leaf (implemented, pre-define)	A system-defined filter for block that defines the value and the format of an indication filter string in the implemented namespace.
CIM_IndicationFilter	EMC_VNXe_FileIndicationFilterLe af (implemented, pre-define)	A system-defined filter for file that defines the value and the format of an indication filter string in the implemented namespace.
CIM_IndicationSubscription	EMC_IndicationSubscription	Defines a subscription to a specific IndicationFilter instance by a specific indication handler (as represented by a ListenerDestinationCIMXML instance).
CIM_ListenerDestinationCIMXML	EMC_ListenerDestinationCIMXML	The destination for CIM Export Messages to be delivered via CIM-XML.

CIM_AlertIndecation

Table 106 - Referenced properties/methods for CIM_AlertIndication

CIM property	Description/notes
SystemCreationClassName	CreationClassName of the scoping system. Set to EMC_VNXe_StorageSystemLeaf
SystemName	Name of the scoping system
ProviderName	Set to SMIS
AlertingManagedElement	Information of the entity for which this indication is generated Set as object path of the element
AlertingElementFormat	Format of the AlertingManagedElement property Set to 2: CIMObjectPath
OwningEntity	Set to SNIA
MessageID	ID of the message

CIM property	Description/notes
Message	Text description of the alert
MessageArguments	Arguments list of the message
PerceivedSeverity	Severity of the Alert Indication from the notifier
AlertType	Primary classification of the indication

CIM_InstCreation

Table 107 - Referenced properties/methods for CIM_InstCreation

CIM property	Description/notes
IndicationTime	The time and date of creation of the indication.
SourceInstance	A copy of the instance that changed to generate the Indication. SourceInstance contains the current values of the properties selected by the Indication Filter's Query.
SourceInstanceModelPath	The Model Path of the SourceInstance.

CIM_InstDeletion

Table 108 - Referenced properties/methods for CIM_InstDeletion

CIM property	Description/notes
IndicationTime	The time and date of creation of the indication.
SourceInstance	A copy of the instance that changed to generate the Indication. SourceInstance contains the current values of the properties selected by the Indication Filter's Query.
SourceInstanceModelPath	The model path of the SourceInstance.

CIM_InstModification

Table 109 - Referenced properties/methods for CIM_InstModification

CIM property	Description/notes
IndicationTime	The time and data of creation of the indication.
SourceInstance	A copy of the instance that changed to generate the Indication. SourceInstance contains the current values of the properties selected by the Indication Filter's Query.
SourceInstanceModelPath	The model path of the SourceInstance.

EMC_IndicationFilter

Table 110 - Referenced properties/methods for EMC_IndicationFilter

CIM property	Description/notes
SystemCreationClassName	CreationClassName of the scoping system
SystemName	Name of the scoping system
CreationClassName	Name of class used to create this instance
Name	Name of the filter
Query	A query expression that defines the condition under which indication will be generated
QueryLanguage	The language in which the query is expressed. Set to DMTF:CQL
IndividualSubscriptionSupported	Set to FALSE
SourceNamespace	The path to a local namespace where the indication originate
SourceNamespaces	Names of the local namespaces where the indication originate

EMC_VNXe_BlockIndicationFilterLeaf

Table 111 - Referenced properties/methods for EMC_VNXe_BlockIndicationFilterLeaf

CIM property	Description/notes
SystemCreationClassName	CreationClassName of the scoping system
SystemName	Name of the scoping system
CreationClassName	Name of class used to create this instance
Name	Name of the filter
Query	A query expression that defines the condition under which indication will be generated
QueryLanguage	The language in which the query is expressed
IndividualSubscriptionSupported	Set to TRUE
SourceNamespace	The path to a local namespace where the indication originate
SourceNamespaces	Names of the local namespaces where the indication originate

EMC_VNXe_FileIndicationFilterLeaf

Table 112 - Referenced properties/methods for EMC_VNXe_FileIndicationFilterLeaf

CIM property	Description/notes
SystemCreationClassName	CreationClassName of the scoping system

CIM property	Description/notes
SystemName	Name of the scoping system
CreationClassName	Name of class used to create this instance
Name	Name of the filter
Query	A query expression that defines the condition under which indication will be generated
QueryLanguage	The language in which the query is expressed
IndividualSubscriptionSupported	Set to FALSE
SourceNamespace	The path to a local namespace where the indication originate
SourceNamespaces	Names of the local namespaces where the indication originate

EMC_ListenerDestinationCIMXML

Table 113 - Referenced properties/methods for EMC_ListenerDestinationCIMXML

CIM property	Description/notes
SystemCreationClassName	CreationClassName of the scoping system
SystemName	Name of the scoping system
CreationClassName	Name of class used to create this instance
Name	Name of the filter
Destination	The destination URL to which CIM-XML Export Messages are to be delivered

EMC_IndicationSubscription

Table 114 - Referenced properties/methods for EMC_IndicationSubscription

CIM property	Description/notes
Filter	Reference of EMC_IndicationFilter
Handler	Reference of EMC_ListenerDestinationCIMXML

Job Control Subprofile

Overview

In some profiles, such as the Block Service Package, some or all of the methods described may take some time to execute (longer than a HTTP timeout). In this case, a mechanism is needed to handle asynchronous execution of the method as a Job.

This subprofile defines the constructs and behavior for job control for SNIA profiles that make use of the subprofile.

When the client executes an asynchronous method, a reference to an instance of ConcreteJob is returned and the return value for the method is set to Method parameters checked - job started.

The associations OwningJobElement and AffectedJobElement indicate the service whose method created the job by side-effect, and the element being affected by the job. The job itself may create, modify, or delete many elements during its execution. The nature of this impact is the creation or deletion of the instances or associations, or the modification of instance properties. These elements are said to be affected by the job. The elements linked by AffectedJobElement may change through the execution of the job. In the VNXe implementation, for each job there will be at most one affected job element for each job.

NOTE: For more details, refer to [Clause 26: Job Control Subprofile in Storage Management Technical Specification, Part 2 Common Profiles, Version 1.5.0, Revision 6](#).

Class diagram

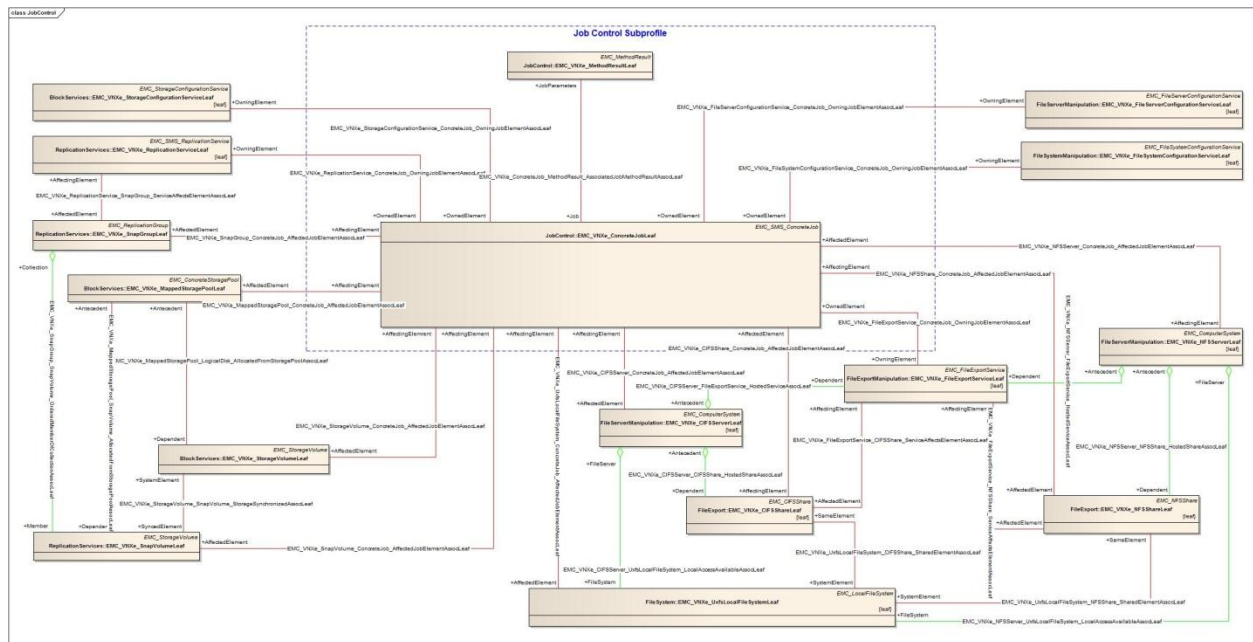


Figure 18 - Job Control Subprofile class diagram

Method of the Profile

This profile does not include any extrinsic methods.

This profile supports DeleteInstance on *CIM_ConcreteJob*.

Client considerations

Model specification

DeleteInstance on ConcreteJob

Instance of ConcreteJob with JobStatus completed ConcreteJob with can be deleted.

Use case: Obtain job-affected element

This use case describes how to obtain job-affected elements.

From the *CIM_ConcreteJob* returned from an extrinsic method, traverse *CIM_AffectedJobElement* to *CIM_ManagedElement* (Result role is AffectedElement) to obtain the job-affected element.

NOTE: The result affected element may be different during job execution. For example, a job to create a *StorageVolume* may start off pointing to a *Pool* until the *Volume* is instantiated, at which point the association would change to the *StorageVolume*.

NOTE: If a job is to delete something, the job-affected element may be a NULL reference when the job is completed.

CIM Element

The implemented classes and associations related to Job Control Subprofile on the VNXe storage system are described as follows:

Table 115 - CIM Elements for Job Control Subprofile

CIM Class	Implemented Class	Description
CIM_AssociatedJobMethodResult	EMC_VNXe_ConcreteJob_MethodResult_AssociatedJobMethodResultAssocLeaf	Represents the association between a job and its method result.
CIM_AffectedJobElement	EMC_VNXe_CIFSServer_ConcreteJob_AffectedJobElementAssocLeaf	Represents the association between a job and CIFSServer affected by the job.
CIM_AffectedJobElement	EMC_VNXe_CIFSShare_ConcreteJob_AffectedJobElementAssocLeaf	Represents the association between a job and CIFSShare affected by the job.
CIM_AffectedJobElement	EMC_VNXe_MappedStoragePool_ConcreteJob_AffectedJobElementAssocLeaf	Represents the association between a job and StoragePool affected by the job.
CIM_AffectedJobElement	EMC_VNXe_NFSServer_ConcreteJob_AffectedJobElementAssocLeaf	Represents the association between a job and NFSServer affected by the job.
CIM_AffectedJobElement	EMC_VNXe_NFSShare_ConcreteJob_AffectedJobElementAssocLeaf	Represents the association between a job and NFSShare affected by the job.
CIM_AffectedJobElement	EMC_VNXe_SnapGroup_ConcreteJob_AffectedJobElementAssocLeaf	Represents the association between a job and SnapGroup affected by the job.

CIM_AffectedJobElement	EMC_VNXe_SnapVolume_ConcreteJob_AffectedJobElementAssocLeaf	Represents the association between a job and SnapVolume affected by the job.
CIM_AffectedJobElement	EMC_VNXe_StorageVolume_ConcreteJob_AffectedJobElementAssocLeaf	Represents the association between a job and StorageVolume affected by the job.
CIM_AffectedJobElement	EMC_VNXe_UxfsLocalFileSystem_ConcreteJob_AffectedJobElementAssocLeaf	Represents the association between a job and FileSystem affected by the job.
CIM_ConcreteJob	EMC_VNXe_ConcreteJobLeaf	Job returned from extrinsic method executed asynchronously.
CIM_MethodResult	EMC_VNXe_MethodResultLeaf	The parameters for the method which by side-effect created the Job
CIM_OwningJobElement	EMC_VNXe_FileExportService_ConcreteJob_OwningJobElementAssocLeaf	Represents the association between a job and FileExportService that creates the job.
CIM_OwningJobElement	EMC_VNXe_FileServerConfigurationService_ConcreteJob_OwningJobElementAssocLeaf	Represents the association between a job and FileServerConfigurationService that creates the job.
CIM_OwningJobElement	EMC_VNXe_FileSystemConfigurationService_ConcreteJob_OwningJobElementAssocLeaf	Represents the association between a job and FileSystemConfigurationService that creates the job.
CIM_OwningJobElement	EMC_VNXe_ReplicationService_ConcreteJob_OwningJobElementAssocLeaf	Represents the association between a job and ReplicationService that creates the job.
CIM_OwningJobElement	EMC_VNXe_StorageConfigurationService_ConcreteJob_OwningJobElementAssocLeaf	Represents the association between a job and StorageConfigurationService that creates the job.

EMC_VNXe_ConcreteJobLeaf

Table 116 - Referenced properties/methods for EMC_VNXe_ConcreteJobLeaf

CIM property	Description/notes
InstanceID	Job ID
Description	Job Name
DeleteOnCompletion	Set to FALSE The Job won't be deleted automatically when it is completed. It remains until the user explicitly deletes it.
ElapsedTime	Amount of time taken to work on the job.

CIM property	Description/notes
ErrorCode	Error code
ErrorDescription	Error description
TimeBeforeRemoval	Set to 0 since the job won't be deleted automatically.
OperationalStatus	Current operational status of the job.
JobState	State of the job
JobStatus	Status of the job
LocalOrUtcTime	Set to 2: UTC Time
PercentComplete	Percentage of the job completed.
StartTime	Time when the job was stated
StatusDescriptions	Description of status
TimeOfLastStateChange	Latest time when the job state was changed
TimeSubmitted	Time when the job was submitted

EMC_VNXe_MethodResultLeaf

Table 117 - Referenced properties/methods for EMC_VNXe_MethodResultLeaf

CIM property	Description/notes
InstanceID	Job ID
PreCallIndication	Contains a CIM_InstMethodCall Indication that describes the pre-execution values of the extrinsic method invocation.
PostCallIndication	Contains a CIM_InstMethodCall Indication that describes the post-execution values of the extrinsic method invocation.

EMC_VNXe_ConcreteJob_MethodResult_AssociatedJobMethodResultAssocLeaf

Table 118 - Referenced properties/methods for EMC_VNXe_ConcreteJob_MethodResult_AssociatedJobMethodResultAssocLeaf

CIM property	Description/notes
Job	Reference of EMC_VNXe_ConcreteJobLeaf
JobParameters	Reference of EMC_VNXe_MethodResultLeaf which is created from Job Context of the job

EMC_VNXe_StorageConfigurationService_ConcreteJob_OwningJobElementAssocLeaf

**Table 119 - Referenced properties/methods for
EMC_VNXe_StorageConfigurationService_ConcreteJob_OwningJobElementAssocLeaf**

CIM property	Description/notes
OwningElement	Reference of EMC_VNXe_StorageConfigurationServiceLeaf that creates the job.
OwnedElement	Reference of EMC_VNXe_ConcreteJobLeaf which is created by the service.

EMC_VNXe_ReplicationService_ConcreteJob_OwningJobElementAssocLeaf

**Table 120 - Referenced properties/methods for
EMC_VNXe_ReplicationService_ConcreteJob_OwningJobElementAssocLeaf**

CIM property	Description/notes
OwningElement	Reference of EMC_VNXe_ReplicationServiceLeaf that creates the job.
OwnedElement	Reference of EMC_VNXe_ConcreteJobLeaf which is created by the service.

EMC_VNXe_FileServerConfigurationService_ConcreteJob_OwningJobElementAssocLeaf

**Table 121 - Referenced properties/methods for
EMC_VNXe_FileServerConfigurationService_ConcreteJob_OwningJobElementAssocLeaf**

CIM property	Description/notes
OwningElement	Reference of EMC_VNXe_FileServerConfigurationServiceLeaf that creates the job.
OwnedElement	Reference of EMC_VNXe_ConcreteJobLeaf which is created by the service.

EMC_VNXe_FileSystemConfigurationService_ConcreteJob_OwningJobElementAssocLeaf

**Table 122 - Referenced properties/methods for
EMC_VNXe_FileSystemConfigurationService_ConcreteJob_OwningJobElementAssocLeaf**

CIM property	Description/notes
OwningElement	Reference of EMC_VNXe_FileSystemConfigurationServiceLeaf that creates the job.
OwnedElement	Reference of EMC_VNXe_ConcreteJobLeaf which is created by the service.

EMC_VNXe_FileExportService_ConcreteJob_OwningJobElementAssocLeaf

**Table 123 - Referenced properties/methods for
EMC_VNXe_FileExportService_ConcreteJob_OwningJobElementAssocLeaf**

CIM property	Description/notes
OwningElement	Reference of EMC_VNXe_FileExportServiceLeaf that creates the job.
OwnedElement	Reference of EMC_VNXe_ConcreteJobLeaf which is created by the service.

EMC_VNXe_MappedStoragePool_ConcreteJob_AffectedJobElementAssocLeaf

**Table 124 - Referenced properties/methods for
EMC_VNXe_MappedStoragePool_ConcreteJob_AffectedJobElementAssocLeaf**

CIM property	Description/notes
AffectingElement	Reference of EMC_VNXe_ConcreteJobLeaf
AffectedElement	Reference of EMC_VNXe_MappedStoragePoolLeaf which is affected by the job.

EMC_VNXe_StorageVolume_ConcreteJob_AffectedJobElementAssocLeaf

**Table 125 - Referenced properties/methods for
EMC_VNXe_StorageVolume_ConcreteJob_AffectedJobElementAssocLeaf**

CIM property	Description/notes
AffectingElement	Reference of EMC_VNXe_ConcreteJobLeaf
AffectedElement	Reference of EMC_VNXe_StorageVolumeLeaf which is affected by the job.

EMC_VNXe_SnapVolume_ConcreteJob_AffectedJobElementAssocLeaf

**Table 126 - Referenced properties/methods for
EMC_VNXe_SnapVolume_ConcreteJob_AffectedJobElementAssocLeaf**

CIM property	Description/notes
AffectingElement	Reference of EMC_VNXe_ConcreteJobLeaf
AffectedElement	Reference of EMC_VNXe_SnapVolumeLeaf which is affected by the job.

EMC_VNXe_SnapGroup_ConcreteJob_AffectedJobElementAssocLeaf

**Table 127 - Referenced properties/methods for
EMC_VNXe_SnapGroup_ConcreteJob_AffectedJobElementAssocLeaf**

CIM property	Description/notes
AffectingElement	Reference of EMC_VNXe_ConcreteJobLeaf
AffectededElement	Reference of EMC_VNXe_SnapGroupLeaf which is affected by the job.

EMC_VNXe_CIFSServer_ConcreteJob_AffectedJobElementAssocLeaf

**Table 128 - Referenced properties/methods for
EMC_VNXe_CIFSServer_ConcreteJob_AffectedJobElementAssocLeaf**

CIM property	Description/notes
AffectingElement	Reference of EMC_VNXe_ConcreteJobLeaf
AffectededElement	Reference of EMC_VNXe_CIFSServerLeaf which is affected by the job.

EMC_VNXe_NFSServer_ConcreteJob_AffectedJobElementAssocLeaf

**Table 129 - Referenced properties/methods for
EMC_VNXe_NFSServer_ConcreteJob_AffectedJobElementAssocLeaf**

CIM property	Description/notes
AffectingElement	Reference of EMC_VNXe_ConcreteJobLeaf
AffectededElement	Reference of EMC_VNXe_NFSServerLeaf which is affected by the job.

EMC_VNXe_CIFSShare_ConcreteJob_AffectedJobElementAssocLeaf

**Table 130 - Referenced properties/methods for
EMC_VNXe_CIFSShare_ConcreteJob_AffectedJobElementAssocLeaf**

CIM property	Description/notes
AffectingElement	Reference of EMC_VNXe_ConcreteJobLeaf
AffectededElement	Reference of EMC_VNXe_CIFSShareLeaf which is affected by the job.

EMC_VNXe_NFSShare_ConcreteJob_AffectedJobElementAssocLeaf

**Table 131 - Referenced properties/methods for
EMC_VNXe_NFSShare_ConcreteJob_AffectedJobElementAssocLeaf**

CIM property	Description/notes
AffectingElement	Reference of EMC_VNXe_ConcreteJobLeaf
AffectededElement	Reference of EMC_VNXe_NFSShareLeaf which is affected by the job.

EMC_VNXe_UxfsLocalFileSystem_ConcreteJob_AffectedJobElementAssocLeaf

**Table 132 - Referenced properties/methods for
EMC_VNXe_UxfsLocalFileSystem_ConcreteJob_AffectedJobElementAssocLeaf**

CIM property	Description/notes
AffectingElement	Reference of EMC_VNXe_ConcreteJobLeaf
AffectedElement	Reference of EMC_VNXe_UxfsLocalFileSystemLeaf which is affected by the job.

Array Profile

Overview

The Array Profile describes RAID array systems. The SMI-S Provider for the VNXe storage system implements the following component or subprofiles of the Array Profile:

- Multiple Computer System Subprofile
- Physical Package Package
- Health Package
- Indication Subprofile
- Job Control Subprofile
- Software Subprofile
- Access Points Subprofile
- FC Target Ports Subprofile
- iSCSI Target Ports Subprofile
- Block Service Package
- Disk Drive Lite Subprofile
- Extent Composition Subprofile
- Masking and Mapping Subprofile
- Copy Service Subprofile
- Replication Service Profile
- Thin Provisioning Profile

NOTE: For more details, refer to [Clause 4: Array Profile in Storage Management Technical Specification, Part 3 Block Devices, Version 1.5.0, Revision 6](#).

Class diagram

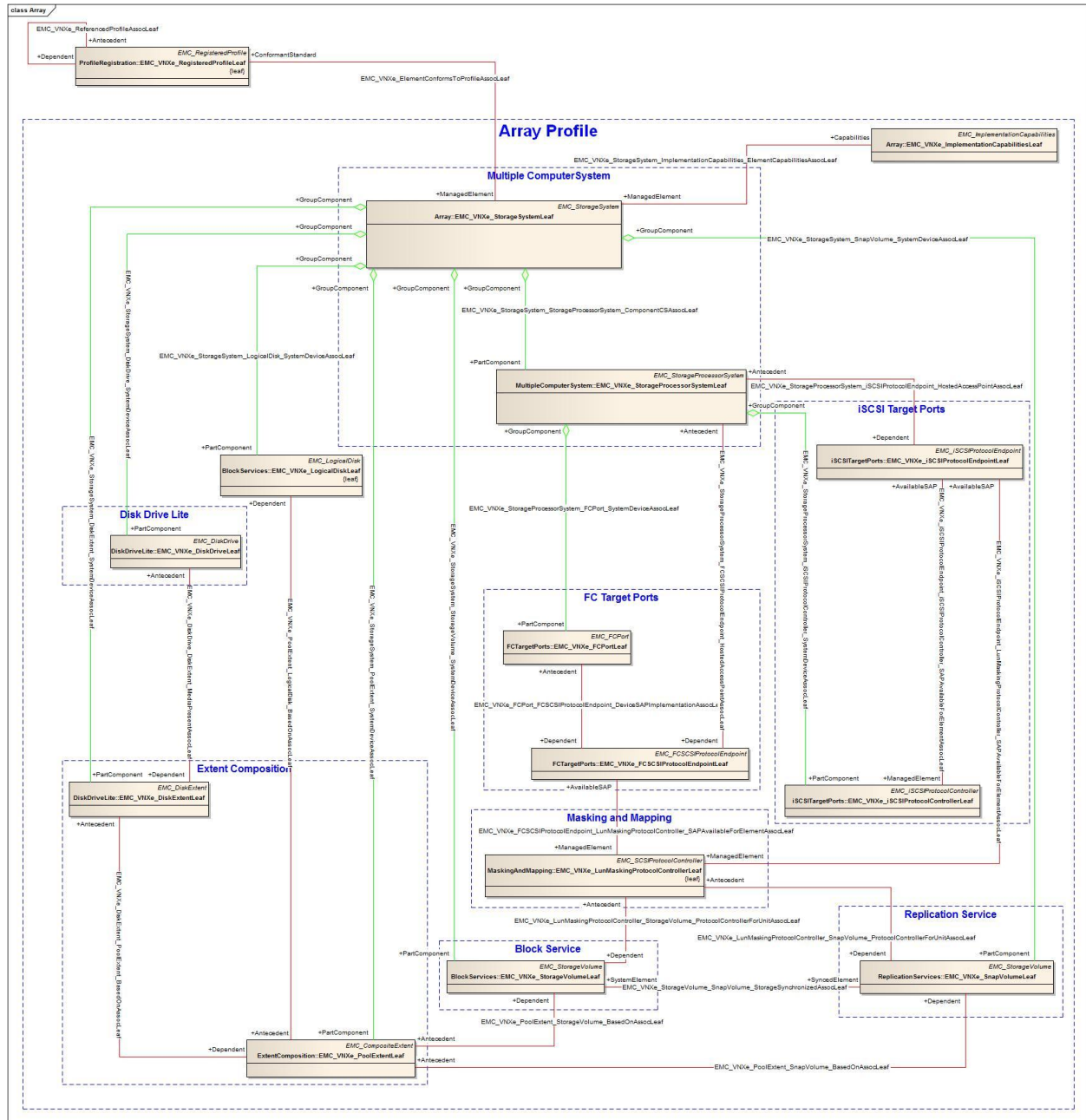


Figure 19- Array Profile class diagram

Method of the Profile

This profile does not include any extrinsic methods.

Client considerations

Use case: Discover Block Server (Array ComputerSystem)

This use case describes how to discover the Block Server (Array ComputerSystem):

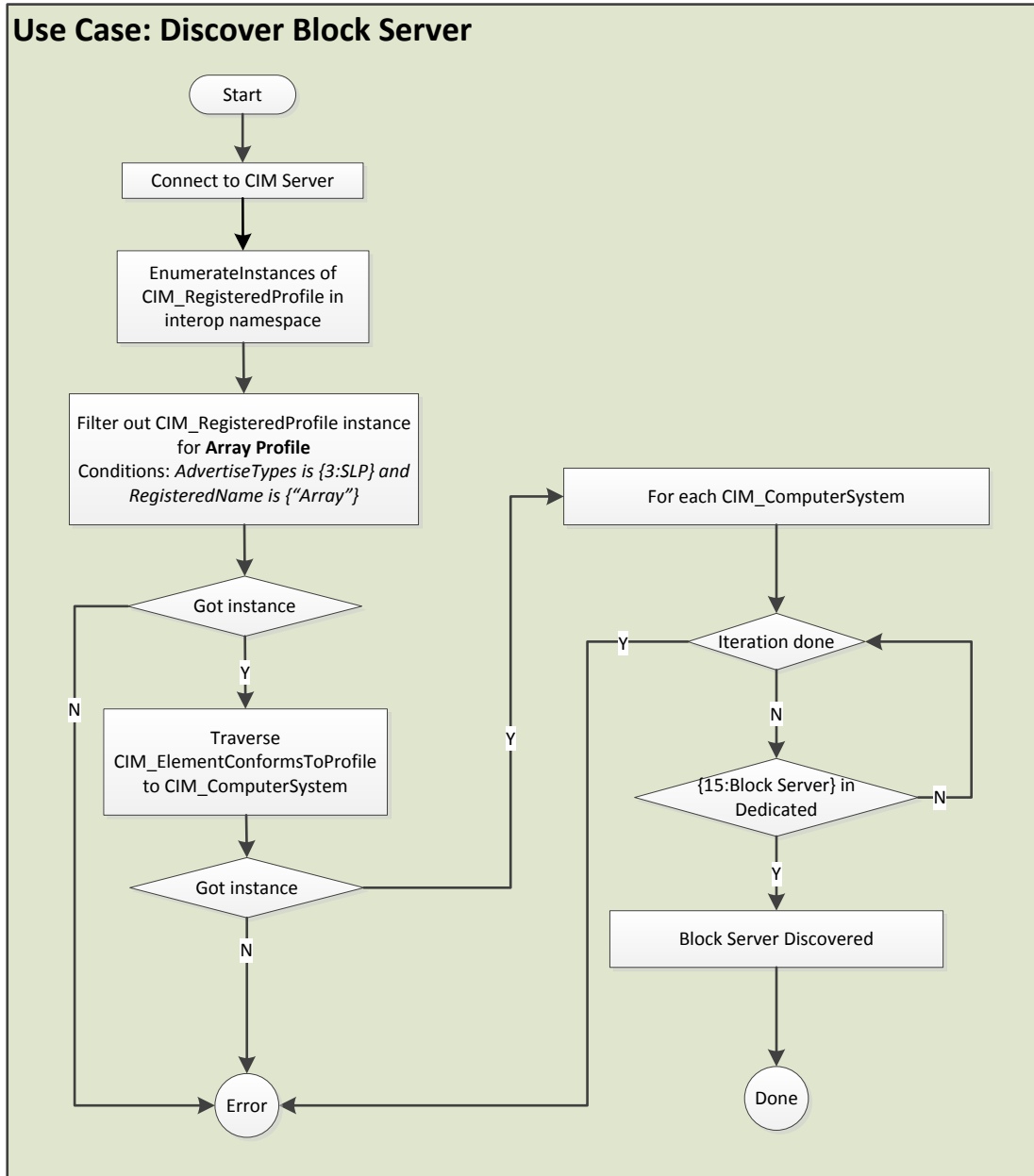


Figure 20 - Flowchart of Block Server discovery

Use case: Discover the capacity optimization support in an array

This use case describes how to discover the capacity optimization support in an array.

From the Block Server (Array ComputerSystem), traverse *CIM_ElementCapabilities* to *CIM_ImplementationCapabilities* to get the capabilities of the Array Profile implementation.

The capacity optimization supported by the Array is defined in the *SupportedCapacityOptimizations* property.

CIM Element

The implemented classes and associations related to Array Profile on the VNXe storage system are described as follows:

Table 133 - CIM Elements for Array Profile

CIM Class	Implemented Class	Description
CIM_ComputerSystem	EMC_VNXe_StorageSystemLeaf	Block Server. Top-level ComputerSystem of Array Profile.
CIM_ElementCapabilities	EMC_VNXe_SorageSystem_ImplementationCapabilities_ElementCapabilitiesAssocLeaf	Associates the Block Server and its ImplementationCapabilities.
CIM_ImplementationCapabilities	EMC_VNXe_ImplementationCapabilitiesLeaf	The capabilities of the profile implementation.

EMC_VNXe_StorageSystemLeaf

For further information, refer to [EMC_VNXe_StorageSystemLeaf](#) in Multiple Computer System Subprofile.

EMC_VNXe_ImplementationCapabilitiesLeaf

Table 134 - Referenced properties/methods for EMC_VNXe_ImplementationCapabilitiesLeaf

CIM property	Description/notes
InstanceID	Set as system name
ElementName	Set to Implementation Capabilities
SupportedCapacityOptimizations	Set to [SNIA:Thin Provisioning]
SupportedViews	Set to []

EMC_VNXe_SorageSystem_ImplementationCapabilities_ElementCapabilitiesAssocLeaf

Table 135 - Referenced properties/methods for EMC_VNXe_SorageSystem_ImplementationCapabilities_ElementCapabilitiesAssocLeaf

CIM property	Description/notes
ManagedElement	Reference of EMC_VNXe_StorageSystemLeaf
Capabilities	Reference of EMC_VNXe_ImplementationCapabilitiesLeaf

Client considerations

Model Specification

- Disk Drive Information

Disk Model: Implemented in *EMC_VNXe_DiskModuleLeaf.Model*

Manufacture: Implemented in *EMC_VNXe_DiskModuleLeaf.Manufacturer*

Part Number: Implemented in *EMC_VNXe_DiskModuleLeaf.PartNumber*

Serial Number: Implemented in *EMC_VNXe_DiskModuleLeaf.SerialNumber*

Firmware Version: Implemented in *EMC_VNXe_DiskSoftwareIdentityLeaf.VersionString*

Capacity: Calculated by *EMC_VNXe_DiskExtentLeaf.BlockSize * ConsumableBlocks*

CIM Element

The implemented classes and associations related to Disk Drive Lite Subprofile on the VNXe Storage System are described as follows:

Table 136 - CIM Elements for Disk Drive Lite Subprofile

CIM Class	Implemented Class	Description
CIM_AssociatedComponentExtent (Pool Component to Primordial Pool)	EMC_VNXe_PrimordialPool_DiskExtent_AssociatedComponentExtentAssocLeaf	Associates the PrimordialPool and DiskExtent.
CIM_BasedOn	EMC_VNXe_DiskExtent_DiskRemainingExtent_BasedOnAssocLeaf	Associates a concrete StorageExtent (DiskRemainingExtent) to the disk drive StorageExtent (DiskExtent) that it is allocated from.
CIM_BasedOn	EMC_VNXe_DiskExtent_PoolExtent_BasedOnAssocLeaf	Associates a concrete StorageExtent (PoolExtent) to the disk drive StorageExtent (DiskExtent) that it is allocated from.
CIM_ConcreteComponent (Disk Extent to Primordial Pool)	EMC_VNXe_PrimordialPool_DiskExtent_ConcreteComponentAssocLeaf	Associates the PrimordialPool and DiskExtent.
CIM_Container	EMC_VNXe_DAEEnclosureChassis_DiskModule_ContainerAssocLeaf	Associates a disk drive physical package (DiskModule) to its higher level package.
CIM_Container	EMC_VNXe_DPEEnclosureChassis_DiskModule_ContainerAssocLeaf	Associates a disk drive physical package (DiskModule) to its higher level package.
CIM_DiskDrive	EMC_VNXe_DiskDriveLeaf	Represents the disk drive.
CIM_ElementSoftwareIdentity	EMC_VNXe_DiskSoftwareIdentity_DiskDrive_ElementSoftwareIdentityAssocLeaf	Associates the firmware (SoftwareIdentity) to a disk drive.
CIM_MediaPresent	EMC_VNXe_DiskDrive_DiskExtent	Associates a disk drive to its storage extent.

CIM Class	Implemented Class	Description
	nt_MediaPresentAssocLeaf	
CIM_PhysicalPackage	EMC_VNXe_DiskModuleLeaf	The physical package for the disk drive.
CIM_Realizes	EMC_VNXe_DiskModule_DiskDrive_RealizesAssocLeaf	Associates the disk drive to its physical package.
CIM_SoftwareIdentity	EMC_VNXe_DiskSoftwareIdentityLeaf	Represents the firmware information for the disk drive.
CIM_StorageExtent	EMC_VNXe_DiskExtentLeaf	The storage extent that represents the storage of the disk drive.
CIM_StorageExtent	EMC_VNXe_DiskRemainingExtentLeaf	The storage extent that represents the remaining storage of the disk drive.
CIM_SystemDevice (Disk Drive System)	EMC_VNXe_StorageSystem_DiskDrive_SystemDeviceAssocLeaf	Associates DiskDrive to its hosting computer system.
CIM_SystemDevice (Storage Extent System)	EMC_VNXe_StorageSystem_DiskExtent_SystemDeviceAssocLeaf	Associates a DiskExtent to its hosting computer system.

EMC_VNXe_DiskDriveLeaf

Table 137 - Referenced properties/methods for EMC_VNXe_DiskDriveLeaf

CIM property	Description/notes
SystemCreationClassName	Set to "EMC_VNXe_StorageSystemLeaf"
SystemName	Set as system name of VNXe
CreationClassName	Set to "EMC_VNXe_DiskDriveLeaf"
DeviceID	Friendly name of the disk drive
Name	Friendly name of the disk drive
OperationalStatus	Current operation status of the disk drive
EnabledState	Set to 5: Not Applicable
RequestedState	Set to 12: Not Applicable
TransitioningToState	Set to 12: Not Applicable
Availability	Set to 2: Unknown
Capabilities	Set to [0: Unknown]
DefaultBlockSize	Set to 512

CIM property	Description/notes
HealthState	Current health state of the disk drive
MaxBlockSize	Set to 512
InterconnectType	Drive interface type

EMC_VNXe_DiskExtentLeaf

Table 138 - Referenced properties/methods for EMC_VNXe_DiskExtentLeaf

CIM property	Description/notes
SystemCreationClassName	Set to "EMC_VNXe_StorageSystemLeaf"
SystemName	Set as system name of VNXe
CreationClassName	Set to EMC_VNXe_DiskExtentLeaf
DeviceID	Friendly name of the disk drive.
ElementName	Friendly name of the extent.
Name	Set to Disk Extent
NameFormat	Set to 7: SNVM
NameNamespace	Set to 7: SNVM
OperationalStatus	Current operation status of the extent.
BlockSize	Set to 512
NumberOfBlocks	Block numbers of the extent.
ConsumableBlocks	Consumable block numbers of the extent.
Primordial	Set to TRUE
Access	Set to 3: Read/Write Supported'
ExtentStatus	Set to [2: None/Not Applicable]
ExtentDiscriminator	Set to [SNIA:Pool Component' , SNIA:Disk Drive]
HealthState	Current health state of the extent.
NoSinglePointOfFailure	Set to FALSE
IsComposite	Set to FALSE
Usage	Set to 2: Unrestricted
EnabledState	Set to 5: Not Applicable

CIM property	Description/notes
RequestedState	Set to 12: Not Applicable
TransitioningToState	Set to 12: Not Applicable

EMC_VNXe_DiskModuleLeaf

For further information, refer to [EMC_VNXe_DiskModuleLeaf](#) in Physical Package Package.

EMC_VNXe_DiskSoftwareIdentityLeaf

Table 139 - Referenced properties/methods for EMC_VNXe_DiskSoftwareIdentityLeaf

CIM property	Description/notes
InstanceID	Set as friendly name of the disk drive
IsEntity	Set to FALSE
IsLargeBuildNumber	Set to FALSE
VersionString	Firmware version of the disk drive

EMC_VNXe_DiskDrive_DiskExtent_MediaPresentAssocLeaf

Table 140 - Referenced properties/methods for EMC_VNXe_DiskDrive_DiskExtent_MediaPresentAssocLeaf

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_DiskExtentLeaf
Dependent	Reference of EMC_VNXe_DiskDriveLeaf

EMC_VNXe_DiskSoftwareIdentity_DiskDrive_ElementSoftwareIdentityAssocLeaf

Table 141 - Referenced properties/methods for EMC_VNXe_DiskSoftwareIdentity_DiskDrive_ElementSoftwareIdentityAssocLeaf

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_DiskSoftwareIdentityLeaf
Dependent	Reference of EMC_VNXe_DiskDriveLeaf

EMC_VNXe_StorageSystem_DiskDrive_SystemDeviceAssocLeaf

Table 142 - Referenced properties/methods for EMC_VNXe_StorageSystem_DiskDrive_SystemDeviceAssocLeaf

CIM property	Description/notes
PartComponent	Reference of EMC_VNXe_DiskDriveLeaf

CIM property	Description/notes
GroupComponent	Reference of EMC_VNXe_StorageSystemLeaf

EMC_VNXe_StorageSystem_DiskExtent_SystemDeviceAssocLeaf

**Table 143 - Referenced properties/methods for
EMC_VNXe_StorageSystem_DiskExtent_SystemDeviceAssocLeaf**

CIM property	Description/notes
PartComponent	Reference of EMC_VNXe_DiskExtentLeaf
GroupComponent	Reference of EMC_VNXe_StorageSystemLeaf

EMC_VNXe_DiskModule_DiskDrive_RealizesAssocLeaf

**Table 144 - Referenced properties/methods for
EMC_VNXe_DiskModule_DiskDrive_RealizesAssocLeaf**

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_DiskModuleLeaf
Dependent	Reference of EMC_VNXe_DiskDriveLeaf

EMC_VNXe_PrimordialPool_DiskExtent_AssociatedComponentExtentAssocLeaf

**Table 145 - Referenced properties/methods for
EMC_VNXe_PrimordialPool_DiskExtent_AssociatedComponentExtentAssocLeaf**

CIM property	Description/notes
PartComponent	Reference of EMC_VNXe_DiskExtentLeaf
GroupComponent	Reference of EMC_VNXe_PrimordialPoolLeaf

EMC_VNXe_PrimordialPool_DiskExtent_ConcreteComponentAssocLeaf

**Table 146 - Referenced properties/methods for
EMC_VNXe_PrimordialPool_DiskExtent_ConcreteComponentAssocLeaf**

CIM property	Description/notes
PartComponent	Reference of EMC_VNXe_DiskExtentLeaf
GroupComponent	Reference of EMC_VNXe_PrimordialPoolLeaf

EMC_VNXe_DiskExtent_DiskRemainingExtent_BasedOnAssocLeaf

**Table 147 - Referenced properties/methods for
EMC_VNXe_DiskExtent_DiskRemainingExtent_BasedOnAssocLeaf**

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_DiskExtentLeaf
Dependent	Reference of EMC_VNXe_DiskRemainingExtentLeaf

[EMC_VNXe_DiskExtent_PoolExtent_BasedOnAssocLeaf](#)

**Table 148 - Referenced properties/methods for
EMC_VNXe_DiskExtent_PoolExtent_BasedOnAssocLeaf**

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_DiskExtentLeaf
Dependent	Reference of EMC_VNXe_PoolExtentLeaf

[EMC_VNXe_DAEEnclosureChassis_DiskModule_ContainerAssocLeaf](#)

For further information, refer to [EMC_VNXe_DAEEnclosureChassis_DiskModule_ContainerAssocLeaf](#) in Physical Package Package.

[EMC_VNXe_DPEEnclosureChassis_DiskModule_ContainerAssocLeaf](#)

For further information, refer to [EMC_VNXe_DPEEnclosureChassis_DiskModule_ContainerAssocLeaf](#) in Physical Package Package.

Extent Composition Subprofile

Overview

The Extent Composition Subprofile allows an implementation that supports the Block Services package to optionally provide an abstraction of how it virtualizes exposable block storage elements from the underlying Primordial Storage Pool. The abstraction is presented to the client as a representative hierarchy of extents. These extents are instances of CompositeExtents and StorageExtents linked by a combination of CompositeExtentBasedOn and BasedOn associations. The foundation of the hierarchy is a set of Primordial Extents.

This subprofile is used optionally with the Array, Virtualization, Self-Contained NAS, NAS Head, and Volume Management profiles. A Primordial storage extent can represent a Disk Drive in the Array or Self-contained NAS, a downstream virtualized Volume used by the Virtualizer or NAS Head Profiles, or an OS Logical Disk in the Volume Management Profile. An exposable block used storage element in this subprofile is defined as a Storage Volume or a Logical Disk.

In the presented hierarchy each dependent extent is formed by those that precede the antecedents in a process of either decomposition or composition.

- **Decomposition**

Decomposition is used to allocate space from an antecedent extent, in order to form a new dependent extent. This allocation may be partial or complete consumption. Complete consumption is the degenerate case in which all space in the antecedent extent is used. In this case the decomposed dependent extent may be either modeled even though it is one to one with the antecedent extent or omitted and the antecedent extent used in its stead.

- **Composition**

Composition is used to form a dependent extent from antecedent extents in the purpose of either concatenating the antecedent blocks to achieve a size goal, or achieving a Quality of Service goal such as mirroring the antecedent extents for redundancy, striping the antecedent extents for performance, or striping the antecedent extents with the addition of parity to achieve redundancy.

These extent “productions” can be assembled in a multi-layer hierarchy.

NOTE: For more details, refer to [Clause 14: Extent Composition Subprofile in Storage Management Technical Specification, Part 3 Block Devices, Version 1.6.0, Revision 4](#).

Class diagram

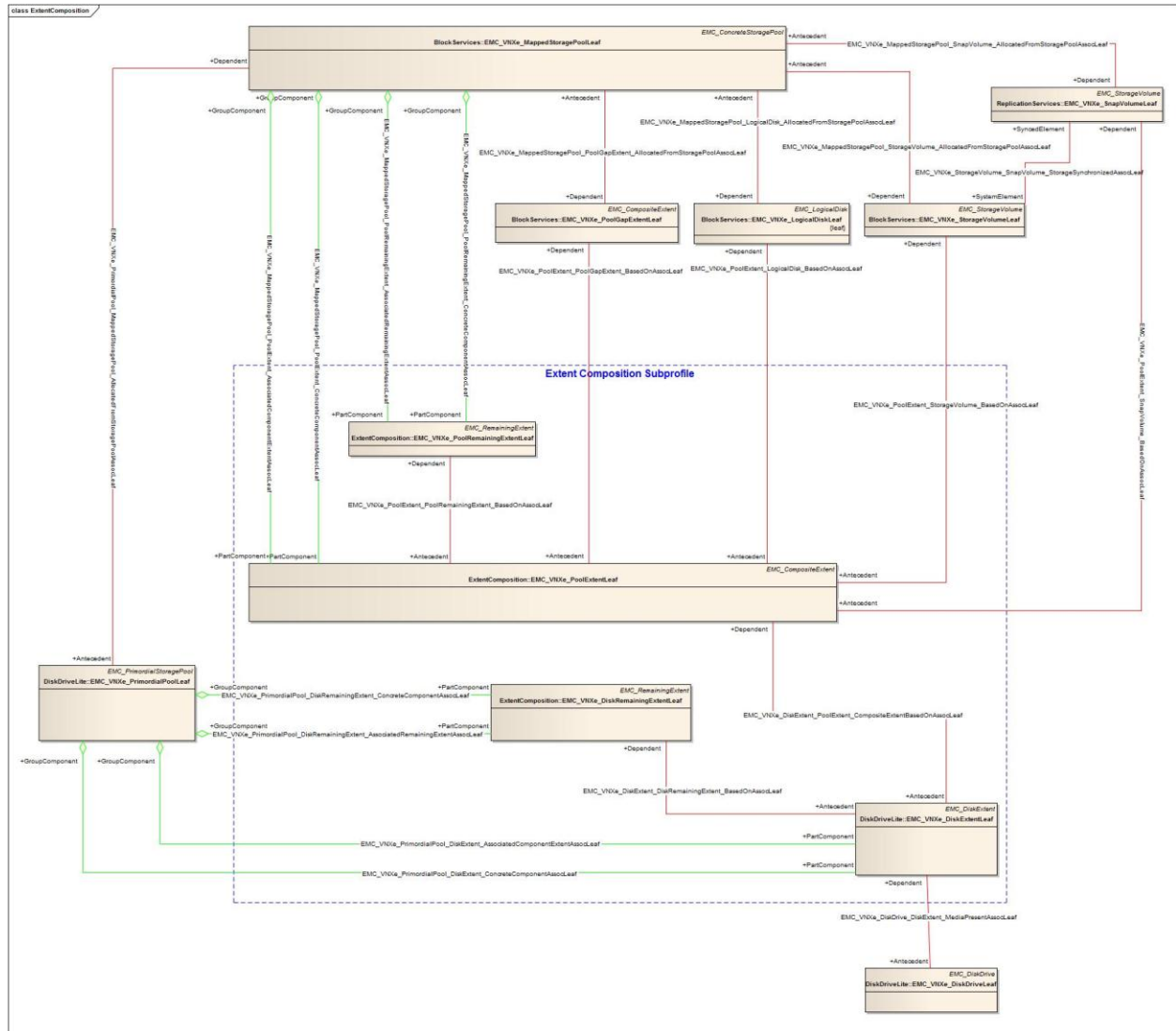


Figure 22 - Extent Composition Subprofile class diagram

Method of the Profile

This profile does not include any extrinsic methods.

Client considerations

Use case: Traverse the virtualization hierarchy of a StorageVolume or LogicalDisk

For more information, refer to [Use Case: Find the Primordial Extents used by a StorageVolume or LogicalDisk](#)

Use case: Find the Primordial Extents used by a StorageVolume or LogicalDisk

This use case describes how to get Primordial Extents used by a StorageVolume or LogicalDisk.

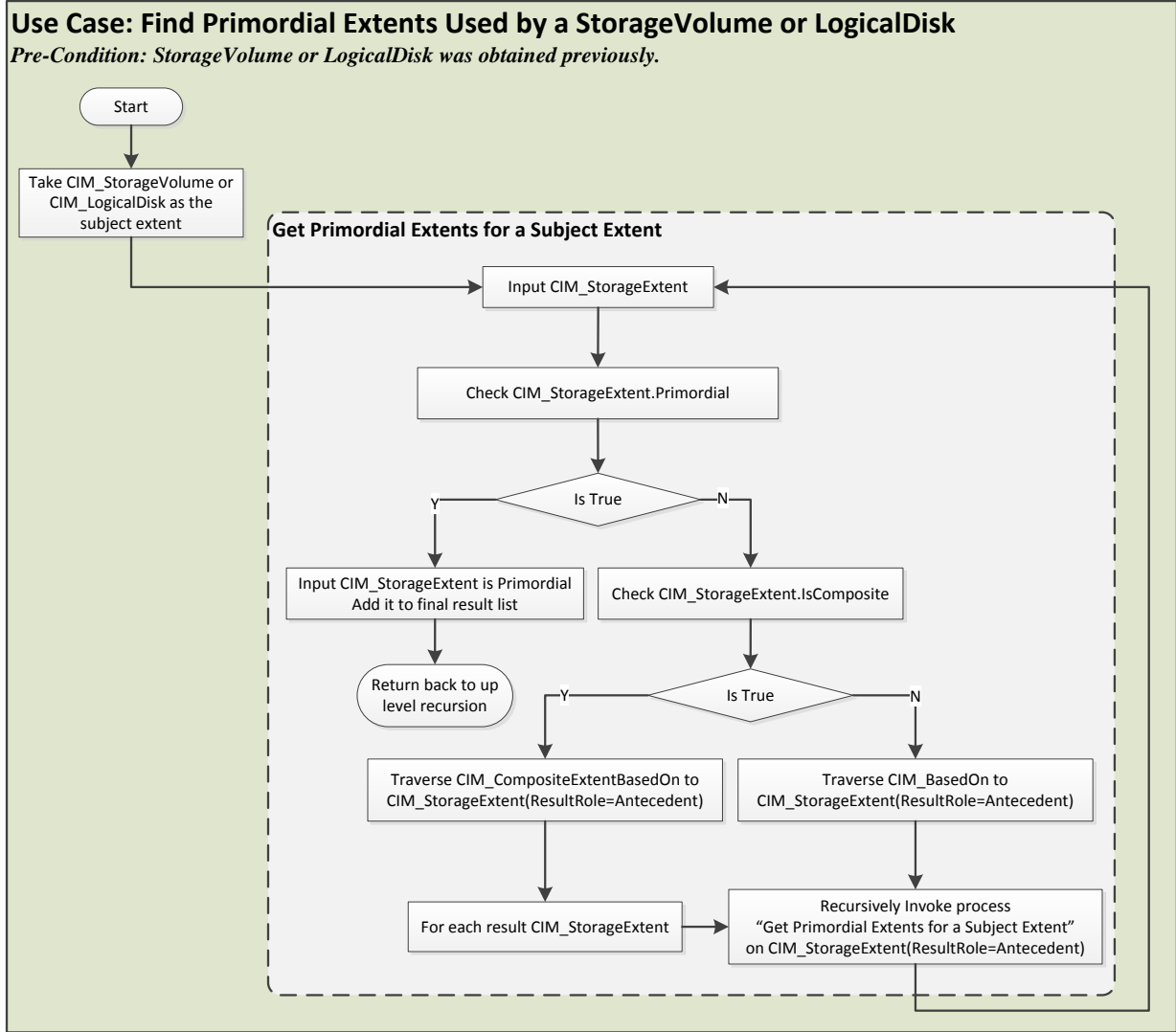


Figure 23 - Flowchart of finding Primordial Extents used by a StorageVolume or LogicalDisk

CIM Element

The implemented classes and associations related to Extent Composition Subprofile on the VNXe storage system are described as follows:

Table 149 - CIM Elements for Extent Composition Subprofile

CIM Class	Implemented Class	Description
CIM_AssociatedComponentExtent	EMC_VNXe_MappedStoragePool_PoolExtent_AssociatedComponentExtentAssocLeaf	Associates PoolExtent (Pool Component) to MappedStoragePool (Concrete Pool).
CIM_AssociatedComponentExtent	EMC_VNXe_PrimordialPool_DiskExtent_AssociatedComponentExtentAssocLeaf	Associates DiskExtent (Primordial) to PrimordialPool (Primordial Pool).
CIM_AssociatedRemainingExtent	EMC_VNXe_MappedStoragePool_ExtentAssocLeaf	Associates PoolRemainingExtent to MappedStoragePool (Concrete Pool).

CIM Class	Implemented Class	Description
	ol_PoolRemainingExtent_AssociatedRemainingExtentAssocLeaf	MappedStoragePool.
CIM_AssociatedRemainingExtent	EMC_VNXe_PrimordialPool_DiskRemainingExtent_AssociatedRemainingExtentAssocLeaf	Associates DiskRemainingExtent to PrimordialPool.
CIM_BasedOn	EMC_VNXe_DiskExtent_DiskRemainingExtent_BasedOnAssocLeaf	Associates DiskRemainingExtent to DiskExtent.
CIM_BasedOn	EMC_VNXe_PoolExtent_LogicalDisk_BasedOnAssocLeaf	Associates LogicalDisk to PoolExtent.
CIM_BasedOn	EMC_VNXe_PoolExtent_PoolGapExtent_BasedOnAssocLeaf	Associates PoolGapExtent to PoolExtent.
CIM_BasedOn	EMC_VNXe_PoolExtent_PoolRemainingExtent_BasedOnAssocLeaf	Associates PoolRemainingExtent to PoolExtent.
CIM_BasedOn	EMC_VNXe_PoolExtent_SnapVolume_BasedOnAssocLeaf	Associates SnapVolume to PoolExtent.
CIM_BasedOn	EMC_VNXe_PoolExtent_StorageVolume_BasedOnAssocLeaf	Associates StorageVolume to PoolExtent.
CIM_CompositeExtentBasedOn	EMC_VNXe_DiskExtent_PoolExtent_CompositeExtentBasedOnAssocLeaf	Associates PoolExtent to DiskExtent.
CIM_ConcreteComponent	EMC_VNXe_MappedStoragePool_PoolExtent_ConcreteComponentAssocLeaf	Associates PoolExtent (Pool Component) to MappedStoragePool (Concrete Pool). Deprecated in SMI-S v1.6.0 r4
CIM_ConcreteComponent	EMC_VNXe_MappedStoragePool_PoolRemainingExtent_ConcreteComponentAssocLeaf	Associates DiskExtent (Primordial) to PrimordialPool (Primordial Pool). Deprecated in SMI-S v1.6.0 r4
CIM_ConcreteComponent	EMC_VNXe_PrimordialPool_DiskExtent_ConcreteComponentAssocLeaf	Associates PoolRemainingExtent to MappedStoragePool. Deprecated in SMI-S v1.6.0 r4
CIM_ConcreteComponent	EMC_VNXe_PrimordialPool_DiskRemainingExtent_ConcreteComponentAssocLeaf	Associates DiskRemainingExtent to PrimordialPool. Deprecated in SMI-S v1.6.0 r4
CIM_StorageExtent	EMC_VNXe_DiskExtentLeaf	Represents the Primordial Extent.
CIM_StorageExtent	EMC_VNXe_DiskRemainingExtentLeaf	Represents the remaining extent of DiskExtent (Primordial Extent).
CIM_StorageExtent	EMC_VNXe_PoolExtentLeaf	Represents the Pool Component Extent.

CIM Class	Implemented Class	Description
CIM_StorageExtent	EMC_VNXe_PoolRemainingExtentLeaf	Represents the remaining extent of PoolExtent (Pool Component).
CIM_SystemDevice	EMC_VNXe_StorageSystem_DiskExtent_SystemDeviceAssocLeaf	Associates DiskExtent to the hosting system.
CIM_SystemDevice	EMC_VNXe_StorageSystem_DiskRemainingExtent_SystemDeviceAssocLeaf	Associates DiskRemainingExtent to the hosting system.
CIM_SystemDevice	EMC_VNXe_StorageSystem_PoolExtent_SystemDeviceAssocLeaf	Associates PoolExtent to the hosting system.
CIM_SystemDevice	EMC_VNXe_StorageSystem_PoolRemainingExtent_SystemDeviceAssocLeaf	Associates PoolRemainingExtent to the hosting system.

EMC_VNXe_DiskExtentLeaf

For detail information, refer to [EMC_VNXe_DiskExtentLeaf](#) in Disk Drive Lite Subprofile.

EMC_VNXe_PoolExtentLeaf

Table 150 - Referenced properties/methods for EMC_VNXe_PoolExtentLeaf

CIM property	Description/notes
SystemCreationClassName	Set to "EMC_VNXe_StorageSystemLeaf"
SystemName	Set as system name of VNXe
CreationClassName	Set to EMC_VNXe_PoolExtentLeaf
DeviceID	Friendly name of the StoragePool to which the extent associates.
ElementName	Name of the StoragePool to which the extent associates.
Name	Set to Pool Extent
NameFormat	Set to 7: SNVM
NameNamespace	Set to 7: SNVM
OperationalStatus	Current operation status of the extent.
BlockSize	Set to 512
NumberOfBlocks	Block numbers of the pool extent.
ConsumableBlocks	Consumable block numbers of the extent.
Primordial	Set to FALSE

CIM property	Description/notes
Access	Set to 3: Read/Write Supported
ExtentStatus	Set to [2: None/Not Applicable]
ExtentDiscriminator	Set to [SNIA:Pool Component' , SNIA: Composite]
ExtentStripeLength	Number of StorageExtents forming the user data stripe.
DataRedundancy	Number of complete copies of data currently maintained.
PackageRedundancy	Number of physical packages can currently fail without data loss.
IsBasedOnUnderlyingRedundancy	Set to TRUE
IsConcatenated	Set to FALSE
HealthState	Current health state of the extent
NoSinglePointOfFailure	Set to FALSE
IsComposite	Set to FALSE
Usage	Set to 2: Unrestricted
EnabledState	Set to 5: Not Applicable
RequestedState	Set to 12: Not Applicable
TransitioningToState	Set to 12: Not Applicable

EMC_VNXe_DiskRemainingExtentLeaf

Table 151 - Referenced properties/methods for EMC_VNXe_DiskRemainingExtentLeaf

CIM property	Description/notes
SystemCreationClassName	Set to EMC_VNXe_StorageSystemLeaf
SystemName	Set as system name of VNXe
CreationClassName	Set to EMC_VNXe_DiskRemainingExtentLeaf
DeviceID	Friendly id of the DiskDrive to which the extent associates.
ElementName	Friendly name of the extent.
Name	Set to Disk Remaining Extent
NameFormat	Set to 7: SNVM
NameNamespace	Set to 7: SNVM
OperationalStatus	Current operation status of the extent.

CIM property	Description/notes
BlockSize	Set to 512
NumberOfBlocks	Block numbers of the extent.
ConsumableBlocks	Consumable block numbers of the extent.
Primordial	Set to FALSE
Access	Set to 3: Read/Write Supported
ExtentStatus	Set to [2: None/Not Applicable]
ExtentDiscriminator	Set to [SNIA: Remaining]
HealthState	Current health state of the disk extent
NoSinglePointOfFailure	Set to FALSE
IsComposite	Set to FALSE
Usage	Set to 2: Unrestricted
EnabledState	Set to 5: Not Applicable
RequestedState	Set to 12: Not Applicable
TransitioningToState	Set to 12: Not Applicable

EMC_VNXe_PoolRemainingExtentLeaf

Table 152 - Referenced properties/methods for EMC_VNXe_PoolRemainingExtentLeaf

CIM property	Description/notes
SystemCreationClassName	Set to EMC_VNXe_StorageSystemLeaf
SystemName	Set as system name of VNXe
CreationClassName	Set to EMC_VNXe_PoolRemainingExtentLeaf
DeviceID	Friendly name of the StoragePool to which the extent associates.
ElementName	Name of the StoragePool to which the extent associates.
Name	Set to Pool Remaining Extent
NameFormat	Set to 7: SNVM
NameNamespace	Set to 7: SNVM
OperationalStatus	Current operation status of the extent.
BlockSize	Set to 512

CIM property	Description/notes
NumberOfBlocks	Block numbers of the extent.
ConsumableBlocks	Consumable block numbers of the extent.
Primordial	Set to FALSE
Access	Set to 3: Read/Write Supported
ExtentStatus	Set to [2: None/Not Applicable]
ExtentDiscriminator	Set to [SNIA: Remaining, SNIA: Composite]
IsBasedOnUnderlyingRedundancy	Set to TRUE
HealthState	Current health state of the extent
NoSinglePointOfFailure	Set to FALSE
IsComposite	Set to FALSE
Usage	Set to 2: Unrestricted
EnabledState	Set to 5: Not Applicable
RequestedState	Set to 12: Not Applicable
TransitioningToState	Set to 12: Not Applicable

EMC_VNXe_StorageSystem_DiskExtent_SystemDeviceAssocLeaf

**Table 153 - Referenced properties/methods for
EMC_VNXe_StorageSystem_DiskExtent_SystemDeviceAssocLeaf**

CIM property	Description/notes
GroupComponent	Reference of EMC_VNXe_StorageSystemLeaf
PartComponent	Reference of EMC_VNXe_DiskExtentLeaf

EMC_VNXe_StorageSystem_PoolExtent_SystemDeviceAssocLeaf

**Table 154 - Referenced properties/methods for
EMC_VNXe_StorageSystem_PoolExtent_SystemDeviceAssocLeaf**

CIM property	Description/notes
GroupComponent	Reference of EMC_VNXe_StorageSystemLeaf
PartComponent	Reference of EMC_VNXe_PoolExtentLeaf

EMC_VNXe_StorageSystem_DiskRemainingExtent_SystemDeviceAssocLeaf

**Table 155 - Referenced properties/methods for
EMC_VNXe_StorageSystem_DiskRemainingExtent_SystemDeviceAssocLeaf**

CIM property	Description/notes
GroupComponent	Reference of EMC_VNXe_StorageSystemLeaf
PartComponent	Reference of EMC_VNXe_DiskRemainingExtentLeaf

EMC_VNXe_StorageSystem_PoolRemainingExtent_SystemDeviceAssocLeaf

**Table 156 - Referenced properties/methods for
EMC_VNXe_StorageSystem_PoolRemainingExtent_SystemDeviceAssocLeaf**

CIM property	Description/notes
GroupComponent	Reference of EMC_VNXe_StorageSystemLeaf
PartComponent	Reference of EMC_VNXe_PoolRemainingExtentLeaf

EMC_VNXe_DiskExtent_PoolExtent_CompositeExtentBasedOnAssocLeaf

**Table 157 - Referenced properties/methods for
EMC_VNXe_DiskExtent_PoolExtent_CompositeExtentBasedOnAssocLeaf**

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_DiskExtentLeaf
Dependent	Reference of EMC_VNXe_PoolExtentLeaf

EMC_VNXe_DiskExtent_DiskRemainingExtent_BasedOnAssocLeaf

**Table 158 - Referenced properties/methods for
EMC_VNXe_DiskExtent_DiskRemainingExtent_BasedOnAssocLeaf**

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_DiskExtentLeaf
Dependent	Reference of EMC_VNXe_DiskRemainingExtentLeaf

EMC_VNXe_PoolExtent_PoolRemainingExtent_BasedOnAssocLeaf

**Table 159 - Referenced properties/methods for
EMC_VNXe_PoolExtent_PoolRemainingExtent_BasedOnAssocLeaf**

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_PoolExtentLeaf
Dependent	Reference of EMC_VNXe_PoolRemainingExtentLeaf

EMC_VNXe_PoolExtent_LogicalDisk_BasedOnAssocLeaf

**Table 160 - Referenced properties/methods for
EMC_VNXe_PoolExtent_LogicalDisk_BasedOnAssocLeaf**

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_PoolExtentLeaf
Dependent	Reference of EMC_VNXe_LogicalDiskLeaf

EMC_VNXe_PoolExtent_StorageVolume_BasedOnAssocLeaf

**Table 161 - Referenced properties/methods for
EMC_VNXe_PoolExtent_StorageVolume_BasedOnAssocLeaf**

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_PoolExtentLeaf
Dependent	Reference of EMC_VNXe_StorageVolumeLeaf

EMC_VNXe_PoolExtent_SnapVolume_BasedOnAssocLeaf

**Table 162 - Referenced properties/methods for
EMC_VNXe_PoolExtent_SnapVolume_BasedOnAssocLeaf**

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_PoolExtentLeaf
Dependent	Reference of EMC_VNXe_SnapVolumeLeaf

EMC_VNXe_PoolExtent_PoolGapExtent_BasedOnAssocLeaf

**Table 163 - Referenced properties/methods for
EMC_VNXe_PoolExtent_PoolGapExtent_BasedOnAssocLeaf**

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_PoolExtentLeaf
Dependent	Reference of EMC_VNXe_PoolGapExtentLeaf

EMC_VNXe_MappedStoragePool_PoolExtent_AssociatedComponentExtentAssocLeaf

**Table 164 - Referenced properties/methods for
EMC_VNXe_MappedStoragePool_PoolExtent_AssociatedComponentExtentAssocLeaf**

CIM property	Description/notes
GroupComponent	Reference of EMC_VNXe_MappedStoragePoolLeaf
PartComponent	Reference of EMC_VNXe_PoolExtentLeaf

EMC_VNXe_PrimordialPool_DiskExtent_AssociatedComponentExtentAssocLeaf

**Table 165 - Referenced properties/methods for
EMC_VNXe_PrimordialPool_DiskExtent_AssociatedComponentExtentAssocLeaf**

CIM property	Description/notes
GroupComponent	Reference of EMC_VNXe_PrimordialPoolLeaf
PartComponent	Reference of EMC_VNXe_DiskExtentLeaf

EMC_VNXe_MappedStoragePool_PoolRemainingExtent_AssociatedRemainingExtentAssocLeaf

**Table 166 - Referenced properties/methods for
EMC_VNXe_MappedStoragePool_PoolRemainingExtent_AssociatedRemainingExtentAssocLeaf**

CIM property	Description/notes
GroupComponent	Reference of EMC_VNXe_MappedStoragePoolLeaf
PartComponent	Reference of EMC_VNXe_PoolRemainingExtentLeaf

EMC_VNXe_PrimordialPool_DiskRemainingExtent_AssociatedRemainingExtentAssocLeaf

**Table 167 - Referenced properties/methods for
EMC_VNXe_PrimordialPool_DiskRemainingExtent_AssociatedRemainingExtentAssocLeaf**

CIM property	Description/notes
GroupComponent	Reference of EMC_VNXe_PrimordialPoolLeaf
PartComponent	Reference of EMC_VNXe_DiskRemainingExtentLeaf

EMC_VNXe_MappedStoragePool_PoolExtent_ConcreteComponentAssocLeaf

**Table 168 - Referenced properties/methods for
EMC_VNXe_MappedStoragePool_PoolExtent_ConcreteComponentAssocLeaf**

CIM property	Description/notes
GroupComponent	Reference of EMC_VNXe_MappedStoragePoolLeaf
PartComponent	Reference of EMC_VNXe_PoolExtentLeaf

EMC_VNXe_MappedStoragePool_PoolRemainingExtent_ConcreteComponentAssocLeaf

**Table 169 - Referenced properties/methods for
EMC_VNXe_MappedStoragePool_PoolRemainingExtent_ConcreteComponentAssocLeaf**

CIM property	Description/notes
GroupComponent	Reference of EMC_VNXe_MappedStoragePoolLeaf

CIM property	Description/notes
PartComponent	Reference of EMC_VNXe_PoolRemainingExtentLeaf

EMC_VNXe_PrimordialPool_DiskExtent_ConcreteComponentAssocLeaf

Table 170 - Referenced properties/methods for EMC_VNXe_PrimordialPool_DiskExtent_ConcreteComponentAssocLeaf

CIM property	Description/notes
GroupComponent	Reference of EMC_VNXe_PrimordialPoolLeaf
PartComponent	Reference of EMC_VNXe_DiskExtentLeaf

EMC_VNXe_PrimordialPool_DiskRemainingExtent_ConcreteComponentAssocLeaf

Table 171 - Referenced properties/methods for EMC_VNXe_PrimordialPool_DiskRemainingExtent_ConcreteComponentAssocLeaf

CIM property	Description/notes
GroupComponent	Reference of EMC_VNXe_PrimordialPoolLeaf
PartComponent	Reference of EMC_VNXe_DiskRemainingExtentLeaf

Table 172 - Extrinsic methods on StorageCapabilities

Method name	Primordial pool capabilities	Concrete pool capabilities	Storage system capabilities
CreateSetting	Not Supported	Supported	Not Supported
GetSupportedStripeLengths	Supported	Supported	Supported
GetSupportedStripeLengthRange	Use GetSupportedStripeLengths Instead	Use GetSupportedStripeLengths Instead	Use GetSupportedStripeLengths Instead
GetSupportedStripeDepths	Not Supported	Not Supported	Not Supported
GetSupportStripeDepthRange	Not Supported	Not Supported	Not Supported
GetSupportedParityLayouts	Supported	Supported	Supported

Method: EMC_VNXe_MappedStoragePoolCapabilitiesLeaf.GetSupportedStripeLengths

- Description**

This method is used to report discrete *ExtentStripeLengths* for *StorageVolume* creation from the Concrete *StoragePool*.

- Parameters**

Table 173 - Signature and parameters of EMC_VNXe_MappedStoragePoolCapabilitiesLeaf.GetSupportedStripeLength

Parameter name	Qualifier	Type	Description/notes
StripeLengths	OUT	uint16[]	List of supported <i>ExtentStripeLengths</i> for a <i>StorageVolume</i> creation or modification.

- Return Results**

Table 174 - Possible return code of EMC_VNXe_MappedStoragePoolCapabilitiesLeaf.GetSupportedStripeLengths

Return code	Type	Description
0	uint32	Method completed OK
3	uint32	Choices not available for this Capability

Method: EMC_VNXe_MappedStoragePoolCapabilitiesLeaf.GetSupportedParityLayouts

- Description**

This method is used to return the type of parity, Non-rotated or rotated, that the Concrete Pool's *StorageCapability* supports.

- Parameters**

**Table 175 - Signature and parameters of
EMC_VNXe_MappedStoragePoolCapabilitiesLeaf.GetSupportedParityLayouts**

Parameter name	Qualifier	Type	Description/notes
ParityLayout	OUT	uint16[]	List of supported Parity for a <i>StorageVolume</i> creation or modification.

- **Return Results**

**Table 176 - Possible return code of
EMC_VNXe_MappedStoragePoolCapabilitiesLeaf.GetSupportedParityLayouts**

Return code	Type	Description
0	uint32	Method completed OK
3	uint32	Choices not available for this Capability

Method: EMC_VNXe_MappedStorageStoragePoolCapabilitiesLeaf.CreateSetting

- **Description**

This method is used to create and populate a generated *StorageSetting* instance supported by the capabilities of concrete *StoragePool*.

NOTE: *StorageSetting* created from *Capabilities* cannot be modified.

- **Parameters**

**Table 177 - Signature and parameters of
EMC_VNXe_StorageConfigurationServiceCapabilitiesLeaf.CreateSetting**

Parameter name	Qualifier	Type	Description/notes
SettingType	IN, Ignored	uint16	Setting type specified by client.
NewSetting	OUT, NULL allowed	CIM_StorageSettingRef	<i>StorageSetting</i> reference created by the <i>StorageCapabilities</i> instance.

- **Return Results**

**Table 178 - Possible return code of
EMC_VNXe_StorageConfigurationServiceCapabilitiesLeaf.CreateSetting**

Return code	Type	Description
0	uint32	Method completed OK
4	uint32	Method failed. No <i>StorageSetting</i> was created.

Method: EMC_VNXe_PrimordialPoolCapabilitiesLeaf.GetSupportedStripeLengths

- **Description**

This method is used to report discrete *ExtentStripeLengths* for *StoragePool* creation from a Primordial *StoragePool*.

- **Parameters**

**Table 179 - Signature and parameters of
EMC_VNXe_PrimordialPoolCapabilitiesLeaf.GetSupportedStripeLengths**

Parameter name	Qualifier	Type	Description/notes
StripeLengths	OUT	uint16[]	List of supported ExtentStripeLengths for a <i>StoragePool</i> creation or modification.

- **Return Results**

**Table 180 - Possible return code of
EMC_VNXe_PrimordialPoolCapabilitiesLeaf.GetSupportedStripeLengths**

Return code	Type	Description
0	uint32	Method completed OK
3	uint32	Choices not available for this Capability

Method: EMC_VNXe_PrimordialPoolCapabilitiesLeaf.GetSupportedParityLayouts

- **Description**

This method is used to return the type of parity, non-rotated or rotated, that the Primordial Pool's *StorageCapability* supports.

- **Parameters**

**Table 181 - Signature and parameters of
EMC_VNXe_PrimordialPoolCapabilitiesLeaf.GetSupportedParityLayouts**

Parameter name	Qualifier	Type	Description/notes
ParityLayout	OUT	uint16[]	List of supported Parity for a <i>StoragePool</i> creation or modification.

- **Return Results**

**Table 182 - Possible return code of
EMC_VNXe_PrimordialPoolCapabilitiesLeaf.GetSupportedParityLayouts**

Return code	Type	Description
0	uint32	Method completed OK
3	uint32	Choices not available for this Capability

Method: EMC_VNXe_StorageConfigurationServiceCapabilitiesLeaf.GetSupportedStripeLengths

- **Description**

This method is used to report discrete *ExtentStripeLengths* for *StoragePool* or *StorageVolume* creation on the storage system.

- **Parameters**

Table 183 - Signature and parameters of EMC_VNXe_StorageConfigurationServiceCapabilitiesLeaf.GetSupportedStripeLengths

Parameter name	Qualifier	Type	Description/notes
StripeLengths	OUT	uint16[]	List of supported <i>ExtentStripeLengths</i> for a <i>StoragePool</i> or <i>StorageVolume</i> creation or modification.

- **Return Results**

Table 184 - Possible return code of EMC_VNXe_StorageConfigurationServiceCapabilitiesLeaf.GetSupportedStripeLengths

Return code	Type	Description
0	uint32	Method completed OK
3	uint32	Choices not available for this Capability

Method: EMC_VNXe_StorageConfigurationServiceCapabilitiesLeaf.GetSupportedParityLayouts

- **Description**

This method is used to return the type of parity, non-rotated or rotated, that the storage system's *StorageCapability* supports.

- **Parameters**

Table 185 - Signature and parameters of EMC_VNXe_StorageConfigurationServiceCapabilitiesLeaf.GetSupportedParityLayouts

Parameter name	Qualifier	Type	Description/notes
ParityLayout	OUT	uint16[]	List of supported Parity for a <i>StoragePool</i> or <i>StorageVolume</i> creation or modification.

- **Return results**

Table 186 - Possible return code of EMC_VNXe_StorageConfigurationServiceCapabilitiesLeaf.GetSupportedParityLayouts

Return code	Type	Description
0	uint32	Method completed OK
3	uint32	Choices not available for this Capability

Extrinsic methods on StoragePool

Table 187 - Extrinsic methods on StoragePool

Method name	Primordial pool	Concrete pool
GetSupportedSizes	Not Supported	Use GetSupportedSizeRange Instead
GetSupportedSizeRange	Not Supported	Supported
GetAvailableExtents	Supported	Supported

Method: EMC_VNXe_MappedStoragePoolLeaf.GetAvailableExtents

- Description**

This method is used to retrieve available StorageExtents of the concrete StoragePool.

- Parameters**

Table 188 - Signature and parameters of EMC_VNXe_MappedStoragePoolLeaf.GetAvailableExtents

Parameter name	Qualifier	Type	Description/notes
Goal	IN, Ignored	CIM_StorageSettingRef	Ignored by implementation which means all available extents will be returned.
AvailableExtents	OUT	CIM_StorageExtentRef []	Reference list of all available extents of the pool.

- Return Results**

Table 189 - Possible return code of EMC_VNXe_MappedStoragePoolLeaf.GetAvailableExtents

Return code	Type	Description
0	uint32	Method completed OK.
4	uint32	Method failed.

Method: EMC_VNXe_MappedStoragePoolLeaf.GetSupportedSizeRange

- Description**

This method is used to report the possible Size range for StorageVolume creation.

- Parameters**

Table 190 – Signature and parameters of EMC_VNXe_MappedStoragePoolLeaf.GetSupportedSizeRange

Parameter name	Qualifier	Type	Description/notes
ElementType	IN,	uint16	Type for element to create.

Parameter name	Qualifier	Type	Description/notes
	NULL allowed		NOTE: Only 2: Storage_Volume and 5: Thin Provisioned Volume' are allowed.
Goal	IN, Ignored	CIM_StorageSettingRef	Goal storage setting which specified raid-level and stripe length.
MinimumVolumeSize	IN, OUT, NULL allowed	uint64	The minimum size for a volume in bytes.
MaximumVolumeSize	IN, OUT, NULL allowed	uint64	The maximum size for a volume in bytes.
VolumeSizeDivisor	IN, OUT, NULL allowed	uint64	A volume size must be a multiple of this value which is specified in bytes.

- **Return code**

Table 191 - Possible return code of EMC_VNXe_MappedStoragePoolLeaf.GetSupportedSizeRange

Return code	Type	Description
0	uint32	Method completed OK
2	uint32	Method not supported by underlying component
3	uint32	The ElementType parameter is invalid.

Method: EMC_VNXe_PrimordialPoolLeaf.GetAvailableExtents

- **Description**

This method is used to retrieve available StorageExtents of the Primordial StoragePool.

- **Parameters**

Table 192 - Signature and parameters of EMC_VNXe_PrimordialPoolLeaf.GetAvailableExtents

Parameter Name	Qualifier	Type	Description/notes
Goal	IN, Ignored	CIM_StorageSettingRef	Ignored by implementation which means all available extents will be returned.
AvailableExtents	OUT,	CIM_StorageExtentRef []	Reference list of all available extents of the pool.

- **Return results**

Table 193 - Possible return code of EMC_VNXe_PrimordialPoolLeaf.GetAvailableExtents

Return code	Type	Description
0	uint32	Method completed OK
4	uint32	Method failed.

Extrinsic methods on StorageConfiguration

Table 194- Extrinsic methods on StorageConfiguration

Method name	StorageConfigurationService
CreateOrModifyStoragePool	Not Supported
DeleteStoragePool	Not Supported
CreateOrModifyElementFromStoragePool	Supported
CreateElementsFromStoragePool	Not Supported
CreateOrModifyElementFromElements	Not Supported
ReturnToStoragePool	Supported
ReturnElementsToStoragePool	Not Supported
RequestUsageChange	Not Supported
GetElementsBasedOnUsage	Not Supported
CreateReplica	Supported (Described in Copy Service Subprofile)
ModifySynchronization	Supported (Described in Copy Service Subprofile)

Method:

EMC_VNXe_StorageConfigurationServiceLeaf.CreateOrModifyElementFromStoragePool

- Description**

This method allows the client to create a new *StorageVolume* from the input *StoragePool*, or modify an existing *StorageVolume*.

NOTE:

- 1) Length of the volume name specified in ElementName is limited in 63 bytes.
- 2) Capacity of the volume specified in Size should be a minimum of 1GB and a maximum of 16TB.

- Parameters**

Table 195 - Signature and parameters of EMC_VNXe_StorageConfigurationServiceLeaf.CreateOrModifyElementFromStoragePool

Parameter name	Qualifier	Type	Description/notes
ElementName	IN, NULL allowed	string	An end user relevant name for the LUN being created or renamed. NOTE: <ul style="list-style-type: none"> If the client creates a <i>StorageVolume</i> but does not provide an <i>ElementName</i>, a system default name such as <i>LUN_<creation time></i> will be used.
ElementType	IN, Mandatory	uint16	Enumeration indicating the type of element being created or modified. NOTE: <ul style="list-style-type: none"> Only 2: Storage Volume and 5: Thin Provisioned Volume' are allowed.
Job	OUT	CIM_ConcreteJob Ref	Reference to the job.
Goal	IN, NULL allowed	CIM_StorageSettingRef	The requirements for the element to maintain.
Size	IN, OUT, Conditional	uint64	As an input parameter, size specifies the desired size. NOTE: <ul style="list-style-type: none"> Required when creating or extending size of <i>StorageVolume</i>. NULL allowed when renaming <i>StorageVolume</i>.
InPool	IN, NULL allowed	CIM_StoragePool Ref	The Pool from which to create the element. NOTE: <ul style="list-style-type: none"> If client requires creating a <i>StorageVolume</i>, <i>InPool</i> must be a reference to a concrete pool. If client requires modifying a <i>StorageVolume</i>, <i>InPool</i> should not be provided.
TheElement	IN, OUT, NULL allowed	CIM_LogicalElementRef	As an input parameter: if null, creates a new element. If not null, then the method modifies the specified element. As an output parameter, it is a reference to the resulting element.
EMCEndPoints	IN, Ignored	CIM_SCSIProtocolEndpointRef []	By default when a new device is created, it will be mapped to all available adapters and ports. To override this default behavior, supply one or more end points.
EMCNumberOfDevices	IN, Ignored	uint32	To request more than one device to be created, supply a number for this parameter.

Parameter name	Qualifier	Type	Description/notes
EMCMapElements	IN, Ignored	boolean	To prevent this mapping operation, indicate false for this parameter.

- **Return results**

**Table 196 - Possible return code of
EMC_VNXe_StorageConfigurationServiceLeaf.CreateOrModifyElementFromStoragePool**

Return code	Type	Description
4096	uint32	Method parameters checked, job started. All input parameters are valid, no error or exception occurs when starting the job.
4	uint32	Failed: <ul style="list-style-type: none"> • Any exception is thrown when calling underlying component.
5	uint32	Invalid parameter: <ul style="list-style-type: none"> • If ElementType is not 2: Storage Volume or 5: Thin Provisioned Storage Volume when creating a new volume or modifying an existing volume. • If ElementName is not valid. • If InPool is NULL or not type of EMC_VNXe_MappedStoragePoolLeaf when creating a new volume. • If Size is NULL when creating a new volume. • If Goal is not type of EMC_VNXe_StorageSettingLeaf or EMC_VNXe_GeneratedSettingLeaf or not supported by the InPool when creating a new volume or modifying an existing volume. • If TheElement is not type of EMC_VNXe_StorageVolumeLeaf when modifying an existing storage volume. • If InPool has value when modifying an existing storage volume.
4097	uint32	Size not supported: <ul style="list-style-type: none"> • If Size is less than the minimal or larger than the maximum value that the system supported.
NA	CIM_Error	Exceptions will be thrown when any of the below cases is met: <ul style="list-style-type: none"> • Any of the required parameters is NULL. • Any of optional input Goal is not of the right CIM/SNIA type defined in the MOF files. • InPool is not type of CIM_StoragePool. • The Element is not type of CIM_LogicalElement.

Return code	Type	Description
		<ul style="list-style-type: none"> Any of the Instance References – InPool and TheElement cannot be found in the system.

Method: EMC_VNXe_StorageConfigurationServiceLeaf.ReturnToStoragePool

- Description**

This method allows client to delete an existing *StorageVolume* or *SnapVolume* previously created from a *StoragePool*. The freed space is returned to the source *StoragePool*.

- Parameters**

Table 197 - Signature and parameters of EMC_VNXe_StorageConfigurationServiceLeaf.ReturnToStoragePool

Parameter name	Qualifier	Type	Description/notes
TheElement	IN, Mandatory	CIM_LogicalElementRef	Reference to the storage element to return to the storage pool.
Job	OUT	CIM_ConcreteJobRef	Reference to the job.

- Return Results**

Table 198 - Possible return code of EMC_VNXe_StorageConfigurationServiceLeaf.ReturnToStoragePool

Return code	Type	Description
4096	uint32	Method parameters checked, job started. All input parameters are valid, no error or exception occurs when starting the job.
4	uint32	Failed: <ul style="list-style-type: none"> Any exception is thrown when calling underlying component.
5	uint32	Invalid parameter: <ul style="list-style-type: none"> If TheElement is not type of EMC_VNXe_StorageVolumeLeaf. If TheElement is not type of EMC_VNXe_SnapVolumeLeaf. (for Copy Service)
NA	CIM_Error	The ECOM will throw exceptions when any of the below cases are met: <ul style="list-style-type: none"> Any of the required parameters is NULL. The Element is not type of CIM_LogicalElement. Any of the Instance References – TheElement cannot be found in the system.

Client considerations

Model specification

- StoragePool Manipulation
StoragePool creation, deletion and modification are not supported by VNXe SMI-S Provider.
- LogicalDevice Manipulation
LogicalDisk is **Read-Only** on the VNXe SMI-S.
StorageVolume is only supported to be created from *StoragePool*.
- StorageSetting (to StorageCapabilities)
All *CIM_StorageSetting* (associated to *CIM_StorageCapabilities*) are predefined, which means property *ChangeableType* is set to '0: Fixed, Not changeable'. User generated storage settings are not supported.
Different storage pool can be linked to the same *CIM_StorageSetting* instance because only predefined storage settings are supported.
- StorageVolume (LUN) Creation
When SMI-S API creates a *StorageVolume*, it will create a standalone virtual disk. The *StorageVolume* will get the default FAST VP tiering policy of "Start high then auto-tier".
- StorageVolume (LUN) Enumeration
All Virtual Disks, no matter they are Standalone or in an Application (modeled as *CIM_ReplicationGroup*), and also no matter they are in Generic SCSI Storage Application or other type of Applications. In SMI-S view, there will be one *CIM_StorageVolume* instance per virtual disk to CIM client.
- StorageVolume (LIN) Deletion
A *StorageVolume* can be deleted successfully through SMI-S API under following conditions:
 - 1) There is no host access;
 - 2) There is no snapshot on the *StorageVolume*;
 - 3) The *StorageVolume* is not involved by any Replication Group.Since Application is modeled as SMI-S Replication Group while Virtual Disks in Applications other than Generic SCSI Storage cannot be removed from its owner Application. As a result, SMI-S clients are only able to delete Standalone Virtual Disk or Virtual Disks within Generic SCSI Storage Application. It also avoids the deletion of Virtual Disks for other purposes.

Use case: Discover system's Storage Configuration Capabilities

This use case describes how to discover system's Storage Configuration Capabilities.

1. From Block Server (Top-level Array ComputerSystem), traverse *CIM_HostedService* to *CIM_StorageConfigurationService* to get storage configuration service provided by the system.
2. From *CIM_StorageConfigurationService*, traverse *CIM_ElementCapabilities* to *CIM_StorageConfigurationCapabilities* to get system's storage configuration capabilities.

Use case: Discover pool's Storage configuration Capabilities

This use case describes how to discover pool's Storage Configuration Capabilities.

1. From Block Server (Top-level Array ComputerSystem), traverse *CIM_HostedStoragePool* to *CIM_StoragePool* to get storage pools hosted by the system.
2. From *CIM_StoragePool*, traverse *CIM_ElementCapabilities* to *CIM_StorageConfigurationCapabilities* to get pool's storage configuration capabilities.

Use case: create StorageVolume (LUN) from StoragePool

This use case describes how to create a StorageVolume (LUN) from a storage pool.

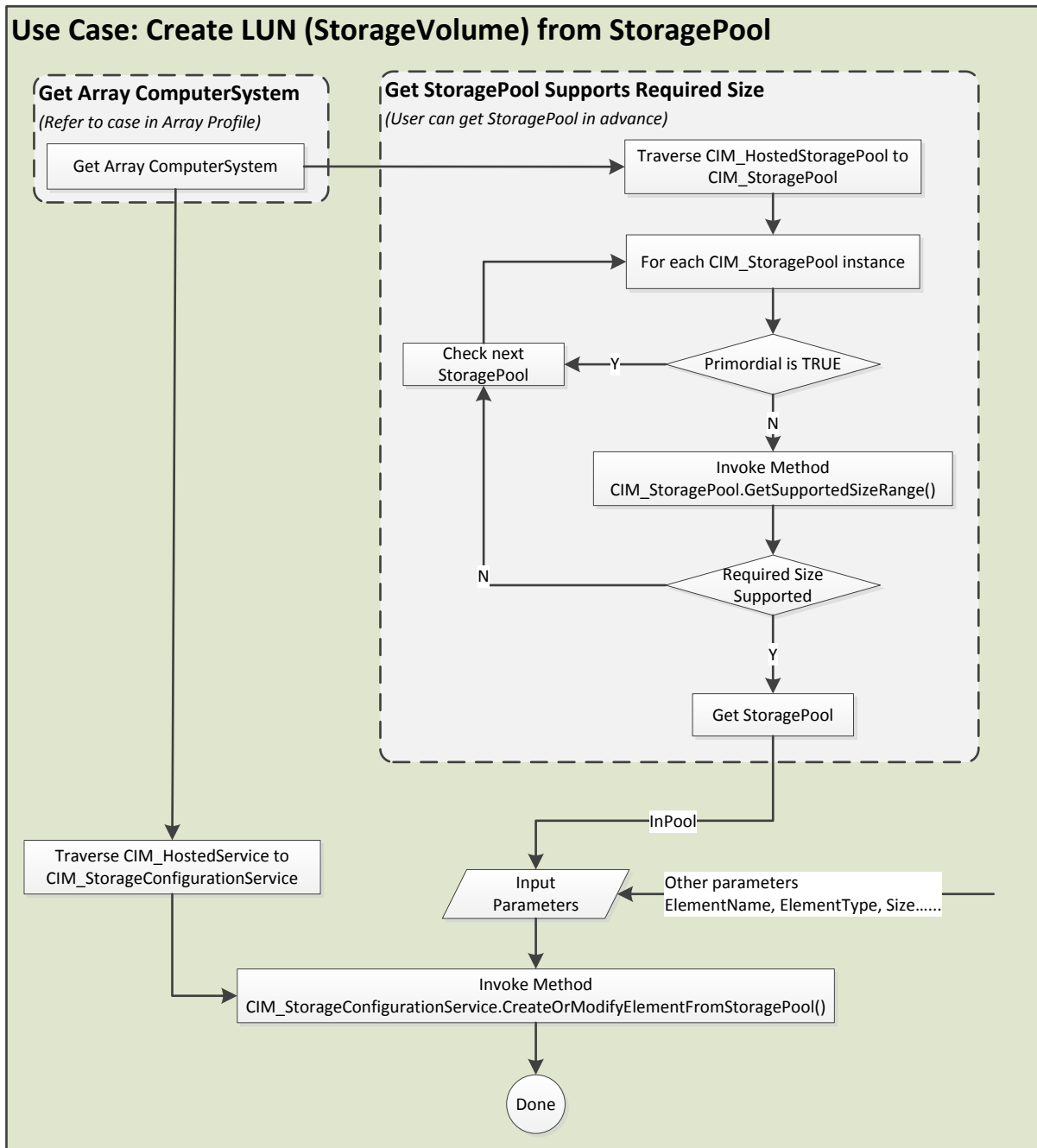


Figure 25 - Flowchart of creating LUN from StoragePool

Use case: Expand a StorageVolume (LUN)

This use case describes how to expand a StorageVolume (LUN).

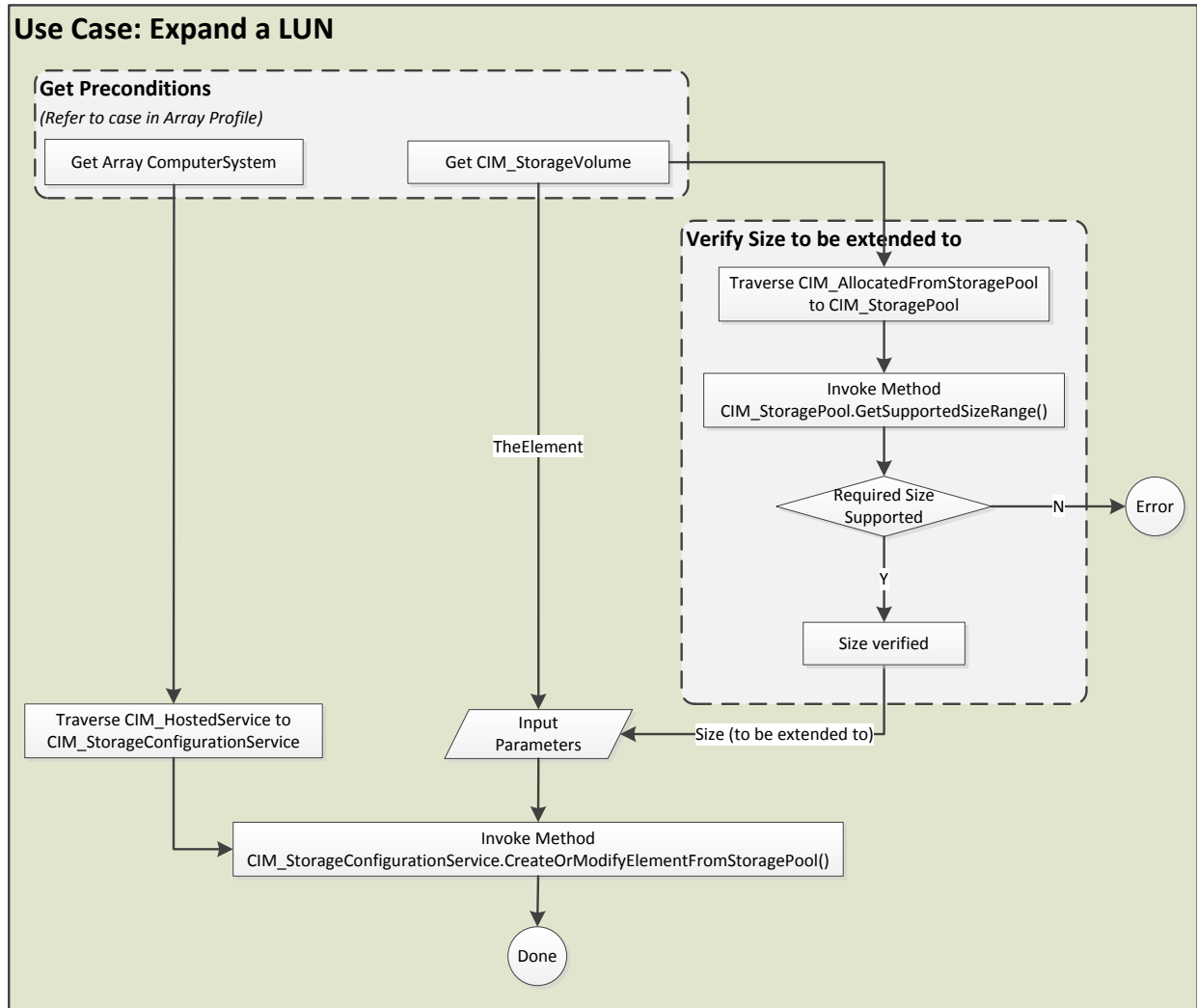


Figure 26 - Flowchart of expanding a LUN

Use case: Rename a StorageVolume (LUN)

This use case describes how to rename a StorageVolume (LUN).

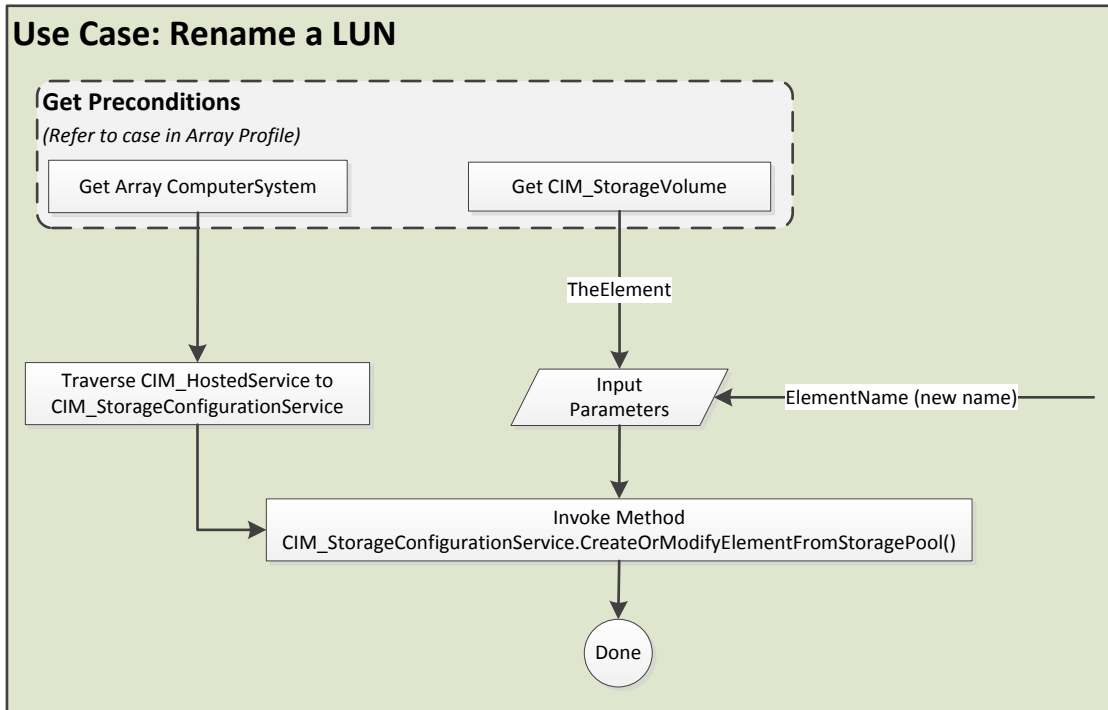


Figure 27 - Flowchart of Renaming a LUN

Use case: Delete StorageVolume (LUN)

This use case describes how to delete a StorageVolume (LUN).

Users can invoke *CIM_StorageConfigurationService.ReturnToStoragePool* to delete a *StorageVolume*.

The storage capacity used by the LUN will be automatically returned to the *StoragePool* where the LUN was allocated from.

CIM Elements

The implemented classes and associations related to the Block Services Package on the VNXe Storage System are described as follows:

Table 199 - CIM Elements implemented on the VNXe for Block Services Package

CIM Class	Implemented Class	Description
CIM_AllocatedFromStoragePool	EMC_VNXe_MappedStoragePool_LogicalDisk_AllocatedFromStoragePoolAssocLeaf	Represents the association between the Logical Disk and Concrete Storage Pool.
CIM_AllocatedFromStoragePool	EMC_VNXe_PrimordialPool_MappedStoragePool_AllocatedFromStoragePoolAssocLeaf	Represents the association between the Primordial Pool and Concrete Storage Pool.
CIM_AllocatedFromStoragePool	EMC_VNXe_MappedStoragePool_StorageVolume_AllocatedFromStoragePoolAssocLeaf	Represents the association between the Storage Volume and Concrete Storage Pool.

CIM Class	Implemented Class	Description
CIM_ElementCapabilities	EMC_VNXe_MappedStoragePool_MappedStoragePoolCapabilities_ElementCapabilitiesAssocLeaf	Represents the association between the Concrete Storage Pool and the Concrete Storage Pool Capabilities.
CIM_ElementCapabilities	EMC_VNXe_PrimordialPool_PrimordialPoolCapabilities_ElementCapabilitiesAssocLeaf	Represents the association between the Primordial Storage Pool and the Primordial Storage Pool Capabilities.
CIM_ElementCapabilities	EMC_VNXe_StorageConfigurationService_StorageConfigurationCapabilities_ElementCapabilitiesAssocLeaf	Represents the association between the Storage Configuration Service and the global Storage Configuration Capabilities.
CIM_ElementCapabilities	EMC_VNXe_StorageConfigurationService_StorageConfigurationServiceCapabilities_ElementCapabilitiesAssocLeaf	Represents the association between the Storage Configuration Service and the global Storage Pool Capabilities.
CIM_ElementCapabilities	EMC_VNXe_MappedStoragePool_MappedStoragePoolStorageConfigurationCapabilities_ElementCapabilitiesAssocLeaf	Represents the association between the Concrete Storage Pool and StorageConfigurationCapabilities of the Concrete Storage Pool.
CIM_ElementCapabilities	EMC_VNXe_PrimordialPool_PrimordialPoolStorageConfigurationCapabilities_ElementCapabilitiesAssocLeaf	Represents the association between the Primordial Storage Pool and StorageConfigurationCapabilities of the Primordial Storage Pool.
CIM_ElementSettingData	EMC_VNXe_LogicalDisk_LogicalDiskSetting_ElementSettingDataAssocLeaf	Represents the association between the Logical Disk and the Logical Disk Setting.
CIM_ElementSettingData	EMC_VNXe_StorageVolume_StorageVolumeSetting_ElementSettingDataAssocLeaf	Represents the association between the Storage Volume and the Storage Volume Setting.
CIM_HostedStoragePool	EMC_VNXe_StorageSystem_MappedStoragePool_HostedStoragePoolAssocLeaf	Represents the association between the Storage System and the Concrete Storage Pool.
CIM_HostedStoragePool	EMC_VNXe_StorageSystem_PrimordialPool_HostedStoragePoolAssocLeaf	Represents the association between the Storage System and the Primordial Storage Pool.
CIM_LogicalDisk	EMC_VNXe_LogicalDiskLeaf	Represents the Logical Disk.
CIM_StorageCapabilities	EMC_VNXe_MappedStoragePoolCapabilitiesLeaf	Represents the Storage Capabilities of Concrete Storage Pool.

CIM Class	Implemented Class	Description
CIM_StorageCapabilities	EMC_VNXe_PrimordialPoolCapabilitiesLeaf	Represents the Storage Capabilities of Primordial Storage Pool.
CIM_StorageCapabilities	EMC_VNXe_StorageConfigurationServiceCapabilitiesLeaf	Represents the Storage Capabilities of the System.
CIM_StorageConfigurationCapabilities (Pool's)	EMC_VNXe_PrimordialPoolStorageConfigurationCapabilitiesLeaf	Represents the Storage Configuration Capabilities of Primordial Storage Pool.
CIM_StorageConfigurationCapabilities (Pool's)	EMC_VNXe_StorageConfigurationCapabilitiesLeaf	Represents the Storage Configuration Capabilities of the System.
CIM_StorageConfigurationCapabilities (System's)	EMC_VNXe_MappedStoragePoolStorageConfigurationCapabilitiesLeaf	Represents the Storage Configuration Capabilities of Concrete Storage Pool.
CIM_StorageConfigurationService	EMC_VNXe_StorageConfigurationServiceLeaf	Represents the Storage Configuration Service.
CIM_StoragePool (Concrete)	EMC_VNXe_MappedStoragePoolLeaf	Represents the Concrete Storage Pool.
CIM_StoragePool (Primordial)	EMC_VNXe_PrimordialPoolLeaf	Represents the Primordial Storage Pool.
CIM_StorageSetting	EMC_VNXe_GeneratedSettingLeaf	Represents the generated Storage Setting output from CreateSetting.
CIM_StorageSetting	EMC_VNXe_StorageSettingLeaf	Represents the global Storage Setting.
CIM_StorageSetting	EMC_VNXe_StorageVolumeSettingLeaf	Represents the Storage Setting of Storage Volume.
CIM_StorageSetting	EMC_VNXe_LogicalDiskSettingLeaf	Represents the Storage Setting of Logical Disk.
CIM_StorageSettingsAssociatedToCapabilities	EMC_VNXe_MappedStoragePoolCapabilities_StorageSetting_StorageSettingsAssociatedToCapabilitiesAssocLeaf	Represents the association between the Concrete Storage Pool Capabilities and the global Storage Setting.
CIM_StorageSettingsGeneratedFromCapabilities	EMC_VNXe_MappedStoragePoolCapabilities_GeneratedSetting_StorageSetting_StorageSettingsGeneratedFromCapabilitiesAssocLeaf	Represents the association between the Concrete Storage Pool Capabilities and the Storage Setting generated from it.
CIM_StorageVolume	EMC_VNXe_StorageVolumeLeaf	Represents the Storage Volume.
CIM_SystemDevice	EMC_VNXe_StorageSystem_LogicalDisk_SystemDeviceAssocLeaf	Represents the association between the Storage System and

CIM Class	Implemented Class	Description
		the Logical Disk.
CIM_SystemDevice	EMC_VNXe_StorageSystem_PoolGapExtent_SystemDeviceAssocLeaf	Represents the association between the Storage System and the Pool Gap Extent.
CIM_SystemDevice	EMC_VNXe_StorageSystem_StorageVolume_SystemDeviceAssocLeaf	Represents the association between the Storage System and the Storage Volume.

EMC_VNXe_PrimordialPoolLeaf

Table 200 - Referenced properties/methods for EMC_VNXe_PrimordialPoolLeaf

CIM property	Description/notes
InstanceID	Unique string to identify the instance. Set as PoolID
PoolID	FriendlyID of the pool
ElementName	Set as PoolID
Primordial	Set to TRUE.
TotalManagedSpace	Total capacity of the pool.
RemainingManagedSpace	Remaining usable capacity of the pool.
Usage	Set to {2: Unrestricted }
OperationalStatus	Current operational status of the pool.

EMC_VNXe_MappedStoragePoolLeaf

Table 201 - Referenced properties/methods for EMC_VNXe_MappedStoragePoolLeaf

CIM property	Description/notes
InstanceID	Unique string to identify the instance. Set as PoolID
PoolID	FriendlyID of the pool
ElementName	Set as Name
Name	Name of the pool
Primordial	Set to FALSE.
TotalManagedSpace	Total capacity of the pool.

CIM property	Description/notes
RemainingManagedSpace	Remaining usable capacity of the pool.
SpaceLimit	Set as TotalManagedSpace
SpaceLimitDetermination	Set to {4: Limitless}
ThinProvisionMetaDataSpace	Size of metadata consumed by this pool.
Usage	Set to {2: Unrestricted }
OperationalStatus	Current operational status of the pool.

EMC_VNXe_LogicalDiskLeaf

Table 202 - Referenced properties/methods for EMC_VNXe_LogicalDiskLeaf

CIM property	Description/notes
SystemCreationClassName	Set to EMC_VNXe_StorageSystemLeaf
SystemName	Set as system name of VNXe
CreationClassName	Set to EMC_VNXe_LogicalDiskLeaf
DeviceID	Friendly ID of the Filesystem resides on the LogicalDisk.
ElementName	Name of the Filesystem resides on the LogicalDisk.
Name	Friendly ID of the Filesystem resides on the LogicalDisk.
NameFormat	Set to 1: Other
NameNamespace	Set to 8: OS Device Namespace
OperationalStatus	Current operation status of the LogicalDisk.
BlockSize	Set to 512
NumberOfBlocks	Block numbers of the LogicalDisk.
ConsumableBlocks	Consumable block numbers of the LogicalDisk.
Primordial	Set to FALSE
ExtentStatus	Set to [17: Export]
ExtentDiscriminator	Set to ['SNIA: Allocated]
ThinlyProvisioned	Indicates if the LogicalDisk is thinly provisioned.
DataRedundancy	Number of complete copies of data currently maintained.
DeltaReservation	Percentage that specifies the amount of space that should be reserved

CIM property	Description/notes
	in a replica for caching changes.
PackageRedundancy	Number of physical packages can currently fail without data loss.
IsBasedOnUnderlyingRedundancy	Set to TRUE
HealthState	Current health state of the extent
NoSinglePointOfFailure	Set to FALSE
IsComposite	Set to FALSE
Usage	Set to 2: Unrestricted
EnabledState	Set to 5: Not Applicable
RequestedState	Set to 12: Not Applicable
TransitioningToState	Set to 12: Not Applicable

EMC_VNXe_StorageVolumeLeaf

Table 203 - Referenced properties/methods for EMC_VNXe_StorageVolumeLeaf

CIM property	Description/notes
SystemCreationClassName	Set to EMC_VNXe_StorageSystemLeaf
SystemName	Set as system name of VNXe
CreationClassName	Set to EMC_VNXe_StorageVolumeLeaf
DeviceID	Friendly ID of the StorageVolume (LUN).
Access	Indicates the read and write support.
ElementName	Name of the StorageVolume (LUN).
Name	Friendly ID of the StorageVolume (LUN).
NameFormat	Set to 7: SNVM
NameNamespace	Set to 7: SNVM
OtherIdentifyingInfo	Additional data for DeviceID information.
IdentifyingDescription	Set to [NAA;VPD83Type3]
OperationalStatus	Current operation status of the StorageVolume (LUN).
BlockSize	Set to 512
NumberOfBlocks	Block numbers of the StorageVolume (LUN).

CIM property	Description/notes
ConsumableBlocks	Consumable block numbers of the StorageVolume (LUN).
Primordial	Set to FALSE
ExtentStatus	Set to [1: Unknown]
ExtentDiscriminator	Set to ['SNIA: Allocated]
ThinlyProvisioned	Indicates if the StorageVolume (LUN) is thinly provisioned.
DataRedundancy	Number of complete copies of data currently maintained.
DeltaReservation	Percentage that specifies the amount of space that should be reserved in a replica for caching changes.
PackageRedundancy	Number of physical packages can currently fail without data loss.
ParityLayout	Parity layout of the StorageVolume (LUN).
EMCRaidLevel	Raid level of the StorageVolume (LUN).
IsBasedOnUnderlyingRedundancy	Set to TRUE
HealthState	Current health state of the extent
NoSinglePointOfFailure	Set to FALSE
IsComposite	Set to FALSE
Usage	Set to 2: Unrestricted
EnabledState	Set to 5: Not Applicable
RequestedState	Set to 12: Not Applicable
TransitioningToState	Set to 12: Not Applicable

EMC_VNXe_StorageConfigurationServiceLeaf

Table 204 - Referenced properties/methods for EMC_VNXe_StorageConfigurationServiceLeaf

CIM property	Description/notes
SystemCreationClassName	Set to EMC_VNXe_StorageSystemLeaf
SystemName	Set as system name of VNXe
CreationClassName	Set to EMC_VNXe_StorageConfigurationServiceLeaf
Name	Set to StorageConfigurationService

EMC_VNXe_PrimitivePoolStorageConfigurationCapabilitiesLeaf

**Table 205 - Referenced properties/methods for
EMC_VNXe_PrimitivePoolStorageConfigurationCapabilitiesLeaf**

CIM property	Description/notes
InstanceID	Set as the InstanceID of the Primitive pool.
ElementName	Same as InstanceID.
SupportedAsynchronousActions	Set to empty array.
SupportedSynchronousActions	Set to empty array.
SupportedStorageElementFeatures	Set to empty array.
SupportedStorageElementTypes	Set to empty array.
SupportedStoragePoolFeatures	Set to empty array.
ThinProvisionedDefaultReserve	Set to 0.
ThinProvisionedClientSettableReserve	Set to FALSE.

EMC_VNXe_MappedStoragePoolStorageConfigurationCapabilitiesLeaf

**Table 206 - Referenced properties/methods for
EMC_VNXe_MappedStoragePoolStorageConfigurationCapabilitiesLeaf**

CIM property	Description/notes
InstanceID	Set as the InstanceID of the Concrete pool.
ElementName	Same as InstanceID.
SupportedAsynchronousActions	Set to [5: Storage Element Creation, 6: Storage Element Return, 7: Storage Element Modification].
SupportedSynchronousActions	Set to empty array.
SupportedStorageElementFeatures	Set to [6: Single InPool, 3: StorageVolume Creation, 5: StorageVolume Modification, 12: Element Capacity Expansion].
SupportedStorageElementTypes	Set to [2: StorageVolume, 5: ThinlyProvisionedStorageVolume].
SupportedStoragePoolFeatures	Set to empty array.
SupportedCopyTypes	Set to [4: UnSyncAssoc]
ThinProvisionedDefaultReserve	Set to 0.
ThinProvisionedClientSettableReserve	Set to FALSE.

EMC_VNXe_StorageConfigurationCapabilitiesLeaf

Table 207 - Referenced properties/methods for EMC_VNXe_StorageConfigurationCapabilitiesLeaf

CIM property	Description/notes
InstanceID	Set as the InstanceID of the Primordial pool.
ElementName	Same as InstanceID.
MaximumElementCreateCount	Set to 1.
MaximumElementDeleteCount	Set to 1.
SupportedAsynchronousActions	Set to [5: Storage Element Creation, 6: Storage Element Return, 7: Storage Element Modification, 8: Replica Creation, 9: Replica Modification].
SupportedSynchronousActions	Set to empty array.
SupportedStorageElementFeatures	Set to [6: Single InPool, 3: StorageVolume Creation, 5: StorageVolume Modification, 12: Element Capacity Expansion].
SupportedStorageElementTypes	Set to [2: StorageVolume, 5: ThinlyProvisionedStorageVolume].
SupportedStoragePoolFeatures	Set to empty array.
SupportedCopyTypes	Set to [4: UnSyncAssoc]
ThinProvisionedDefaultReserve	Set to 0.
ThinProvisionedClientSettableReserve	Set to FALSE.

EMC_VNXe_PrimordialPoolCapabilitiesLeaf

Table 208 - Referenced properties/methods for EMC_VNXe_PrimordialPoolCapabilitiesLeaf

CIM property	Description/notes
InstanceID	Unique ID for this instance.
ElementName	User friendly name for this instance.
ElementType	Type of storage element to which this StorageCapabilities applies. Set to {5: StoragePool}
NoSinglePointOfFailure	Indicates whether 'no single point of failure' is supported.
NoSinglePointOfFailureDefault	Default value for NoSinglePointOfFailure.
DataRedundancyMin	Minimum number of complete copies of data that can be maintained.
DataRedundancyMax	Maximum number of complete copies of data that can be maintained.
DataRedundancyDefault	Default number of complete copies of data that can be maintained.

CIM property	Description/notes
PackageRedundancyMin	Minimum number of redundancy packages that will be used.
PackageRedundancyMax	Maximum number of redundancy packages that will be used.
PackageRedundancyDefault	Default number of redundancy packages that will be used.
ExtentStripeLengthDefault	Default number of underlying StorageExtents across which data is striped.
ParityLayoutDefault	Indicates whether a parity-based storage organization is using rotated or non-rotated.

EMC_VNXe_MappedStoragePoolCapabilitiesLeaf

Table 209 - Referenced properties/methods for EMC_VNXe_MappedStoragePoolCapabilitiesLeaf

CIM property	Description/notes
InstanceID	Unique ID for this instance.
ElementName	User friendly name for this instance.
ElementType	Type of storage element to which this StorageCapabilities applies. Set to {5: StoragePool}
NoSinglePointOfFailure	Indicates whether 'no single point of failure' is supported.
NoSinglePointOfFailureDefault	Default value for NoSinglePointOfFailure.
DataRedundancyMin	Minimum number of complete copies of data that can be maintained.
DataRedundancyMax	Maximum number of complete copies of data that can be maintained.
DataRedundancyDefault	Default number of complete copies of data that can be maintained.
PackageRedundancyMin	Minimum number of redundancy packages that will be used.
PackageRedundancyMax	Maximum number of redundancy packages that will be used.
PackageRedundancyDefault	Default number of redundancy packages that will be used.
ExtentStripeLengthDefault	Default number of underlying StorageExtents across which data is striped.
ParityLayoutDefault	Indicates whether a parity-based storage organization is using rotated or non-rotated.

EMC_VNXe_StorageConfigurationServiceCapabilitiesLeaf

Table 210 - Referenced properties/methods for EMC_VNXe_StorageConfigurationServiceCapabilitiesLeaf

CIM property	Description/notes
InstanceID	Unique ID for this instance.
ElementName	User friendly name for this instance.
ElementType	Type of storage element to which this StorageCapabilities applies. Set to {6: StorageConfigurationService}
NoSinglePointOfFailure	Indicates whether 'no single point of failure' is supported.
NoSinglePointOfFailureDefault	Default value for NoSinglePointOfFailure.
DataRedundancyMin	Minimum number of complete copies of data that can be maintained.
DataRedundancyMax	Maximum number of complete copies of data that can be maintained.
DataRedundancyDefault	Default number of complete copies of data that can be maintained.
PackageRedundancyMin	Minimum number of redundancy packages that will be used.
PackageRedundancyMax	Maximum number of redundancy packages that will be used.
PackageRedundancyDefault	Default number of redundancy packages that will be used.
ExtentStripeLengthDefault	Default number of underlying StorageExtents across which data is striped.
ParityLayoutDefault	Indicates whether a parity-based storage organization is using rotated or non-rotated.

EMC_VNXe_GeneratedSettingLeaf

Table 211 - Referenced properties/methods for EMC_VNXe_GeneratedSettingLeaf

CIM property	Description/notes
InstanceID	Unique ID for this instance.
ElementName	User friendly name for this instance.
NoSinglePointOfFailure	Indicates whether 'no single point of failure' is supported.
ChangeableType	Set to {1: Changeable – Transient}
ParityLayout	Indicates whether a parity-based storage organization is using rotated or non-rotated.
DataRedundancyMin	Minimum number of complete copies of data that can be maintained.
DataRedundancyMax	Maximum number of complete copies of data that can be maintained.
DataRedundancyGoal	Desired number of complete copies of data that can be maintained.
PackageRedundancyMin	Minimum number of redundancy packages that will be used.

CIM property	Description/notes
PackageRedundancyMax	Maximum number of redundancy packages that will be used.
PackageRedundancyGoal	Desired number of redundancy packages that will be used.
ExtentStripeLength	Number of underlying StorageExtents across which data is striped.
InitialStorageTierMethodology	Initial storage tier for this element.
InitialStorageTieringSelection	Value used to set the property StorageTieringSelection of the associated element.
RelativePerformanceOrderLimit	The storage tiering limit applied to the element.
InitialRelativePerformanceOrderSet	A set of values. The associated element can only be placed in storage tiers that have a RelativePerformanceOrder value included in this set.

EMC_VNXe_StorageSettingLeaf

Table 212 - Referenced properties/methods for EMC_VNXe_StorageSettingLeaf

CIM property	Description/notes
InstanceID	Unique ID for this instance.
ElementName	User friendly name for this instance.
NoSinglePointOfFailure	Indicates whether 'no single point of failure' is supported.
ChangeableType	Set to {0: Fixed – Not Changeable}
ParityLayout	Indicates whether a parity-based storage organization is using rotated or non-rotated.
DataRedundancyMin	Minimum number of complete copies of data that can be maintained.
DataRedundancyMax	Maximum number of complete copies of data that can be maintained.
DataRedundancyGoal	Desired number of complete copies of data that can be maintained.
PackageRedundancyMin	Minimum number of redundancy packages that will be used.
PackageRedundancyMax	Maximum number of redundancy packages that will be used.
PackageRedundancyGoal	Desired number of redundancy packages that will be used.
ExtentStripeLengthMin	Minimum number of underlying StorageExtents across which data is striped.
ExtentStripeLengthMax	Maximum number of underlying StorageExtents across which data is striped.
ExtentStripeLength	Number of underlying StorageExtents across which data is striped.

EMC_VNXe_LogicalDiskSettingLeaf

Table 213 - Referenced properties/methods for EMC_VNXe_LogicalDiskSettingLeaf

CIM property	Description/notes
InstanceID	Unique ID for this instance.
ElementName	User friendly name for this instance.
NoSinglePointOfFailure	Indicates whether 'no single point of failure' is supported.
ChangeableType	Set to {2: Changeable –Persistent}
ParityLayout	Indicates whether a parity-based storage organization is using rotated or non-rotated.
DataRedundancyMin	Minimum number of complete copies of data that can be maintained.
DataRedundancyMax	Maximum number of complete copies of data that can be maintained.
DataRedundancyGoal	Desired number of complete copies of data that can be maintained.
PackageRedundancyMin	Minimum number of redundancy packages that will be used.
PackageRedundancyMax	Maximum number of redundancy packages that will be used.
PackageRedundancyGoal	Desired number of redundancy packages that will be used.
ExtentStripeLengthMin	Minimum number of underlying StorageExtents across which data is striped.
ExtentStripeLengthMax	Maximum number of underlying StorageExtents across which data is striped.
ExtentStripeLength	Number of underlying StorageExtents across which data is striped.

EMC_VNXe_StorageVolumeSettingLeaf

Table 214 - Referenced properties/methods for EMC_VNXe_StorageVolumeSettingLeaf

CIM property	Description/notes
InstanceID	Unique ID for this instance.
ElementName	User friendly name for this instance.
NoSinglePointOfFailure	Indicates whether no single point of failure' is supported.
ChangeableType	Set to {2: Changeable –Persistent}
ParityLayout	Indicates whether a parity-based storage organization is using rotated or non-rotated.
DataRedundancyMin	Minimum number of complete copies of data that can be maintained.

CIM property	Description/notes
DataRedundancyMax	Maximum number of complete copies of data that can be maintained.
DataRedundancyGoal	Desired number of complete copies of data that can be maintained.
PackageRedundancyMin	Minimum number of redundancy packages that will be used.
PackageRedundancyMax	Maximum number of redundancy packages that will be used.
PackageRedundancyGoal	Desired number of redundancy packages that will be used.
ExtentStripeLengthMin	Minimum number of underlying StorageExtents across which data is striped.
ExtentStripeLengthMax	Maximum number of underlying StorageExtents across which data is striped.
ExtentStripeLength	Number of underlying StorageExtents across which data is striped.

EMC_VNXe_PrimordialPool_MappedStoragePool_AllocatedFromStoragePoolAssocLeaf

**Table 215 - Referenced properties/methods for
EMC_VNXe_PrimordialPool_MappedStoragePool_AllocatedFromStoragePoolAssocLeaf**

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_PrimordialPoolLeaf
Dependent	Reference of EMC_VNXe_MappedStoragePoolLeaf

EMC_VNXe_MappedStoragePool_LogicalDisk_AllocatedFromStoragePoolAssocLeaf

**Table 216 - Referenced properties/methods for
EMC_VNXe_MappedStoragePool_LogicalDisk_AllocatedFromStoragePoolAssocLeaf**

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_MappedStoragePoolLeaf
Dependent	Reference of EMC_VNXe_LogicalDiskLeaf

EMC_VNXe_MappedStoragePool_StorageVolume_AllocatedFromStoragePoolAssocLeaf

**Table 217 - Referenced properties/methods for
EMC_VNXe_MappedStoragePool_StorageVolume_AllocatedFromStoragePoolAssocLeaf**

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_MappedStoragePoolLeaf
Dependent	Reference of EMC_VNXe_StorageVolumeLeaf

[EMC_VNXe_StorageConfigurationService_StorageConfigurationCapabilities_ElementCapabilitiesAssocLeaf](#)

Table 218 - Referenced properties/methods for EMC_VNXe_StorageConfigurationService_StorageConfigurationCapabilities_ElementCapabilitiesAssocLeaf

CIM property	Description/notes
ManagedElement	Reference of EMC_VNXe_StorageConfigurationServiceLeaf
Capabilities	Reference of EMC_VNXe_StorageConfigurationCapabilitiesLeaf

[EMC_VNXe_PrimordialPool_PrimordialPoolStorageConfigurationCapabilities_ElementCapabilitiesAssocLeaf](#)

Table 219 - Referenced properties/methods for EMC_VNXe_PrimordialPool_PrimordialPoolStorageConfigurationCapabilities_ElementCapabilitiesAssocLeaf

CIM property	Description/notes
ManagedElement	Reference of EMC_VNXe_PrimordialPoolLeaf
Capabilities	Reference of EMC_VNXe_PrimordialPoolStorageConfigurationCapabilitiesLeaf

[EMC_VNXe_MappedStoragePool_MappedStoragePoolStorageConfigurationCapabilities_ElementCapabilitiesAssocLeaf](#)

Table 220 - Referenced properties/methods for EMC_VNXe_MappedStoragePool_MappedStoragePoolStorageConfigurationCapabilities_ElementCapabilitiesAssocLeaf

CIM property	Description/notes
ManagedElement	Reference of EMC_VNXe_MappedStoragePoolLeaf
Capabilities	Reference of EMC_VNXe_MappedStoragePoolStorageConfigurationCapabilitiesLeaf

[EMC_VNXe_StorageConfigurationService_StorageConfigurationServiceCapabilities_ElementCapabilitiesAssocLeaf](#)

Table 221 - Referenced properties/methods for EMC_VNXe_StorageConfigurationService_StorageConfigurationServiceCapabilities_ElementCapabilitiesAssocLeaf

CIM property	Description/notes
ManagedElement	Reference of EMC_VNXe_StorageConfigurationServiceLeaf
Capabilities	Reference of EMC_VNXe_StorageConfigurationServiceCapabilitiesLeaf

EMC_VNXe_PrimordialPool_PrimordialPoolCapabilities_ElementCapabilitiesAssocLeaf

**Table 222 - Referenced properties/methods for
EMC_VNXe_PrimordialPool_PrimordialPoolCapabilities_ElementCapabilitiesAssocLeaf**

CIM property	Description/notes
ManagedElement	Reference of EMC_VNXe_PrimordialPoolLeaf
Capabilities	Reference of EMC_VNXe_PrimordialPoolCapabilitiesLeaf

EMC_VNXe_MappedStoragePool_MappedStoragePoolCapabilities_ElementCapabilitiesAssocLeaf

**Table 223 - Referenced properties/methods for
EMC_VNXe_MappedStoragePool_MappedStoragePoolCapabilities_ElementCapabilitiesAssocLeaf**

CIM property	Description/notes
ManagedElement	Reference of EMC_VNXe_MappedStoragePoolLeaf
Capabilities	Reference of EMC_VNXe_MappedStoragePoolCapabilitiesLeaf

EMC_VNXe_LogicalDisk_LogicalDiskSetting_ElementSettingDataAssocLeaf

**Table 224 - Referenced properties/methods for
EMC_VNXe_LogicalDisk_LogicalDiskSetting_ElementSettingDataAssocLeaf**

CIM property	Description/notes
ManagedElement	Reference of EMC_VNXe_LogicalDiskLeaf
SettingData	Reference of EMC_VNXe_LogicalDiskSettingLeaf

EMC_VNXe_StorageVolume_StorageVolumeSetting_ElementSettingDataAssocLeaf

**Table 225 - Referenced properties/methods for
EMC_VNXe_StorageVolume_StorageVolumeSetting_ElementSettingDataAssocLeaf**

CIM property	Description/notes
ManagedElement	Reference of EMC_VNXe_StorageVolumeLeaf
SettingData	Reference of EMC_VNXe_StorageVolumeSettingLeaf

EMC_VNXe_MappedStoragePoolCapabilities_StorageSetting_StorageSettingsAssociatedToCapabilitiesAssocLeaf

**Table 226 - Referenced properties/methods for
EMC_VNXe_MappedStoragePoolCapabilities_StorageSetting_StorageSettingsAssociatedToCapabilitiesAssocLeaf**

CIM property	Description/notes
--------------	-------------------

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_MappedStoragePoolCapabilitiesLeaf
Dependent	Reference of EMC_VNXe_StorageSettingLeaf
DefaultSetting	Indicates whether the setting is a default setting

EMC_VNXe_MappedStoragePoolCapabilities_GeneratedSetting_StorageSetting_StorageSettingsGeneratedFromCapabilitiesAssocLeaf

Table 227 - Referenced properties/methods for EMC_VNXe_MappedStoragePoolCapabilities_GeneratedSetting_StorageSettingsGeneratedFromCapabilitiesAssocLeaf

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_MappedStoragePoolCapabilitiesLeaf
Dependent	Reference of EMC_VNXe_GeneratedSettingLeaf
DefaultSetting	Indicates whether the setting is a default setting.

EMC_VNXe_StorageSystem_MappedStoragePool_HostedStoragePoolAssocLeaf

Table 228 - Referenced properties/methods for EMC_VNXe_StorageSystem_MappedStoragePool_HostedStoragePoolAssocLeaf

CIM property	Description/notes
GroupComponent	Reference of EMC_VNXe_StorageSystemLeaf
PartComponent	Reference of EMC_VNXe_MappedStoragePoolLeaf

EMC_VNXe_StorageSystem_PrimordialPool_HostedStoragePoolAssocLeaf

Table 229 - Referenced properties/methods for EMC_VNXe_StorageSystem_PrimordialPool_HostedStoragePoolAssocLeaf

CIM property	Description/notes
GroupComponent	Reference of EMC_VNXe_StorageSystemLeaf
PartComponent	Reference of EMC_VNXe_PrimordialPoolLeaf

EMC_VNXe_StorageSystem_LogicalDisk_SystemDeviceAssocLeaf

Table 230 - Referenced properties/methods for EMC_VNXe_StorageSystem_LogicalDisk_SystemDeviceAssocLeaf

CIM property	Description/notes
GroupComponent	Reference of EMC_VNXe_StorageSystemLeaf

CIM property	Description/notes
PartComponent	Reference of EMC_VNXe_LogicalDiskLeaf

EMC_VNXe_StorageSystem_PoolGapExtent_SystemDeviceAssocLeaf

**Table 231 - Referenced properties/methods for
EMC_VNXe_StorageSystem_PoolGapExtent_SystemDeviceAssocLeaf**

CIM property	Description/notes
GroupComponent	Reference of EMC_VNXe_StorageSystemLeaf
PartComponent	Reference of EMC_VNXe_PoolGapExtentLeaf

EMC_VNXe_StorageSystem_StorageVolume_SystemDeviceAssocLeaf

**Table 232 - Referenced properties/methods for
EMC_VNXe_StorageSystem_StorageVolume_SystemDeviceAssocLeaf**

CIM property	Description/notes
GroupComponent	Reference of EMC_VNXe_StorageSystemLeaf
PartComponent	Reference of EMC_VNXe_StorageVolumeLeaf

Thin Provisioning Profile

Overview

The Block Services with Thin Provisioning Profile is a specialization of the Block Services Package, that adds support for Thin provisioning. All the provisions of the Block Services Package apply, in addition to those defined in this profile.

This profile is nearly compatible with the Block Services Package. A client supporting the Block Services Package interacting with a Block Server with Thin Provisioning Profile agent should be able to actively manage Fully-provisioned volumes and pools. Discovery is slightly impacted because the system sees instances of both fully and thinly provisioned pools and volumes. The model is the same, but the client needs to consider the values of thin-provisioning-specific properties to fully understand capacity utilization.

NOTE: For more details, refer to [Clause 30: Block Services Package in Storage Management Technical Specification, Part 3 Block Devices, Version 1.6.0, Revision 4](#).

Class diagram

Refer to [Block Services Package Class Diagram](#)

Method of the Profile

Refer to [Methods of Block Services Package](#)

Client considerations

Model specification

- Pool Capacity

In VNXe, *StoragePool.SpaceLimitDetermination* is set to 4: Limitless. It means that the Block Server does not have a defined limit on the capacity for creating or expanding children. Clients that support Thin provisioning should not use *StoragePool.SpaceLimit*.

Use case: Creating a Thinly provisioned StorageVolume

Set ElementType to 5: Thin Provisioned Volume. Refer to [Use Case: Create LUN \(StorageVolume\)](#)

CIM Element

Refer to [CIM Elements of Block Services Package](#).

Automated Storage Tiering Profile

Overview

The Automated Storage Tiering profile, a component profile, includes classes and methods to expose the storage tiering feature of the storage array.

To enable storage arrays to provide faster access to data based on how frequently the data is accessed, it is helpful to monitor the frequency of data access, enabling the optimized placement of the data in the appropriate storage tier.

In VNXe, SMI-S clients do not have the ability to create the storage tiers. The storage tiers are created by the VNXe system based on the performance characteristics of the underlying hardware and the quality of service associated with that hardware.

Once the storage tiers have been identified, the storage array will monitor the data usage of various storage elements (for example, volumes) and “automatically” move the data to the appropriate storage tier in order to optimize the response time of the applications using that data. At the same time, as the data in high performing storage tiers is used less often, the storage array will move that data to a less performing storage tier.

NOTE: For more details, refer to [Clause 31: Automated Storage Tiering Profile in Storage Management Technical Specification, Part 3 Block Devices, Version 1.5.0, Revision 6](#).

Class diagram

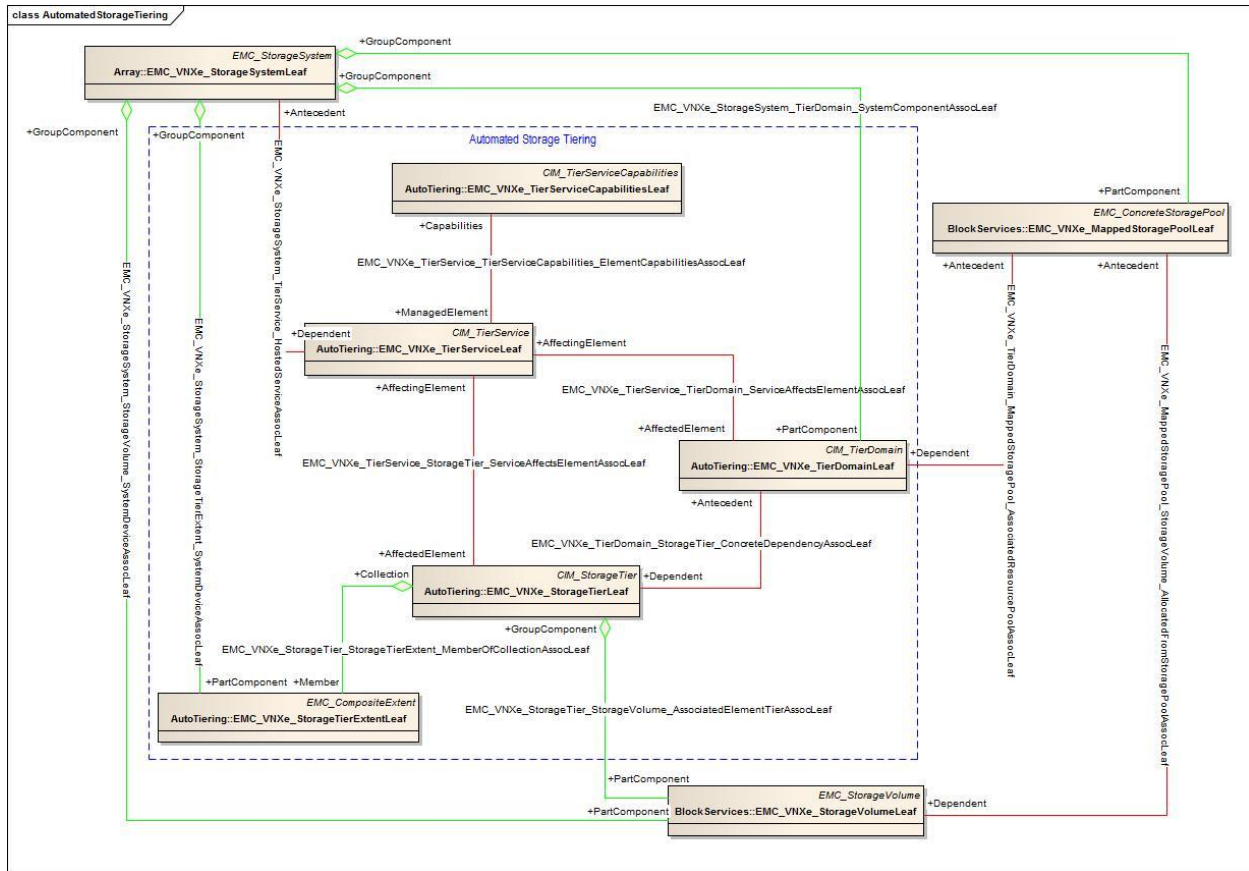


Figure 28 - Automated Storage Tying class diagram

Method of the Profile

This subprofile does not include any extrinsic methods.

Client considerations

Model Specification

In VNXe, the storage system manages the storage tiers including:

- creating the storage tiers
- monitoring the activities of the storage elements (for example, volumes)
- moving the storage elements to the appropriate storage tiers (with `StorageTier.RelativePerformanceOrder` having a value less than or equal to `StorageVolume.RelativePerformanceOrder` or one of the values in the `RelativePerformanceOrderSet`)

Use case: Discover Automated Storage Tying Capabilities

This use case describes how to discover Automated Storage Tying capabilities.

1. From the *ComputerSystem* representing the storage system (the Top-Level Computer System), traverse *CIM_HostedService* to *CIM_TierService* to obtain tiering service.
2. From *CIM_TierService*, traverse *CIM_ElementCapabilities* to *CIM_TierServiceCapabilities* to obtain tier capabilities.
3. Examine properties on *CIM_TierServiceCapabilities* to learn storage system tier capabilities.

Use case: Create StorageVolume with Storage Tiering

This use case describes how to create a new storage volume and request the newly created storage volume to be placed in an appropriate storage tier.

[Preconditions]

Client has identified the array device (*Top-level CIM_ComputerSystem*) and the *CIM_StoragePool* (the source pool from which the storage volume will be created).

[Following steps create an advanced storage setting instance]

1. From the *CIM_StoragePool*, traverse *CIM_ElementCapabilities* to *CIM_StorageCapabilities* to get the storage capabilities of the storage pool.
2. Invoke method *CreateSetting* of the *CIM_StorageCapabilities* instance obtained in step 1 to create a *CIM_AdvancedStorageSetting* instance for following storage volume creation.
3. Check property '*InitialStorageTierMethodology*' of the *CIM_AdvancedStorageSetting* instance created in step 2 to see if the value satisfying.
4. If not, invoke method *ModifyInstance* upon the *CIM_AdvancedStorageSetting* instance to change value of '*InitialStorageTierMethodology*' to required value.

[Following steps create a storage volume with storage tiering]

5. From the *Top-level ComputerSystem*, traverse *CIM_HostedService* to *CIM_StorageConfigurationService* to get the service instance.
6. Invoke method *CreateOrModifyElementFromStoragePool* by passing reference of the source pool, the advance storage setting and other required parameters.
7. Wait for the job returned from step 6 to be completed.
8. After the job completion, traverse *CIM_AffectedJobElement* to *CIM_ManagedElement* to get the newly created storage volume with specified storage tiering.

Use case: Modify Tiering Policy of StorageVolume

This use case describes how to modify the tiering policy of an existing storage volume.

[Preconditions]

Client has identified the array device (*Top-level CIM_ComputerSystem*) and the *CIM_StorageVolume* to be modified.

[Following steps create an advanced storage setting instance]

1. From *CIM_StorageVolume*, traverse *CIM_AllocatedFromStoragePool* to get the source pool instance.
2. From the *CIM_StoragePool* obtained in step 1, traverse *CIM_ElementCapabilities* to *CIM_StorageCapabilities* to get the storage capabilities of the storage pool.

3. Invoke method *CreateSetting* of the *CIM_StorageCapabilities* instance obtained in step 2 to create a *CIM_AdvancedStorageSetting* instance for following storage volume creation.
4. Check property '*InitialStorageTierMethodology*' of the *CIM_AdvancedStorageSetting* instance created in step 3 to see if the value satisfying.
5. If not, invoke method *ModifyInstance* upon the *CIM_AdvancedStorageSetting* instance to change value of '*InitialStorageTierMethodology*' to required value.

[Following steps modify tiering policy of the storage volume]

6. From the *Top-level ComputerSystem*, traverse *CIM_HostedService* to *CIM_StorageConfigurationService* to get the service instance.
7. Invoke method *CreateOrModifyElementFromStoragePool* by passing reference of the storage volume, the advance storage setting and other required parameters.
8. Wait for the job returned from step 7 to be completed.
9. After the job completion, traverse *CIM_AffectedJobElement* to *CIM_ManagedElement* to get the modified storage volume with specified storage tiering.

CIM Element

Table 233 - CIM Elements for Automated Storage Tiering

CIM Class	Implemented Class	Description
CIM_AssociatedElementTier	EMC_VNXe_StorageTier_StorageVolume_AssociatedElementTierAssocLeaf	Association between storage volume and storage tier.
CIM_AssociatedResourcePool	EMC_VNXe_TierDomain_MappedStoragePool_AssociatedResourcePoolAssocLeaf	Association between tier domain and dependent resource pool.
CIM_CompositeExtent	EMC_VNXe_StorageTierExtentLeaf	Storage extent comprises a tier.
CIM_ConcreteDependency (TierDomain to StorageTier)	EMC_VNXe_TierDomain_StorageTier_ConcreteDependencyAssocLeaf	Association between storage tiers and tier domain.
CIM_ElementCapabilities	EMC_VNXe_TierService_TierServiceCapabilities_ElementCapabilitiesAssocLeaf	Association between tier service and its capabilities.
CIM_HostedService	EMC_VNXe_StorageSystem_TierService_HostedServiceAssocLeaf	Association between tier service and system hosts the service.
CIM_MemberOfCollection (Identifies StorageExtents comprising a tier)	EMC_VNXe_StorageTier_StorageTierExtent_MemberOfCollectionAssocLeaf	Association between storage tier and storage extents comprising the tier.
CIM_ServiceAffectsElement (Between TierService and StorageTier)	EMC_VNXe_TierService_StorageTier_ServiceAffectsElementAssocLeaf	Association between tier service and tiers affected by the service.
CIM_ServiceAffectsElement (Between	EMC_VNXe_TierService_TierDomain_ServiceAffectsElementAssocLeaf	Association between tier service and tier domains affected by the

TierService and TierDomain)		service.
CIM_StorageTier	EMC_VNXe_StorageTierLeaf	This class represents a collection of storage objects identified as a storage tier.
CIM_SystemComponent (TierDomain to ComputerSystem)	EMC_VNXe_StorageSystem_TierDomain_SystemComponentAssocLeaf	Association between storage system and tier domains on the system.
CIM_TierDomain	EMC_VNXe_TierDomainLeaf	This class represents one or more storage tiers.
CIM_TierService	EMC_VNXe_TierServiceLeaf	This class provides methods to allow clients to manage storage tiers.
CIM_TierServiceCapabilities	EMC_VNXe_TierServiceCapabilitiesLeaf	This class defines the capabilities of the tier service.

EMC_VNXe_TierServiceLeaf

Table 234 - Referenced properties/methods for EMC_VNXe_TierServiceLeaf

CIM property	Description/notes
SystemCreationClassName	Set to EMC_VNXe_StorageSystemLeaf
SystemName	Set as system name of VNXe
CreationClassName	Set to EMC_VNXe_TierServiceLeaf
Name	Set to TierService

EMC_VNXe_TierServiceCapabilitiesLeaf

Table 235 - Referenced properties/methods for EMC_VNXe_TierServiceCapabilitiesLeaf

CIM property	Description/notes
InstanceID	Unique ID of the instance.
ElementName	User friendly name of the instance.
SupportedFeatures	Enumeration indicating the supported features of the storage tiering service. Set to [2: System Creates StorageTiers, 4: System Creates TierDomains, 6: Supports SubLUN]
SupportedTierFeature	Enumeration indicating the supported features of the storage tiers. Set to [2: StorageTiers Based On Performance Only]
SupportedStorageObjects	Enumeration indicating the supported storage objects that can be used to form a storage tier.

CIM property	Description/notes
	Set to [4: StorageExtent]
SupportedAsynchronousActions	Identify methods using job control. Set to NULL
SupportedSynchronousActions	Identify methods not using job control. Set to NULL

EMC_VNXe_StorageTierLeaf

Table 236 - Referenced properties/methods for EMC_VNXe_StorageTierLeaf

CIM property	Description/notes
InstanceID	Unique ID of the instance.
ElementName	User friendly name of the instance.
RelativePerformanceOrder	A number starting from 0 to indicate the relative performance characteristics of the storage tier. The smaller the number, the higher the performance characteristics.
StorageObjectType	Type of storage objects comprising a storage tier
Technology	Technology of the underlying disk drives used.
State	Whether the storage tier is actively being used or not.
Dynamic	If any newly created storage object is added to the storage tier automatically.
TotalCapacity	Total capacity of the storage tier in bytes.
DeleteOnEmptyStorageTier	If the storage tier will be deleted automatically when the storage tier becomes empty.

EMC_VNXe_TierDomainLeaf

Table 237 - Referenced properties/methods for EMC_VNXe_TierDomainLeaf

CIM property	Description/notes
CreationClassName	Name of the class. Set to EMC_VNXe_TierDomainLeaf
Name	Name of the tier domain.
ElementName	User friendly name of the tier domain.
NameFormat	Format of Name property.

CIM property	Description/notes
OtherIdentifyingInfo	Additional data to identify the instance.
IdentifyingDescriptions	Explanations and details behind the entries in OtherIdentifyingInfo.

EMC_VNXe_StorageTierExtentLeaf

Table 238 - Referenced properties/methods for EMC_VNXe_StorageTierExtentLeaf

CIM property	Description/notes
SystemCreationClassName	Set to EMC_VNXe_StorageSystemLeaf
SystemName	Set as system name of VNXe
CreationClassName	Set to EMC_VNXe_StorageTierExtentLeaf
DeviceID	ID of the extent
ElementName	User friendly name of the extent.
Name	Set to Storage Tier Extent
NameFormat	Set to 7: SNVM
NameNamespace	Set to 7: SNVM
OperationalStatus	Current operation status of the extent.
BlockSize	Set to 512
NumberOfBlocks	Block numbers of the pool extent.
ConsumableBlocks	Consumable block numbers of the extent.
Primordial	Set to FALSE
Access	Set to 3: Read/Write Supported
ExtentStatus	Set to [2: None/Not Applicable]
ExtentDiscriminator	Set to [SNIA:Pool Component, SNIA: Composite]
ExtentStripeLength	Number of StorageExtents forming the user data stripe.
DataRedundancy	Number of complete copies of data currently maintained.
PackageRedundancy	Number of physical packages can currently fail without data loss.
IsBasedOnUnderlyingRedundancy	Set to TRUE
IsConcatenated	Set to FALSE

CIM property	Description/notes
HealthState	Current health state of the extent
NoSinglePointOfFailure	Set to FALSE
IsComposite	Set to FALSE
Usage	Set to 2: Unrestricted
EnabledState	Set to 5: Not Applicable
RequestedState	Set to 12: Not Applicable
TransitioningToState	Set to 12: Not Applicable

EMC_VNXe_StorageSystem_TierService_HostedServiceAssocLeaf

**Table 239 - Referenced properties/methods for
EMC_VNXe_StorageSystem_TierService_HostedServiceAssocLeaf**

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_StorageSystemLeaf
Dependent	Reference of EMC_VNXe_TierServiceLeaf

EMC_VNXe_TierService_TierServiceCapabilities_ElementCapabilitiesAssocLeaf

**Table 240 - Referenced properties/methods for
EMC_VNXe_TierService_TierServiceCapabilities_ElementCapabilitiesAssocLeaf**

CIM property	Description/notes
ManagedElement	Reference of EMC_VNXe_TierServiceLeaf
Capabilities	Reference of EMC_VNXe_TierServiceCapabilitiesLeaf

EMC_VNXe_StorageSystem_TierDomain_SystemComponentAssocLeaf

**Table 241 - Referenced properties/methods for
EMC_VNXe_StorageSystem_TierDomain_SystemComponentAssocLeaf**

CIM property	Description/notes
GroupComponent	Reference of EMC_VNXe_StorageSystemLeaf
PartComponent	Reference of EMC_VNXe_TierDomainLeaf

EMC_VNXe_TierService_StorageTier_ServiceAffectsElementAssocLeaf

**Table 242 - Referenced properties/methods for
EMC_VNXe_TierService_StorageTier_ServiceAffectsElementAssocLeaf**

CIM property	Description/notes
AffectingElement	Reference of EMC_VNXe_TierServiceLeaf
AffectedElement	Reference of EMC_VNXe_StorageTierLeaf

EMC_VNXe_TierService_TierDomain_ServiceAffectsElementAssocLeaf

**Table 243 - Referenced properties/methods for
EMC_VNXe_TierService_TierDomain_ServiceAffectsElementAssocLeaf**

CIM property	Description/notes
AffectingElement	Reference of EMC_VNXe_TierServiceLeaf
AffectedElement	Reference of EMC_VNXe_TierDomainLeaf

EMC_VNXe_TierDomain_StorageTier_ConcreteDependencyAssocLeaf

**Table 244 - Referenced properties/methods for
EMC_VNXe_TierDomain_StorageTier_ConcreteDependencyAssocLeaf**

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_TierDomainLeaf
Dependent	Reference of EMC_VNXe_StorageTierLeaf

EMC_VNXe_StorageTier_StorageTierExtent_MemberOfCollectionAssocLeaf

**Table 245 - Referenced properties/methods for
EMC_VNXe_StorageTier_StorageTierExtent_MemberOfCollectionAssocLeaf**

CIM property	Description/notes
Collection	Reference of EMC_VNXe_StorageTierLeaf
Member	Reference of EMC_VNXe_StorageTierExtentLeaf

EMC_VNXe_StorageTier_StorageVolume_AssociatedElementTierAssocLeaf

**Table 246 - Referenced properties/methods for
EMC_VNXe_StorageTier_StorageVolume_AssociatedElementTierAssocLeaf**

CIM property	Description/notes
GroupComponent	Reference of EMC_VNXe_StorageTierLeaf
PartComponent	Reference of EMC_VNXe_StorageVolumeLeaf
Allocated	Indicates what portion the storage volume is associated with the storage tier.

EMC_VNXe_TierDomain_MappedStoragePool_AssociatedResourcePoolAssocLeaf

**Table 247 - Referenced properties/methods for
EMC_VNXe_TierDomain_MappedStoragePool_AssociatedResourcePoolAssocLeaf**

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_MappedStoragePoolLeaf
Dependent	Reference of EMC_VNXe_TierDomainLeaf

Copy Service Subprofile

Overview

The Copy Services Subprofile defines a management interface for local mirror management, local snapshot management, and clone management. In VNXe, only local snapshot management is implemented and supported.

The synchronization view is modeled with a StorageSynchronized association. A client can determine the type and state of the synchronized view by inspecting the properties of the association instance.

NOTE: For more details, refer to [Clause 9: Copy Services Subprofile in Storage Management Technical Specification, Part 3 Block Devices, Version 1.6.0, Revision 4](#).

Class diagram

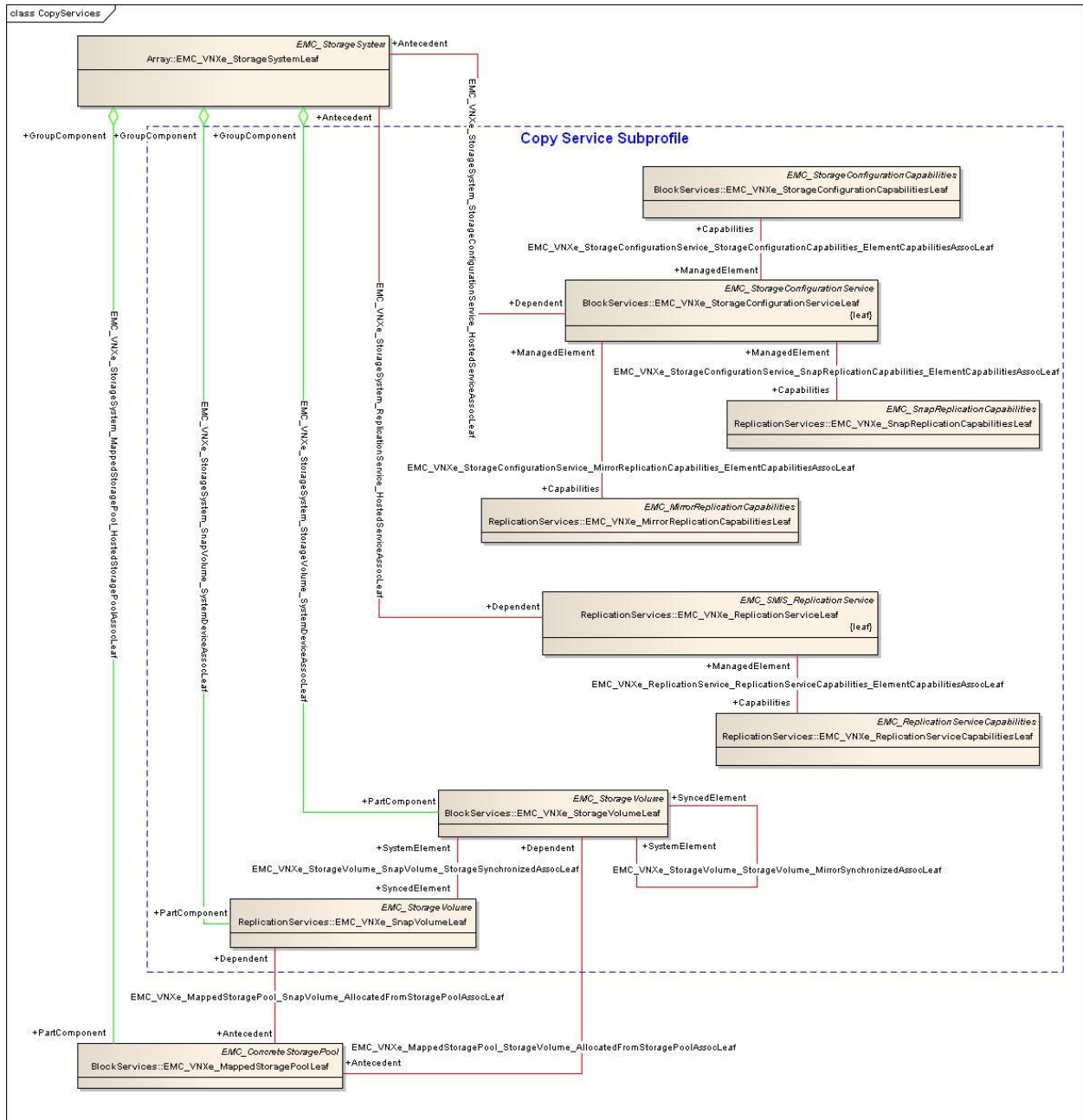


Figure 29 - Copy Services Subprofile Class Diagram

Method of the Profile

This section describes each extrinsic method supported by this profile.

Intrinsic methods on StorageConfigurationService

Table 248 - Extrinsic methods on StorageConfigurationService

Method name	Primordial Pool
-------------	-----------------

Method name	Primordial Pool
CreateReplica	Supported
ModifySynchronization	Supported
AttachReplica	Not Supported

Method: StorageConfigurationService.CreateReplica

- Description**

This method allows the client to create a single-LUN snapshot or mirror.

NOTE:

This method cannot support storage object replication well. To obtain more functionality, the client should use [Method: ReplicationService.CreateElementReplica](#).

- Parameters**

Table 249 - Signature and parameters of EMC_VNXe_StorageConfigurationServiceLeaf.CreateReplica

Parameter name	Qualifier	Type	Description/note
ElementName	IN, NULL allowed	String	An end user relevant name for the standalone-LUN snapshot being created. NOTE: <ul style="list-style-type: none"> ElementName only takes effect when creating a snapshot. If the client creates a standalone-LUN snapshot but does not provide an <i>ElementName</i>, a system default name like “SNAPVOLUME_<creation time>” will be used.
Job	OUT	CIM_ConcreteJob Ref	Reference to the job.
SourceElement	IN, Mandatory	CIM_LogicalElementRef	Reference to the source storage element to be snapped. NOTE: <ul style="list-style-type: none"> Type of <i>SourceElement</i> must be reference of <i>EMC_VNXe_StorageVolumeLeaf</i>.
TargetElement	OUT	CIM_LogicalElementRef	Reference to the created target storage element (the replica). NOTE: <ul style="list-style-type: none"> Since the method is supported in asynchronous mode, the output of this parameter will be NULL. Client should use <i>Job</i> and <i>AffectedJobElement</i>

Parameter name	Qualifier	Type	Description/note
			association to check the result.
TargetSettingGoal	IN, NULL allowed, Ignored	CIM_StorageSettingRef	Definition for the StorageSetting to be maintained by the target storage object (the replica).
TargetPool	IN, NULL allowed, Ignored	CIM_StoragePoolRef	Underlying storage for the target element (the replica) will be drawn from TargetPool if specified; otherwise the allocation is implementation specific.
CopyType	IN, Mandatory	Uint16	Type of copy that will be made. NOTE: <ul style="list-style-type: none"> Allowed value is '2: Async' (for mirror) or '4: UnSyncAssoc' (for snapshot).

- **Return Results**

Table 250 - Possible return code of EMC_VNXe_StorageConfigurationServiceLeaf.CreateReplica

Return code	Type	Description
4096	uint32	Method parameters checked, job started. All input parameters are valid, no error or exception occurs when starting the job.
4	uint32	Failed: <ul style="list-style-type: none"> Any error that occurs or exception thrown when calling the underlying component.
5	uint32	Invalid Parameter: <ul style="list-style-type: none"> ElementName is already used by other snap. CopyType is not supported. SourceElement is not valid.
NA	CIM_Error	Exceptions will be thrown when any of the below cases is met: <ul style="list-style-type: none"> Any of the required parameters is NULL. Any of input parameters is not of the right CIM/SNIA type defined in the MOF files. Any of the input references cannot be found in the system.

Method: StorageConfigurationService.ModifySynchronization

- **Description**

This method allows the client to perform LUN mirror operations.

NOTE:

This method cannot support storage object replication well. To obtain more functionality, the client should use [Method: ReplicationService.ModifyReplicaSynchronization](#).

- Parameters

Table 251 - Signature and parameters of EMC_VNXe_StorageConfigurationServiceLeaf.ModifySynchronization

Parameter name	Qualifier	Type	Description/note
Operation	IN, Mandatory	Uint16	Type of modification to be made to the replica.
Job	OUT	CIM_ConcreteJob Ref	Reference to the job.
Synchronization	IN, Mandatory	CIM_StorageSync hronizedRef	Reference to the <i>StorageSynchronized</i> association describing the storage source/replica relationship.

- Return Results

Table 252 - Possible return code of EMC_VNXe_StorageConfigurationServiceLeaf.ModifySynchronization

Return code	Type	Description
4096	uint32	Method parameters checked, job started. All input parameters are valid, no error or exception occurs when starting the job.
4	uint32	Failed: <ul style="list-style-type: none"> Any error that occurs or exception thrown when calling the underlying component.
5	uint32	Invalid Parameter: <ul style="list-style-type: none"> Operation is not supported. Synchronization type is not valid.
NA	CIM_Error	Exceptions will be thrown when any of the below cases is met: <ul style="list-style-type: none"> Any of the required parameters is NULL. Any of input parameters is not of the right CIM/SNIA type defined in the MOF files. Any of the input references cannot be found in the system.

Extrinsic methods on ReplicationService

Table 253 - Extrinsic methods on ReplicationService

Method name	Primordial Pool
-------------	-----------------

Method name	Primordial Pool
CreateElementReplica	Supported
ModifyReplicaSynchronization	Supported

Method: *ReplicationService.CreateElementReplica*

For detail information, refer to [Method: ReplicationService.CreateElementReplica](#) in Replication Services Profile.

Method: *ReplicationService.ModifyReplicaSynchronization*

For detail information, refer to [Method: ReplicationService.ModifyReplicaSynchronization](#) in Replication Services Profile.

Client considerations

Model Specification

For storage replication, clients should use the [Replication Services Profile](#) to obtain more functionality. The Copy Services profile has been deprecated and there are limitations to how well it supports storage replication.

Use case: Discover Copy support and capabilities

This use case describes how to discover system’s storage configuration capabilities.

1. From Block Server (Top-level Array ComputerSystem), traverse *CIM_HostedService* to *CIM_StorageConfigurationService* to locate hosted storage configuration service.
2. From *CIM_StorageConfigurationService*, traverse *CIM_ElementCapabilities* to *CIM_StorageConfigurationCapabilities* to get system’s storage configuration capabilities.
3. From *CIM_StorageConfigurationService*, traverse *CIM_ElementCapabilities* to *CIM_StorageReplicationCapabilities* to get system’s storage replication capabilities.

Client should consider both the *StorageConfigurationCapabilities* and the *StorageReplicationCapabilities* to learn the system’s copy capabilities.

Use case: Create Standalone-LUN Snapshot

This use case describes how to create a standalone-LUN snapshot via Copy Services.

There are two methods supported for creating a standalone-LUN snapshot:

- Through *CIM_StorageConfigurationService.CreateReplica* implemented in Copy Services.
 1. From Block Server (Top-level Array ComputerSystem), traverse *CIM_HostedService* to *CIM_StorageConfigurationService* to locate hosted storage configuration service.
 2. Invoke method *CIM_StorageConfigurationService.CreateReplica()* to create a Standalone-LUN snapshot, by passing following parameters:
 - ElementName* – User friendly name for the snapshot.
 - CopyType* – Type of the copy to be made; it should be set to 4 (UnSyncAssoc).

SourceElement – Reference of the source StorageVolume or SnapVolume .

3. Pull status of the Job output from the method to check status of the copy process.
- Through *CIM_StorageReplicationService.CreateElementReplica* implemented in Replication Services.

For detailed information, refer to [Create Individual LUN Snapshot](#) in Replication Services.

Use case: Delete Standalone-LUN Snapshot

This use case describes how to delete a standalone-LUN snapshot.

There are two methods supported for deleting a standalone-LUN snapshot:

- *CIM_StorageConfigurationService.ReturnToStoragePool* for Copy Services.
 1. From Block Server (Top-level Array ComputerSystem), traverse *CIM_HostedService* to *CIM_StorageConfigurationService* to locate hosted storage configuration service.
 2. Invoke method *CIM_StorageConfigurationService.ReturnToStoragePool()* to delete the Standalone-LUN snapshot, by passing following parameters:

TheElement – Reference of the snapshot to be deleted.
 3. Pull status of the Job output from the method to check status of the deletion process.
- *CIM_ReplicationService.ModifyReplicaSynchronization* for Replication Services.

For detailed information, refer to [Delete Individual LUN Snapshot](#) in Replication Services.

CIM Element

The implemented classes and associations related to the Copy Services Subprofile on the VNXe Storage System are described as follows:

Table 254 - CIM Elements for Copy Services Subprofile

CIM Class	Implemented Class	Description
CIM_StorageConfigurationService	EMC_VNXe_StorageConfigurationServiceLeaf	Represents Storage Configuration Service provided by the system.
CIM_ReplicationService	EMC_VNXe_ReplicationServiceLeaf	Represents Replication Service provided by the system.
CIM_StorageConfigurationCapabilities	EMC_VNXe_StorageConfigurationCapabilitiesLeaf	Represents Storage Configuration Capabilities of the system.
CIM_StorageReplicationCapabilities	EMC_VNXe_SnapReplicationCapabilitiesLeaf	Represents Storage Replication Capabilities of the system.
CIM_ReplicationServiceCapabilities	EMC_VNXe_ReplicationServiceCapabilitiesLeaf	Represents Capabilities of Replication Service provided by the system.
CIM_ElementCapabilities	EMC_VNXe_StorageConfigurationService_StorageConfigurationCapabilities_ElementCapabilitiesAssocLeaf	Associates Storage Configuration Service and Storage Configuration Capabilities of the system.

CIM Class	Implemented Class	Description
CIM_ElementCapabilities	EMC_VNXe_StorageConfigurationService_StorageReplicationCapabilities_ElementCapabilitiesAssocLeaf	Associates Storage Configuration Service and Storage Replication Capabilities of the system.
CIM_ElementCapabilities	EMC_VNXe_ReplicationService_ReplicationServiceCapabilities_ElementCapabilitiesAssocLeaf	Associates Replication Service and Capabilities of the service.
CIM_HostedService	EMC_VNXe_StorageSystem_StorageConfigurationService_HostedServiceAssocLeaf	Associates Storage Configuration Service and the system hosting the service.
CIM_HostedService	EMC_VNXe_StorageSystem_ReplicationService_HostedServiceAssocLeaf	Associates Replication Service and the system hosting the service.

EMC_VNXe_StorageConfigurationServiceLeaf

For detailed information, refer to [EMC_VNXe_StorageConfigurationServiceLeaf](#) in Block Services Package.

EMC_VNXe_ReplicationServiceLeaf

For detailed information, refer to [EMC_VNXe_ReplicationServiceLeaf](#) in Replication Services Profile.

EMC_VNXe_StorageConfigurationCapabilitiesLeaf

For detailed information, refer to [EMC_VNXe_StorageConfigurationCapabilitiesLeaf](#) in Block Services Package.

EMC_VNXe_SnapReplicationCapabilitiesLeaf

For detailed information, refer to [EMC_VNXe_SnapReplicationCapabilitiesLeaf](#) in Replication Services Profile.

EMC_VNXe_ReplicationServiceCapabilitiesLeaf

For detailed information, refer to [EMC_VNXe_ReplicationServiceCapabilitiesLeaf](#) in Replication Services Profile.

EMC_VNXe_StorageConfigurationService_StorageConfigurationCapabilities_ElementCapabilitiesAssocLeaf

Table 255 - Referenced properties/methods for EMC_VNXe_StorageConfigurationService_StorageConfigurationCapabilities_ElementCapabilitiesAssocLeaf

CIM property	Description/notes
ManagedElement	Reference of EMC_VNXe_StorageConfigurationServiceLeaf
Capabilities	Reference of EMC_VNXe_StorageConfigurationCapabilitiesLeaf

EMC_VNXe_StorageConfigurationService_StorageReplicationCapabilities_ElementCapabilitiesAssocLeaf

Table 256 - Referenced properties/methods for EMC_VNXe_StorageConfigurationService_SnapReplicationCapabilities_ElementCapabilitiesAssocLeaf

CIM property	Description/notes
ManagedElement	Reference of EMC_VNXe_StorageConfigurationServiceLeaf
Capabilities	Reference of EMC_VNXe_SnapReplicationServiceLeaf

EMC_VNXe_ReplicationService_ReplicationServiceCapabilities_ElementCapabilitiesAssocLeaf

Table 257 - Referenced properties/methods for EMC_VNXe_ReplicationService_ReplicationServiceCapabilities_ElementCapabilitiesAssocLeaf

CIM property	Description/notes
ManagedElement	Reference of EMC_VNXe_ReplicationServiceLeaf
Capabilities	Reference of EMC_VNXe_ReplicationServiceCapabilitiesLeaf

EMC_VNXe_StorageSystem_StorageConfigurationService_HostedServiceAssocLeaf

Table 258 - Referenced properties/methods for EMC_VNXe_StorageSystem_StorageConfigurationService_HostedServiceAssocLeaf

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_StorageSystemLeaf
Dependent	Reference of EMC_VNXe_StorageConfigurationServiceLeaf

EMC_VNXe_StorageSystem_ReplicationService_HostedServiceAssocLeaf

Table 259 - Referenced properties/methods for EMC_VNXe_StorageSystem_ReplicationService_HostedServiceAssocLeaf

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_StorageSystemLeaf
Dependent	Reference of EMC_VNXe_ReplicationServiceLeaf

Replication Services Profile

Overview

Replication Services, a *component* profile, specifies attributes and methods to copy data from a source element to a target element.

The Replication Services Profile extends the functionality of the Copy Services Subprofile by including enhanced local replication for thinly provisioned storage objects and support for replication groups and consistency groups.

NOTE: For more details, refer to *Clause 26: Replication Services Profile in Storage Management Technical Specification, Part 3 Block Devices, Version 1.6.0, Revision 4.*

Class diagram

Services and Capabilities Discovery

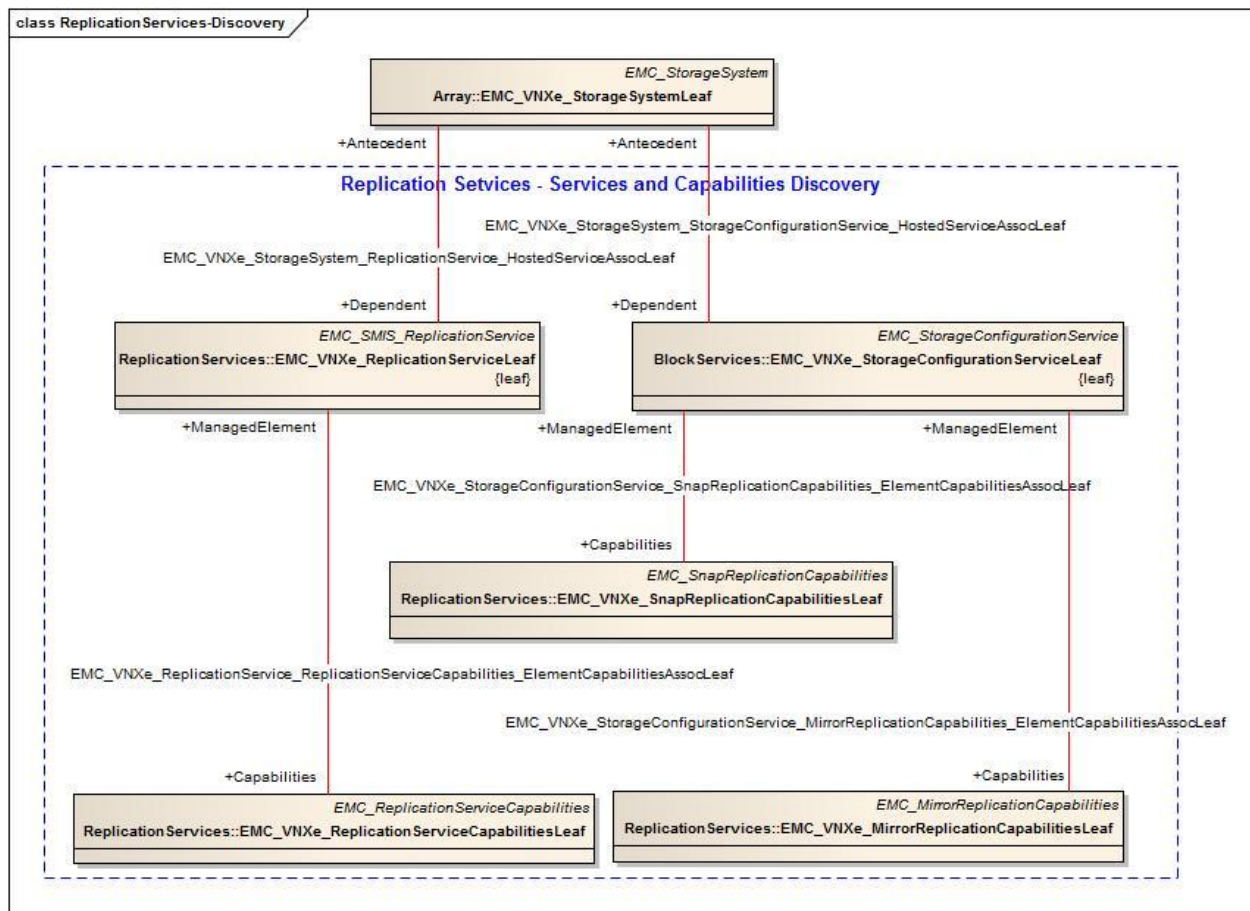


Figure 30 - Replication Services and Capabilities Discovery

Element and Group Replication

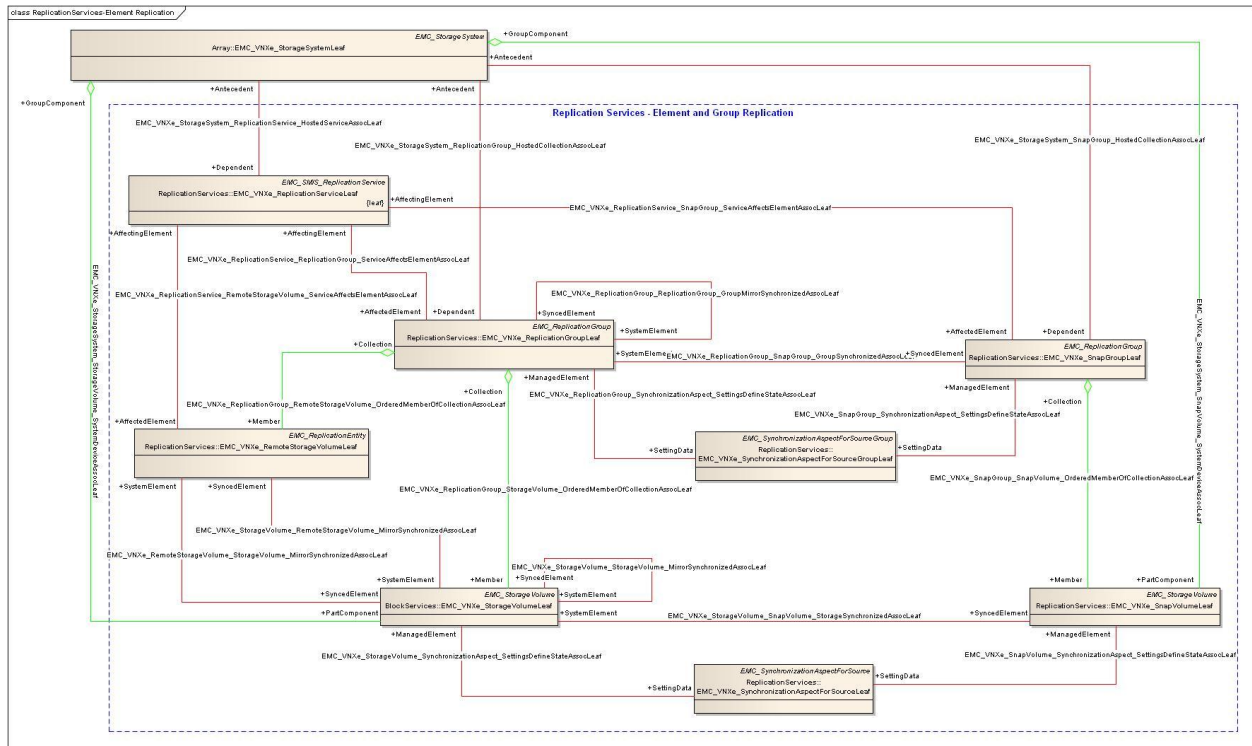


Figure 31 - Element and Group Replication

Using StoragePool for Replicas

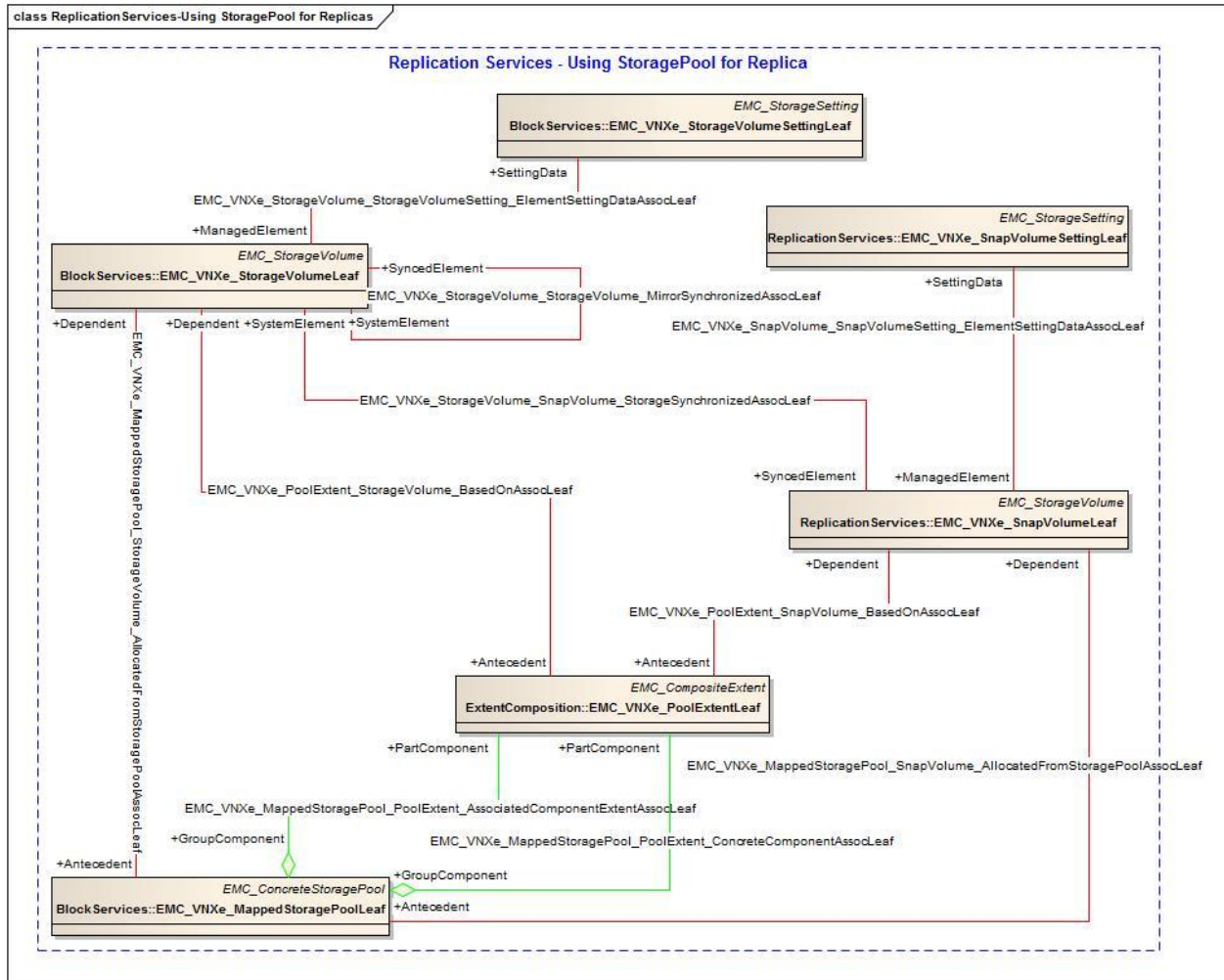


Figure 32 - Using StoragePool for Replicas

Access and Protocol Endpoint

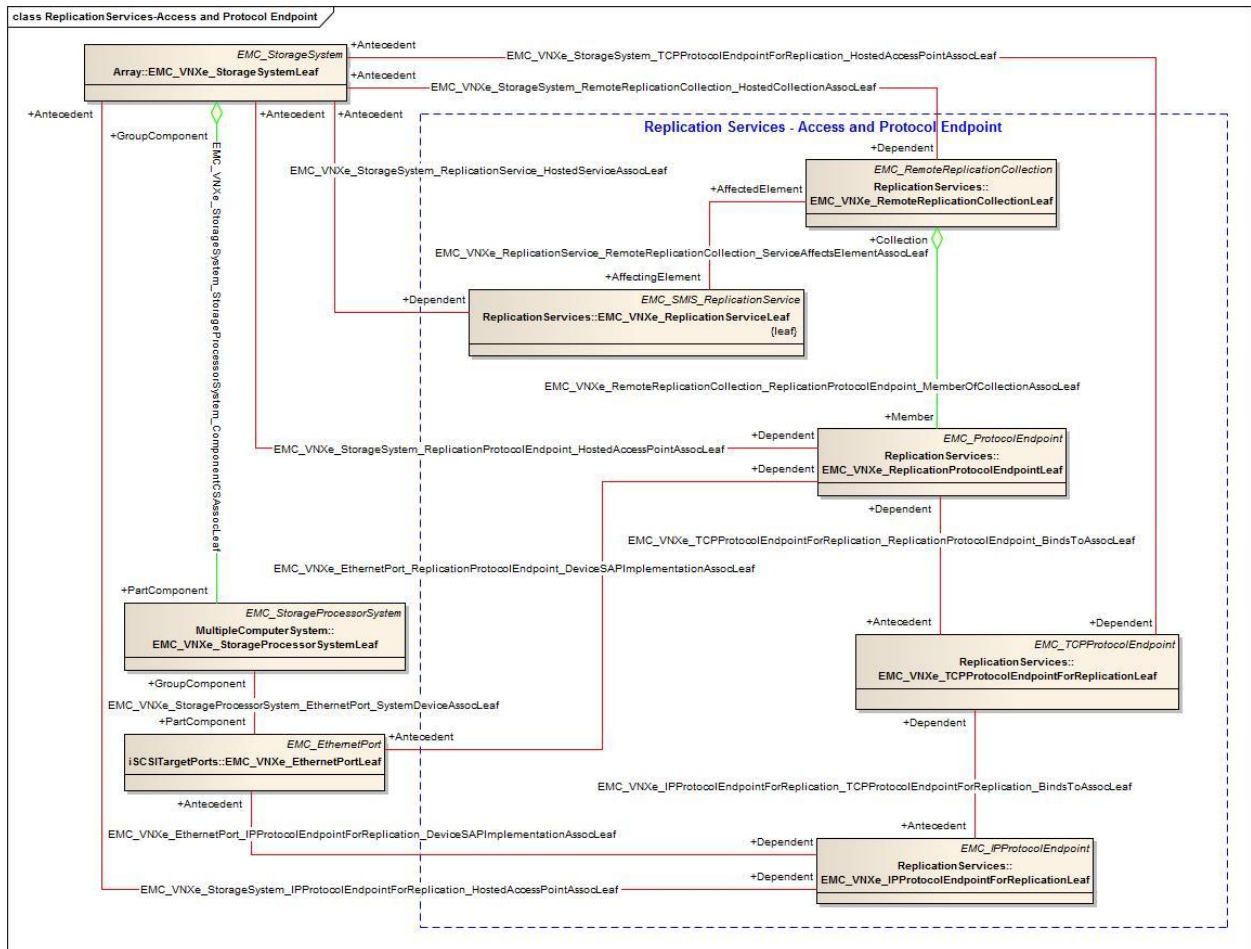


Figure 33 - Replication Services Access and Protocol Endpoint

Method of the Profile

Extrinsic Methods on Group Management

Method: *ReplicationService.CreateGroup*

- **Description**

This method allows the client to create a new consistency replication group.

- **Parameters**

Table 260 - Signature and parameters of `EMC_VNXe_ReplicationServiceLeaf.CreateGroup`

Parameter name	Qualifier	Type	Description/note
GroupName	IN, NULL allowed	String	Name of the replication group to be created. NOTE: <ul style="list-style-type: none"> • If <i>ElementName</i> is not provided, a system default name like “LUNGROUP_<creation time>” will be

Parameter name	Qualifier	Type	Description/note
			used.
Members	IN, NULL allowed	CIM_LogicalElementRef []	List of elements to add to the group -- order is maintained. NOTE: <ul style="list-style-type: none"> If NULL, the group will be empty. Type of elements must be <i>EMC_VNXe_StorageVolumeLeaf</i>.
Persistent	IN, NULL allowed, Ignored	Boolean	
DeleteOnEmptyElement	IN, NULL allowed	Boolean	Whether to delete the group when the last element is removed from the group.
DeleteOnUnassociated	IN, NULL allowed, Ignored	Boolean	
ServiceAccessPoint	IN, NULL allowed, Ignored	CIM_ServiceAccessPointRef	
ReplicationGroup	OUT	CIM_ReplicationGroupRef	Reference to the created group.
ReplicationSettingData	IN, Embedded, NULL allowed	CIM_ReplicationSettingDataRef	Provides additional replication settings for the method. NOTE: <ul style="list-style-type: none"> Only 'Description' is relevant in the current implementation.

- Return Results**

Table 261 - Possible return code of EMC_VNXe_ReplicationServiceLeaf.CreateGroup

Return code	Type	Description
0	uint32	Success
4	uint32	Failed: <ul style="list-style-type: none"> Any error that occurs or exception thrown when calling the underlying component.
5	uint32	Invalid Parameter:

Return code	Type	Description
		<ul style="list-style-type: none"> Type of StorageElement in Members is not type of EMC_VNXe_StorageVolumeLeaf.
6	Uint32	In Use <ul style="list-style-type: none"> One or more elements in Members are not a standalone-LUN. (Client cannot add LUNs already in group into a new group. Client can only add a standalone-LUN into a group)
NA	CIM_Error	Exceptions will be thrown when any of the below cases is met: <ul style="list-style-type: none"> Any of the required parameters is NULL. Any of input parameters is not of the right CIM/SNIA type defined in the MOF files. Any of the input references cannot be found in the system.

Method: ReplicationService.DeleteGroup

- Description**

This method allows the client to delete a consistency replication group.

- Parameters**

Table 262 - Signature and Parameters of EMC_VNXe_ReplicationServiceLeaf.DeleteGroup

Parameter name	Qualifier	Type	Description/note
ReplicationGroup	IN, Mandatory	CIM_ReplicationGroupRef	Reference to the replication group to be deleted.
ServiceAccessPoint	IN, NULL allowed, Ignored	CIM_ServiceAccessPointRef	
RemoveElements	IN, NULL allowed	Boolean	If true, delete the group even if it is not empty.
ReplicationSettingData	IN, Embedded, NULL allowed, Ignored	CIM_ReplicationSettingDataRef	

- Return Results**

Table 263 - Possible return code of EMC_VNXe_ReplicationServiceLeaf.DeleteGroup

Return code	Type	Description
0	uint32	Success

Return code	Type	Description
4	uint32	Failed: <ul style="list-style-type: none"> Any error that occurs or exception thrown when calling the underlying component.
5	uint32	Invalid Parameter: <ul style="list-style-type: none"> ReplicationGroup is not empty but RemoveElements is not set to true explicitly.
NA	CIM_Error	Exceptions will be thrown when any of the below cases is met: <ul style="list-style-type: none"> Any of the required parameters is NULL. Any of input parameters is not of the right CIM/SNIA type defined in the MOF files. Any of the input references cannot be found in the system.

Method: ReplicationService.AddMembers

- Description**

This method allows the client to add members to an existing consistency replication group.

- Parameters**

Table 264 - Signature and Parameters of EMC_VNXe_ReplicationServiceLeaf.AddMembers

Parameter name	Qualifier	Type	Description/note
Members	IN, Mandatory	CIM_LogicalElementRef []	List of elements to be added to the group in the order supplied.
ReplicationGroup	IN, Mandatory	CIM_ReplicationGroupRef	Reference to the replication group where the members to be added.
ServiceAccessPoint	IN, NULL allowed, Ignored	CIM_ServiceAccessPointRef	
ReplicationSettingData	IN, NULL allowed, Embedded, Ignored	CIM_ReplicationSettingDataRef	

- Return Results**

Table 265 - Possible return code of EMC_VNXe_ReplicationServiceLeaf.AddMembers

Return code	Type	Description
0	uint32	Success

Return code	Type	Description
4	uint32	Failed: <ul style="list-style-type: none"> Any error that occurs or exception thrown when calling the underlying component.
5	uint32	Invalid Parameter: <ul style="list-style-type: none"> One or more elements in Members are not in type of <i>EMC_VNXe_StorageVolumeLeaf</i>. One or more elements in Members are not standalone-LUN. (Client cannot add LUNs already in group into a new group. Only standalone-LUN can be added into a group).
NA	CIM_Error	Exceptions will be thrown when any of the below cases is met: <ul style="list-style-type: none"> Any of the required parameters is NULL. Any of input parameters is not of the right CIM/SNIA type defined in the MOF files. Any of the input references cannot be found in the system.

Method: ReplicationService.RemoveMembers

- Description**

This method allows the client to remove members from an existing consistency replication group.

- Parameters**

Table 266 - Signature and Parameters of EMC_VNXe_ReplicationServiceLeaf.RemoveMembers

Parameter name	Qualifier	Type	Description/note
Members	IN, Mandatory	CIM_LogicalElementRef []	List of elements to be removed from the group.
DeleteOnEmptyElement	IN, NULL allowed	Boolean	If true and removal of the members causes the group to become empty, the group will be deleted.
ReplicationGroup	IN, Mandatory	CIM_ReplicationGroupRef	Reference to the replication group where the members are to be removed.
ServiceAccessPoint	IN, NULL allowed, Ignored	CIM_ServiceAccessPointRef	
ReplicationSettingData	IN, NULL allowed, Embedded, Ignored	CIM_ReplicationSettingDataRef	

- **Return Results**

Table 267 - Possible return code of EMC_VNXe_ReplicationServiceLeaf.RemoveMembers

Return code	Type	Description
0	uint32	Success
4	uint32	Failed: <ul style="list-style-type: none"> • Any error that occurs or exception thrown when calling the underlying component.
5	uint32	Invalid Parameter: <ul style="list-style-type: none"> • One or more elements in Members are not in type of <i>EMC_VNXe_StorageVolumeLeaf</i>. • One or more elements in Members don't belong to the group.
NA	CIM_Error	Exceptions will be thrown when any of the below cases is met: <ul style="list-style-type: none"> • Any of the required parameters is NULL. • Any of input parameters is not of the right CIM/SNIA type defined in the MOF files. • Any of the input references cannot be found in the system.

Extrinsic Methods on Replication Management

Method: ReplicationService.GetPeerSystems

- **Description**

This method allows the client get all of the peer systems for replication.

- **Parameters**

Table 268 - Signature and Parameters of EMC_VNXe_ReplicationServiceLeaf.GetPeerSystems

Parameter name	Qualifier	Type	Description/note
Options	IN, NULL allowed	uint16	Specifies which peer systems to return: all known systems or those are currently reachable.
Job	OUT	CIM_ConcreteJob Ref	Reference to the job. NOTE: <ul style="list-style-type: none"> • Since the method is supported in synchronous mode, the output of this parameter will be NULL.
Systems	OUT	CIM_ComputerSy stem []	Reference to the list of peer ComputerSystems.

- **Return Results**

Table 269 - Possible return code of EMC_VNXe_ReplicationServiceLeaf.GetPeerSystems

Return code	Type	Description
0	uint32	Success
4	uint32	Failed: <ul style="list-style-type: none"> Any error that occurs or exception thrown when calling the underlying component.

Method: ReplicationService.GetServiceAccessPoints

- Description**

This method allows the client get all ServiceAccessPoints of a peer system.

- Parameters**

Table 270 - Signature and Parameters of EMC_VNXe_ReplicationServiceLeaf.GetServiceAccessPoints

Parameter name	Qualifier	Type	Description/note
System	IN, Mandatory	CIM_ComputerSystemRef	Specifies the peer system.
Job	OUT	CIM_ConcreteJobRef	Reference to the job. NOTE: <ul style="list-style-type: none"> Since the method is supported in synchronous mode, the output of this parameter will be NULL.
ServiceAccessPoints	OUT	CIM_ServiceAccessPointRef []	Reference to the list of ServiceAccessPoints for the supplied system.

- Return Results**

Table 271 - Possible return code of EMC_VNXe_ReplicationServiceLeaf.GetServiceAccessPoints

Return code	Type	Description
0	uint32	Success
4	uint32	Failed: <ul style="list-style-type: none"> Any error that occurs or exception thrown when calling the underlying component.
5	uint32	Invalid Parameter: <ul style="list-style-type: none"> <i>System</i> is not valid.

Method: ReplicationService.GetReplicationRelationshipInstances

- Description**

This method allows the client get all of the synchronization relationships known to the processing replication service.

- **Parameters**

**Table 272 - Signature and Parameters of
EMC_VNXe_ReplicationServiceLeaf.GetReplicationRelationshipInstances**

Parameter name	Qualifier	Type	Description/note
Type	IN, NULL allowed	uint16	This parameter specifies the type of synchronization relationships, StorageSynchronized or GroupSynchronized
SyncType	IN, NULL allowed	uint16	This parameter specifies the desired synchronization type.
Mode	IN, NULL allowed	uint16	This parameter specifies the desired synchronization mode,
Locality	IN, NULL allowed	uint16	This parameter specifies the desired locality.
CopyState	IN, NULL allowed	uint16	This parameter specifies whether only to retrieve synchronization relationships that are currently in this CopyState.
Job	OUT	CIM_ConcreteJob Ref	Reference to the job. NOTE: <ul style="list-style-type: none"> • Since the method is supported in synchronous mode, the output of this parameter will be NULL.
Synchronizations	OUT, Embedded	CIM_ServiceAccessPointRef []	Reference to the list of Synchronizations found.

- **Return Results**

**Table 273 - Possible return code of
EMC_VNXe_ReplicationServiceLeaf.GetReplicationRelationshipInstances**

Return code	Type	Description
0	uint32	Success
4	uint32	Failed: <ul style="list-style-type: none"> • Any error that occurs or exception thrown when calling the underlying component.

Method: ReplicationService.GetReplicationRelationships

- **Description**

This method allows the client get all of the synchronization relationships known to the processing replication service.

- **Parameters**

**Table 274 - Signature and Parameters of
EMC_VNXe_ReplicationServiceLeaf.GetReplicationRelationships**

Parameter name	Qualifier	Type	Description/note
Type	IN, NULL allowed	uint16	This parameter specifies the type of synchronization relationships, StorageSynchronized or GroupSynchronized
SyncType	IN, NULL allowed	uint16	This parameter specifies the desired synchronization type.
Mode	IN, NULL allowed	uint16	This parameter specifies the desired synchronization mode,
Locality	IN, NULL allowed	uint16	This parameter specifies the desired locality.
CopyState	IN, NULL allowed	uint16	This parameter specifies whether only to retrieve synchronization relationships that are currently in this CopyState.
Job	OUT	CIM_ConcreteJob Ref	Reference to the job. NOTE: <ul style="list-style-type: none"> • Since the method is supported in synchronous mode, the output of this parameter will be NULL.
Synchronizations	OUT	CIM_ServiceAccessPointRef []	Reference to the list of Synchronizations found.

- **Return Results**

**Table 275 - Possible return code of
EMC_VNXe_ReplicationServiceLeaf.GetReplicationRelationships**

Return code	Type	Description
0	uint32	Success
4	uint32	Failed: <ul style="list-style-type: none"> • Any error that occurs or exception thrown when calling the underlying component.

Method: ReplicationService.GetAvailableTargetElements

- **Description**

This method allows the client get all of the candidate target elements for the supplied source element.

- Parameters

**Table 276 - Signature and Parameters of
EMC_VNXe_ReplicationServiceLeaf.GetAvailableTargetElements**

Parameter name	Qualifier	Type	Description/note
SourceElement	IN, Mandatory	CIM_LogicalElementRef	The source storage object. NOTE: <ul style="list-style-type: none"> It must be a StorageVolume.
SyncType	IN, Mandatory	uint16	This parameter specifies the type of copy. NOTE: <ul style="list-style-type: none"> It must be set to '6: Mirror'.
Mode	IN, NULL allowed	uint16	This parameter specifies the synchronization mode, NOTE: <ul style="list-style-type: none"> If provided, it must be set to '3: Asynchronous'.
ReplicationSettingData	IN, NULL allowed, Embedded, Ignored	uint16	
TargetAccessPoint	IN, NULL allowed	CIM_ServiceAccessPointRef	
TargetSettingGoal	IN, NULL allowed, Ignored	CIM_SettingDataRef	
TargetPools	IN, NULL allowed, Ignored	CIM_ResourcePoolRef	
Job	OUT	CIM_ConcreteJobRef	Reference to the job. NOTE: <ul style="list-style-type: none"> Since the method is supported in synchronous mode, the output of this parameter will be NULL.
Candidates	OUT	CIM_LogicalElementRef []	Reference to the List of of the candidate target elements.

- Return Results

**Table 277 - Possible return code of
EMC_VNXe_ReplicationServiceLeaf.GetAvailableTargetElements**

Return code	Type	Description
0	uint32	Success
4	uint32	Failed: <ul style="list-style-type: none"> Any error that occurs or exception thrown when calling the underlying component.
5	uint32	Invalid Parameter.

Method: ReplicationService.CreateElementReplica

- Description**

This method allows the client to start a job to create a new storage object which is a replica of the specified source storage object (SourceElement).

- Parameters**

Table 278 - Signature and Parameters of EMC_VNXe_ReplicationServiceLeaf.CreateElementReplica

Parameter name	Qualifier	Type	Description/note
ElementName	IN, NULL allowed	string	An end user relevant name for the replica being created. NOTE: <ul style="list-style-type: none"> If the client creates a replica but does not provide an <i>ElementName</i>, a system default name like “LUNSNAP_<creation time>” will be used.
SyncType	IN, Mandatory	Boolean	Type of copy that will be made. NOTE: <ul style="list-style-type: none"> <i>SyncType</i> must be set to '6: Mirror' or '7: Snapshot'.
Mode	IN, NULL allowed	uint16	Whether the target elements will be updated synchronously or asynchronously. NOTE: <ul style="list-style-type: none"> If not NULL, <i>Mode</i> must be set to '3: Asynchronous'
SourceElement	IN, Mandatory	CIM_LogicalElementRef	Reference to the source storage element to be replicated. NOTE: <ul style="list-style-type: none"> Type of <i>SourceElement</i> must be reference of <i>EMC_VNXe_StorageVolumeLeaf</i>.
SourceAccessPoint	IN, NULL allowed,	CIM_ServiceAccessPointRef	

Parameter name	Qualifier	Type	Description/note
	Ignored		
TargetElement	IN, OUT, NULL allowed	CIM_LogicalElementRef	Reference to the created target storage element (the replica). NOTE: <ul style="list-style-type: none"> If SyncType is set to '6: Mirror', this parameter is mandatory as an input. If SyncType is set to '7: Snapshot', this parameter is ignored as an input. Since the method is supported in asynchronous mode, the output of this parameter will be NULL. Use <i>Job</i> and <i>AffectedJobElement</i> association to check the result.
TargetAccessPoint	IN, NULL allowed	CIM_ServiceAccessPointRef	Reference to target access point information.
ReplicationSettingData	IN, NULL allowed, Embedded	CIM_ReplicationSettingDataRef	Setting data for the given SyncType. NOTE: <ul style="list-style-type: none"> Client is only supported to set 'DeltaUpdateInterval' for mirror replication.
Job	OUT	CIM_ConcreteJobRef	Reference to the job.
Synchronization	OUT	CIM_SynchronizeRef	Reference to the created association between the source and the target element. NOTE: <ul style="list-style-type: none"> Since the method is supported in asynchronous mode, the output of this parameter will be NULL. The association will be formed only after the job is completed.
TargetSettingGoal	IN, NULL allowed, Ignored	CIM_SettingDataRef	
TargetPool	IN, NULL allowed, Ignored	CIM_ResourcePoolRef	
WaitForCopyState	IN, NULL allowed,	uint16	

Parameter name	Qualifier	Type	Description/note
	Ignored		

- **Return Results**

Table 279 - Possible return code of EMC_VNXe_ReplicationServiceLeaf.CreateElementReplica

Return code	Type	Description
0	uint32	Success
4	uint32	Failed: <ul style="list-style-type: none"> • Any error that occurs or exception thrown when calling the underlying component.
5	uint32	Invalid Parameter: <ul style="list-style-type: none"> • <i>ElementName</i> is already used by other replica. • <i>SyncType</i> is not supported. • <i>Mode</i> is not supported. • <i>SourceElement</i> is not valid. • <i>TargetAccessPoint</i> is not valid. • <i>ReplicationSettingData</i> is not valid.
NA	CIM_Error	Exceptions will be thrown when any of the below cases is met: <ul style="list-style-type: none"> • Any of the required parameters is NULL. • Any of input parameters is not of the right CIM/SNIA type defined in the MOF files. • Any of the input references cannot be found in the system.

Method: ReplicationService.CreateGroupReplica

This method allows the client to start a job to create a new group of storage objects which are replicas of the specified source storage or a group of source storage objects (SourceElement).

- **Parameters**

Table 280 - Signature and Parameters of EMC_VNXe_ReplicationServiceLeaf.CreateGroupReplica

Parameter name	Qualifier	Type	Description/note
RelationshipName	IN, NULL allowed	string	An end user relevant name for the replica being created. NOTE: <ul style="list-style-type: none"> • If the client creates a replica but does not provide a <i>RelationshipName</i>, a system default name like

Parameter name	Qualifier	Type	Description/note
			"SNAPGROUP_<creation time>" will be used.
SyncType	IN, Mandatory	Boolean	Type of copy that will be made. NOTE: <ul style="list-style-type: none"> SyncType must be set to '6: Mirror' or '7: Snapshot'.
Mode	IN, NULL allowed	uint16	Whether the target elements will be updated synchronously or asynchronously. NOTE: <ul style="list-style-type: none"> If not NULL, Mode must be set to '3: Asynchronous'
SourceGroup	IN, Mandatory	CIM_ReplicationGroupRef	Reference to the group of source storage elements to be replicated. NOTE: <ul style="list-style-type: none"> Type of SourceGroup must be reference of EMC_VNXe_ReplicationGroupLeaf.
SourceElement	IN, NULL allowed, Ignored	CIM_LogicalElementRef	
SourceAccessPoint	IN, NULL allowed, Ignored	CIM_ServiceAccessPointRef	
TargetGroup	OUT	CIM_ReplicationGroupRef	Reference to the created group (the replicas). NOTE: <ul style="list-style-type: none"> If SyncType is set to '6: Mirror', this parameter is mandatory as an input. If SyncType is set to '7: Snapshot', this parameter is ignored as an input. Since the method is supported in asynchronous mode, the output of this parameter will be NULL. Client should use Job and AffectedJobElement association to check the result.
TargetElementCount	IN, NULL allowed, Ignored	CIM_LogicalElementRef	
TargetAccessPoint	IN,	CIM_ServiceAccessPointRef	Reference to target access point information.

Parameter name	Qualifier	Type	Description/note
	NULL allowed	ssPointRef	
Consistency	IN, NULL allowed, Ignored	uint16	
ReplicationSettingData	IN, NULL allowed, Embedded	CIM_ReplicationSettingDataRef	Setting data for the given SyncType. NOTE: <ul style="list-style-type: none"> For snapshot, client is only supported to set 'ElementName'. This will be name of the snap group. For mirror replication, client is only supported to set 'DeltaUpdateInterval'.
Job	OUT	CIM_ConcreteJobRef	Reference to the job.
Synchronization	OUT	CIM_SynchronizeRef	Reference to the created association between the source and the target group. NOTE: <ul style="list-style-type: none"> Since the method is supported in asynchronous mode, the output of this parameter will be NULL. The association will be formed only after the job is completed.
TargetSettingGoal	IN, NULL allowed, Ignored	CIM_SettingDataRef	
TargetPool	IN, NULL allowed, Ignored	CIM_ResourcePoolRef	
WaitForCopyState	IN, NULL allowed, Ignored	uint16	

- Return Results**

Table 281 - Possible return code of EMC_VNXe_ReplicationServiceLeaf.CreateGroupReplica

Return code	Type	Description
4096	uint32	Method parameters checked, job started. All input parameters are valid, no error or exception occurs when starting the job.

Return code	Type	Description
4	uint32	Failed: <ul style="list-style-type: none"> Any error that occurs or exception thrown when calling the underlying component.
5	uint32	Invalid Parameter: <ul style="list-style-type: none"> <i>RelationshipName</i> is already used by other replica. <i>SyncType</i> is not supported. <i>Mode</i> is not supported. <i>SourceGroup</i> is not valid. <i>TargetAccessPoint</i> is not valid. <i>ReplicationSettingData</i> is not valid.
NA	CIM_Error	Exceptions will be thrown when any of the below cases is met: <ul style="list-style-type: none"> Any of the required parameters is NULL. Any of input parameters is not of the right CIM/SNIA type defined in the MOF files. Any of the input references cannot be found in the system.

Method: *ReplicationService.ModifyReplicaSynchronization*

- Description**

This method allows the client to start a job to modify the synchronization association between two storage objects or replication groups.

- Parameters**

Table 282 - Signature and Parameters of EMC_VNXe_ReplicationServiceLeaf.ModifyReplicaSynchronization

Parameter name	Qualifier	Type	Description/note
Operation	IN, Mandatory	Uint16	Type of modification to be made to the replica.
Synchronization	IN, Mandatory	CIM_StorageSynchronizedRef	Reference to the <i>StorageSynchronized</i> association describing the storage source/replica relationship.
ReplicationSettingData	IN, NULL allowed, Embedded,	CIM_ReplicationSettingDataRef	

Parameter name	Qualifier	Type	Description/note
	Ignored		
SyncPair	IN, NULL allowed, Ignored	CIM_StorageSynchronizedRef []	
Job	OUT	CIM_ConcreteJobRef	Reference to the job.
SettingsState	OUT	CIM_SettingsDefinitionStateRef	Reference to the association between the source element and an instance of <i>SynchronizationAspect</i> . NOTE: <ul style="list-style-type: none"> Output of this parameter will be NULL.
Force	IN, NULL allowed, Ignored	boolean	
WaitForCopyState	IN, NULL allowed, Ignored	uint16	

- Return Results

**Table 283 - Possible return code of
EMC_VNXe_ReplicationServiceLeaf.ModifyReplicaSynchronization**

Return code	Type	Description
4096	uint32	Method parameters checked, job started. All input parameters are valid, no error or exception occurs when starting the job.
4	uint32	Failed: <ul style="list-style-type: none"> Any error that occurs or exception thrown when calling the underlying component.
5	uint32	Invalid Parameter: <ul style="list-style-type: none"> Operation is not supported. Synchronization type is not valid.
NA	CIM_Error	Exceptions will be thrown when any of the below cases is met: <ul style="list-style-type: none"> Any of the required parameters is NULL. Any of input parameters is not of the right CIM/SNIA type defined in the MOF files. Any of the input references cannot be found in the system.

Method: ReplicationService.CreateSynchronizationAspect

- **Description**

This method allows the client to start a job to create a new point-in-time representation (unpromoted snapshot) of a source element.

- **Parameters**

Table 284 - Signature and Parameters of EMC_VNXe_ReplicationServiceLeaf.CreateSynchronizationAspect

Parameter name	Qualifier	Type	Description/note
Name	IN, Mandatory	string	An end user relevant name for the replica being created. NOTE: <ul style="list-style-type: none"> • If the client creates a replica but does not provide an <i>ElementName</i>, a system default name like “LUNSNAP_<creation time>” or “SNAPGROUP_<creation time>” will be used
SyncType	IN, Mandatory	uint16	This parameter specifies the type of copy. NOTE: <ul style="list-style-type: none"> • It must be set to ‘7: Snapshot’.
Mode	IN, NULL allowed	uint16	This parameter specifies the synchronization mode, NOTE: <ul style="list-style-type: none"> • If provided, it must be set to ‘3: Asynchronous’.
SourceGroup	IN, NULL allowed	CIM_ReplicationGroupRef	A group of source storage objects.
SourceElement	IN, NULL allowed	CIM_LogicalElementRef	The source storage object.
SourceAccessPoint	IN, NULL allowed, Ignored	CIM_ServiceAccessPointRef	
Consistency	IN, NULL allowed, Ignored	uint16	
ReplicationSettingData	IN, NULL allowed, Embedded, Ignored	CIM_ReplicationSettingDataRef	

Parameter name	Qualifier	Type	Description/note
Job	OUT	CIM_ConcreteJob Ref	Reference to the job.
SettingsState	OUT	CIM_SettingsDefineStateRef	Reference to the List of of the candidate target elements. NOTE: <ul style="list-style-type: none"> Since the method is supported in asynchronous mode, the output of this parameter will be NULL

- Return Results**

**Table 285 - Possible return code of
EMC_VNXe_ReplicationServiceLeaf.CreateSynchronizationAspect**

Return code	Type	Description
0	uint32	Success
4	uint32	Failed: <ul style="list-style-type: none"> Any error that occurs or exception thrown when calling the underlying component.
5	uint32	Invalid Parameter.
NA	CIM_Error	Exceptions will be thrown when any of the below cases is met: <ul style="list-style-type: none"> Any of the required parameters is NULL. Any of input parameters is not of the right CIM/SNIA type defined in the MOF files. Any of the input references cannot be found in the system.

Method: ReplicationService.ModifySettingsDefineState

- Description**

This method allows the client to start a job to modify the SettingsDefineState association between the storage objects and SynchronizationAspect.

- Parameters**

**Table 286 - Signature and Parameters of
EMC_VNXe_ReplicationServiceLeaf.ModifySettingsDefineState**

Parameter name	Qualifier	Type	Description/note
Operation	IN, Mandatory	Uint16	Type of modification to be made to the replica.
SettingsState	IN, Mandatory	CIM_SettingsDefineStateRef	Reference to the association between the source element and an instance of SynchronizationAspect.

Parameter name	Qualifier	Type	Description/note
TargetElement	IN, NULL allowed, Ignored	CIM_LogicalElementRef	
TargetGroup	IN, NULL allowed, Ignored	CIM_ReplicationGroupRef	
TargetElementCount	IN, NULL allowed, Ignored	uint64	
TargetAccessPoint	IN, NULL allowed, Ignored	CIM_ServiceAccessPointRef	
Synchronization	OUT	CIM_SynchronizationRef	The reference to the created replication association describing the elements/groups relationship. NOTE: <ul style="list-style-type: none"> Output of this parameter will be NULL.
ReplicationSettingData	IN, NULL allowed, Embedded, Ignored	CIM_ReplicationSettingDataRef	
Job	OUT	CIM_ConcreteJobRef	Reference to the job.
TargetSettingGoal	IN, NULL allowed, Ignored	CIM_SettingsDefineStateRef	
TargetPool	IN, NULL allowed, Ignored	CIM_ResourcePoolRef	
WaitForCopyState	IN, NULL allowed, Ignored	uint16	

- **Return Results**

Table 287 - Possible return code of EMC_VNXe_ReplicationServiceLeaf.ModifySettingsDefineState

Return code	Type	Description
4096	uint32	Method parameters checked, job started. All input parameters are valid, no error or exception occurs when starting the job.
4	uint32	Failed: <ul style="list-style-type: none"> Any error that occurs or exception thrown when calling the underlying component.
5	uint32	Invalid Parameter
NA	CIM_Error	Exceptions will be thrown when any of the below cases is met: <ul style="list-style-type: none"> Any of the required parameters is NULL. Any of input parameters is not of the right CIM/SNIA type defined in the MOF files. Any of the input references cannot be found in the system.

Extrinsic Methods on ReplicationServiceCapabilities

Method: *ReplicationServiceCapabilities.ConvertReplicationTypeToSyncType*

- Description**

This method is used to translate *ReplicationType* to the corresponding *SyncType*, *Mode*, *Local/Remote*.

- Parameters**

**Table 288 - Signature and Parameters of
EMC_VNXe_ReplicationServiceCapabilities.ConvertReplicationTypeToSyncType**

Parameter name	Qualifier	Type	Description/note
ReplicationType	IN, Mandatory	uint16	Value representing the ReplicationType.
SyncType	OUT	uint16	SyncType describes the type of copy.
Mode	OUT	uint16	Mode describes whether the target elements will be updated synchronously or asynchronously.
LocalOrRemote	OUT	uint16	Copy to local or remote replica.

- Return Results**

**Table 289 - Possible return code of
EMC_VNXe_ReplicationServiceCapabilities.ConvertReplicationTypeToSyncType**

Return code	Type	Description
0	uint32	Method completed OK

Return code	Type	Description
5	uint32	Invalid Parameter

Method: *ReplicationServiceCapabilities.ConvertSyncTypeToReplicationType*

- **Description**

This method is used to accept the supplied information and return the corresponding *ReplicationType*, which can be passed to other methods to get the additional capabilities.

- **Parameters**

**Table 290 - Signature and Parameters of
EMC_VNXe_ReplicationServiceCapabilities.ConvertSyncTypeToReplicationType**

Parameter name	Qualifier	Type	Description/note
SyncType	IN, Mandatory	uint16	SyncType describes the type of copy.
Mode	IN, Mandatory	uint16	Mode describes whether the target elements will be updated synchronously or asynchronously.
LocalOrRemote	IN, Mandatory	uint16	Copy to local or remote replica.
SupportedReplicationTypes	OUT	uint16	Value representing the supported <i>ReplicationType</i> .

- **Return Results**

**Table 291 - Possible return code of
EMC_VNXe_ReplicationServiceCapabilities.ConvertSyncTypeToReplicationType**

Return code	Type	Description
0	uint32	Method completed OK
5	uint32	Invalid Parameter

Method: *ReplicationServiceCapabilities.GetDefaultConsistency*

- **Description**

This method is used to return the default *Consistency* value for a given *ReplicationType*.

- **Parameters**

**Table 292 - Signature and Parameters of
EMC_VNXe_ReplicationServiceCapabilities.GetDefaultConsistency**

Parameter name	Qualifier	Type	Description/note
----------------	-----------	------	------------------

Parameter name	Qualifier	Type	Description/note
ReplicationType	IN, Mandatory	uint16	Value representing the supported ReplicationType.
DefaultConsistency	OUT	uint16	Default consistency value.

- **Return Results**

**Table 293 - Possible return code of
EMC_VNXe_ReplicationServiceCapabilities.GetDefaultConsistency**

Return code	Type	Description
0	uint32	Method completed OK
5	uint32	Invalid Parameter

Method: ReplicationServiceCapabilities.GetDefaultGroupPersistency

- **Description**

This method is used to return the default *Persistency* for a newly created group.

- **Parameters**

**Table 294 - Signature and Parameters of
EMC_VNXe_ReplicationServiceCapabilities.GetDefaultGroupPersistency**

Parameter name	Qualifier	Type	Description/note
DefaultGroupPersistency	OUT	uint16	Default group persistency value.

- **Return Results**

**Table 295 - Possible return code of
EMC_VNXe_ReplicationServiceCapabilities.GetDefaultGroupPersistency**

Return code	Type	Description
0	uint32	Method completed OK

Method: ReplicationServiceCapabilities.GetSupportedConsistency

- **Description**

This method is used to return the supported *Consistency* for a given *ReplicationType*.

- **Parameters**

**Table 296 - Signature and Parameters of
EMC_VNXe_ReplicationServiceCapabilities.GetSupportedConsistency**

Parameter name	Qualifier	Type	Description/note
----------------	-----------	------	------------------

Parameter name	Qualifier	Type	Description/note
ReplicationType	IN, Mandatory	uint16	Value representing the ReplicationType.
SupportedConsistency	OUT	uint16 []	Array of supported consistency values.

- **Return Results**

**Table 297 - Possible return code of
EMC_VNXe_ReplicationServiceCapabilities.GetSupportedConsistency**

Return code	Type	Description
0	uint32	Method completed OK
5	uint32	Invalid Parameter

Method: *ReplicationServiceCapabilities.GetSupportedCopyStates*

- **Description**

For a given *ReplicationType*, this method is used to return supported *CopyStates* and a parallel array to indicate for a given *CopyState* whether the target element is host accessible or not.

- **Parameters**

**Table 298 - Signature and Parameters of
EMC_VNXe_ReplicationServiceCapabilities.GetSupportedCopyStates**

Parameter name	Qualifier	Type	Description/note
ReplicationType	IN, Mandatory	uint16	Value representing the ReplicationType.
SupportedCopyStates	OUT	uint16 []	Array of supported copy state values.
HostAccessible	OUT	boolean []	Parallel array to SupportedCopyStates[] to indicate whether in that CopyState the target element is host accessible or not.

- **Return Results**

**Table 299 - Possible return code of
EMC_VNXe_ReplicationServiceCapabilities.GetSupportedCopyStates**

Return code	Type	Description
0	uint32	Method completed OK
5	uint32	Invalid Parameter

Method: *ReplicationServiceCapabilities.GetSupportedFeatures*

- **Description**

For a given *ReplicationType*, this method is used to return supported features.

- **Parameters**

**Table 300 - Signature and Parameters of
EMC_VNXe_ReplicationServiceCapabilities.GetSupportedFeatures**

Parameter name	Qualifier	Type	Description/note
ReplicationType	IN, Mandatory	uint16	Value representing the ReplicationType.
Features	OUT	uint16 []	Array of supported Features.

- **Return Results**

**Table 301 - Possible return code of
EMC_VNXe_ReplicationServiceCapabilities.GetSupportedFeatures**

Return code	Type	Description
0	uint32	Method completed OK
5	uint32	Invalid Parameter

Method: *ReplicationServiceCapabilities.GetSupportedGroupCopyStates*

- **Description**

For a given *ReplicationType*, this method is used to return supported replication group *CopyStates*.

- **Parameters**

**Table 302 - Signature and Parameters of
EMC_VNXe_ReplicationServiceCapabilities.GetSupportedGroupCopyStates**

Parameter name	Qualifier	Type	Description/note
ReplicationType	IN, Mandatory	uint16	Value representing the ReplicationType.
SupportedCopyStates	OUT	uint16 []	Array of supported copy state values.

- **Return Results**

**Table 303 - Possible return code of
EMC_VNXe_ReplicationServiceCapabilities.GetSupportedGroupCopyStates**

Return code	Type	Description
0	uint32	Method completed OK

Return code	Type	Description
5	uint32	Invalid Parameter

Method: *ReplicationServiceCapabilities.GetSupportedGroupFeatures*

- **Description**

For a given *ReplicationType*, this method is used to return supported group features.

- **Parameters**

**Table 304 - Signature and Parameters of
EMC_VNXe_ReplicationServiceCapabilities.GetSupportedGroupFeatures**

Parameter name	Qualifier	Type	Description/note
ReplicationType	IN, Mandatory	uint16	Value representing the ReplicationType.
GroupFeatures	OUT	uint16 []	Array of supported group features.

- **Return Results**

**Table 305 - Possible return code of
EMC_VNXe_ReplicationServiceCapabilities.GetSupportedGroupFeatures**

Return code	Type	Description
0	uint32	Method completed OK
5	uint32	Invalid Parameter

Method: *ReplicationServiceCapabilities.GetSupportedOperations*

- **Description**

For a given *ReplicationType*, this method is used to return supported *Operations* on a *GroupSynchronized* association that can be supplied to the *ModifyReplicaSynchronization* method.

- **Parameters**

**Table 306 - Signature and Parameters of
EMC_VNXe_ReplicationServiceCapabilities.GetSupportedOperations**

Parameter name	Qualifier	Type	Description/note
ReplicationType	IN, Mandatory	uint16	Value representing the ReplicationType.
SupportedOperations	OUT	uint16 []	Array of supported operation values.

- **Return Results**

**Table 307 - Possible return code of
EMC_VNXe_ReplicationServiceCapabilities.GetSupportedOperations**

Return code	Type	Description
0	uint32	Method completed OK
5	uint32	Invalid Parameter

Client considerations

Model Specification

- Supported Key Features
 - Only *Async* replication type is supported.
 - Snapshot* and *Mirror* types are supported. *Clone* is not supported.
 - Both *Local* and *Remote* replication are supported.
 - Only *StorageVolume* (Thinly & Non-Thinly Provisioned) replication is supported.
 - Both *Individual* and *Group* element replication are supported.
 - Consistency Management* is supported.

Use case: Disvoicer Replication Services Capabilities

This use case describes how to discover capabilities of replication services.

- From Block Server (Top-level Array ComputerSystem), traverse *CIM_HostedService* to *CIM_ReplicationService* to locate hosted replication service.
- From *CIM_ReplicationService*, traverse *CIM_ElementCapabilities* to *CIM_ReplicationServiceCapabilities* to obtain replication service capabilities.
- From Block Server (Top-level Array ComputerSystem), traverse *CIM_HostedService* to *CIM_StorageConfigurationService* to locate hosted storage configuration service.
- From *CIM_StorageConfigurationService*, traverse *CIM_ElementCapabilities* to *CIM_StorageReplicationCapabilities* to obtain replication capabilities.
- Examine all capabilities obtained to learn system replication capabilities.

Use case: Create Promoted Snapshot of Standalone-LUN

This use case describes how to create a standalone-LUN snapshot and make the snapshot promoted via Replication Services.

- From Block Server (Top-level Array ComputerSystem), traverse *CIM_HostedService* to *CIM_ReplicationService* to locate hosted replication service.
- Invoke method *CIM_ReplicationService.CreateElementReplica()* to create a Standalone-LUN snapshot and make it promoted, by passing following parameters:
 - ElementName* – User defined name for the snapshot.

SyncType – Type of the copy to be made; it should be set to 7 (Snapshot).

SourceElement – Reference of the source volume.

3. Pull status of the Job output from the method to check status of the replication process.

Use case: Delete Promoted Snapshot of Standalone-LUN

This use case describes how to delete a promoted snapshot of a standalone-LUN via Replication Services.

1. From Block Server (Top-level Array ComputerSystem), traverse *CIM_HostedService* to *CIM_ReplicationService* to locate hosted replication service.
2. Invoke method *CIM_ReplicationService.ModifyReplicaSynchronization()* to delete a promoted snapshot of a Standalone-LUN, by passing following parameters:

Operation – Modify operation; it should be set to 19 (Return to ResourcePool).

Synchronization – Reference of the StorageSynchronized associating the replica (promoted volume snapshot) to be deleted and the source object (StorageVolume).

3. Pull status of the Job output from the method to check status of the deletion process.

Use case: Demote Promoted Snapshot of Standalone-LUN

This use case describes how to demote a promoted snapshot of a standalone-LUN via Replication Services.

1. From Block Server (Top-level Array ComputerSystem), traverse *CIM_HostedService* to *CIM_ReplicationService* to locate hosted replication service.
2. Invoke method *CIM_ReplicationService.ModifyReplicaSynchronization()* to delete a promoted snapshot of a Standalone-LUN, by passing following parameters:

Operation – Modify operation; it should be set to 9 (Dissovre).

Synchronization – Reference of the StorageSynchronized associating the replica (promoted volume snapshot) to be deleted and the source object (StorageVolume).

3. Pull status of the Job output from the method to check status of the demotion process.

Use case: Create SynchronizationAspect of Storage Element

This use case describes how to create a SynchronizationAspect (a non-promoted snapshot) of a standalone-LUN or a promoted LUN snapshot via Replication Services.

1. From Block Server (Top-level Array ComputerSystem), traverse *CIM_HostedService* to *CIM_ReplicationService* to locate hosted replication service.
2. Invoke method *CIM_ReplicationService.CreateSynchronizationAspect()* to create a non-promoted snapshot of a storage element (a volume or a promoted volume snapshot), by passing following parameters:

Name – User defined name for the snapshot.

SyncType – Type of the copy to be made; it should be set to 7 (Snapshot).

SourceElement – Reference of the source volume or promoted volume snapshot.

3. Pull status of the Job output from the method to check status of the replication process.

Use case: Delete SynchronizationAspect of Storage Element

This use case describes how to delete a SynchronizationAspect (a non-promoted snapshot) of a standalone-LUN or a promoted LUN snapshot via Replication Services.

1. From Block Server (Top-level Array ComputerSystem), traverse *CIM_HostedService* to *CIM_ReplicationService* to locate hosted replication service.
2. Invoke method *CIM_ReplicationService.ModifySettingsDefineState()* to delete a non-promoted snapshot of a Standalone-LUN or a promoted LUN snapshot, by passing following parameters:

Operation – Modify operation; it should be set to 4 (Delete).

SettingsState – Reference of the SettingsDefineState associating the SettingData (non-promoted snapshot) to be deleted and the ManagedElement (StorageVolume or promoted volume snapshot).

3. Pull status of the Job output from the method to check status of the deletion process.

Use case: Restore Storage Element from SynchronizationAspect

This use case describes how to restore a standalone-LUN or a promoted volume snapshot from non-promoted snapshot via Replication Services.

1. From Block Server (Top-level Array ComputerSystem), traverse *CIM_HostedService* to *CIM_ReplicationService* to locate hosted replication service.
2. Invoke method *CIM_ReplicationService.ModifySettingsDefineState()* to restore a Standalone-LUN or s promoted volume snapshot from non-promoted snapshot, by passing following parameters:

Operation – Modify operation; it should be set to 7 (Restore).

SettingsState – Reference of the SettingsDefineState associating the ManagedElement (StorageVolume or promoted volume snapshot) to be restored and the SettingData (non-promoted snapshot) from which the restoration is applied.

3. Pull status of the Job output from the method to check status of the restoration process.

Use case: Promote Non-Promoted Snapshot of Storage Element

This use case describes how to promote a non-promoted snapshot of a standalone-LUN or a promoted volume snapshot via Replication Services.

1. From Block Server (Top-level Array ComputerSystem), traverse *CIM_HostedService* to *CIM_ReplicationService* to locate hosted replication service.
2. Invoke method *CIM_ReplicationService.ModifySettingsDefineState()* to promote a non-promoted snapshot of a Standalone-LUN or a promoted volume snapshot, by passing following parameters:

Operation – Modify operation; it should be set to 5 (Copy To Target).

SettingsState – Reference of the SettingsDefineState associating the ManagedElement (StorageVolume or promoted volume snapshot) and the SettingData (non-promoted snapshot).

3. Pull status of the Job output from the method to check status of the restoration process.

Use case: Create Replication Group

This use case describes how to create a consistency replication group.

1. From Block Server (Top-level Array ComputerSystem), traverse *CIM_HostedService* to *CIM_ReplicationService* to locate hosted replication service.
2. Invoke method *CIM_ReplicationService.CreateGroup()* to create a consistency replication group.

Use case: Delete Replication Group

This use case describes how to delete a consistency replication group.

1. From Block Server (Top-level Array ComputerSystem), traverse *CIM_HostedService* to *CIM_ReplicationService* to locate hosted replication service.
2. Invoke method *CIM_ReplicationService.DeleteGroup()* to create a consistency replication group, by passing following parameters:

ReplicationGroup – Reference of the group to be deleted.

RemoveElement – If set to TRUE, all elements are removed out of the group and the empty group will be deleted at the same time.

Use case: Add LUN to a Replication Group

This use case describes how to add LUNs to a consistency replication group.

1. From Block Server (Top-level Array ComputerSystem), traverse *CIM_HostedService* to *CIM_ReplicationService* to locate hosted replication service.
2. Invoke method *CIM_ReplicationService.AddMembers()* to add LUNs into consistency replication group, by passing following parameters:

Members – List of references of the LUNs to be added into the group.

ReplicationGroup – Reference of the group where the LUNs to be added into.

Use case: Remove LUN from Replication Group

This use case describes how to remove LUNs from a consistency replication group.

1. From Block Server (Top-level Array ComputerSystem), traverse *CIM_HostedService* to *CIM_ReplicationService* to locate hosted replication service.
2. Invoke method *CIM_ReplicationService.RemoveMembers()* to remove LUNs from consistency replication group, by passing following parameters:

Members – List of references of the LUNs to be removed from the group.

ReplicationGroup – Reference of the group where the LUNs to be removed from.

DeleteOnEmptyElement – If set to TRUE, the group will be deleted when it is empty after removing LUNs from it.

Use case: Create Promoted Snapshot of Replication Group

This use case describes how to create a LUN group snapshot and make the snapshot promoted via Replication Services.

1. From Block Server (Top-level Array ComputerSystem), traverse *CIM_HostedService* to *CIM_ReplicationService* to locate hosted replication service.
2. Invoke method *CIM_ReplicationService.CreateGroupReplica()* to create a group snapshot and make it promoted, by passing following parameters:
 - RelationshipName* – User defined name for the group snapshot.
 - SyncType* – Type of the copy to be made; it should be set to 7 (Snapshot).
 - SourceGroup* – Reference of the source ReplicationGroup.
3. Pull status of the Job output from the method to check status of the replication process.

Use case: Delete Promoted Snapshot of Replication Group

This use case describes how to delete a promoted snapshot of a LUN group via Replication Services.

1. From Block Server (Top-level Array ComputerSystem), traverse *CIM_HostedService* to *CIM_ReplicationService* to locate hosted replication service.
2. Invoke method *CIM_ReplicationService.ModifyReplicaSynchronization()* to delete a promoted group snapshot, by passing following parameters:
 - Operation* – Modify operation; it should be set to 19 (Return to ResourcePool).
 - Synchronization* – Reference of the GroupSynchronized associating the replica (promoted group snapshot) to be deleted and the source group (ReplicationGroup).
3. Pull status of the Job output from the method to check status of the deletion process.

Use case: Demote Promoted Snapshot of Replication Group

This use case describes how to demote a promoted snapshot of a ReplicationGroup via Replication Services.

1. From Block Server (Top-level Array ComputerSystem), traverse *CIM_HostedService* to *CIM_ReplicationService* to locate hosted replication service.
2. Invoke method *CIM_ReplicationService.ModifyReplicaSynchronization()* to delete a promoted snapshot of a LUN group, by passing following parameters:
 - Operation* – Modify operation; it should be set to 9 (Dissovre).
 - Synchronization* – Reference of the StorageSynchronized associating the replica (promoted group snapshot) to be deleted and the source group (ReplicationGroup).
3. Pull status of the Job output from the method to check status of the demotion process.

Use case: Create SynchronizationAspect of Storage Group

This use case describes how to create a SynchronizationAspect (a non-promoted snapshot) of of a LUN group or a promoted LUN group snapshot via Replication Services.

1. From Block Server (Top-level Array ComputerSystem), traverse *CIM_HostedService* to *CIM_ReplicationService* to locate hosted replication service.

2. Invoke method *CIM_ReplicationService.CreateSynchronizationAspect()* to create a non-promoted snapshot of a storage group (a LUN group or a promoted group snapshot), by passing following parameters:

Name – User defined name for the snapshot.

SyncType – Type of the copy to be made; it should be set to 7 (Snapshot).

SourceGroup – Reference of the source group or promoted group snapshot.

3. Pull status of the Job output from the method to check status of the replication process.

Use case: Delete SynchronizationAspect of Storage Group

This use case describes how to delete a SynchronizationAspect (a non-promoted snapshot) of a LUN group or a promoted LUN group snapshot via Replication Services.

1. From Block Server (Top-level Array ComputerSystem), traverse *CIM_HostedService* to *CIM_ReplicationService* to locate hosted replication service.
2. Invoke method *CIM_ReplicationService.ModifySettingsDefineState()* to delete a non-promoted snapshot of a storage group, by passing following parameters:

Operation – Modify operation; it should be set to 4 (Delete).

SettingsState – Reference of the SettingsDefineState associating the SettingData (non-promoted snapshot) to be deleted and the ManagedElement (LUN group or promoted group snapshot).

3. Pull status of the Job output from the method to check status of the deletion process.

Use case: Restore Storage Group from SynchronizationAspect

This use case describes how to restore a LUN group or a promoted group snapshot from non-promoted snapshot via Replication Services.

1. From Block Server (Top-level Array ComputerSystem), traverse *CIM_HostedService* to *CIM_ReplicationService* to locate hosted replication service.
2. Invoke method *CIM_ReplicationService.ModifySettingsDefineState()* to restore a storage group from a non-promoted group snapshot, by passing following parameters:

Operation – Modify operation; it should be set to 7 (Restore).

SettingsState – Reference of the SettingsDefineState associating the ManagedElement (LUN group or promoted group snapshot) to be restored and the SettingData (non-promoted group snapshot) from which the restoration is applied.

3. Pull status of the Job output from the method to check status of the restoration process.

Use case: Promote Non-Promoted Snapshot of Storage Group

This use case describes how to promote a non-promoted snapshot of a LUN group or a promoted group snapshot via Replication Services.

1. From Block Server (Top-level Array ComputerSystem), traverse *CIM_HostedService* to *CIM_ReplicationService* to locate hosted replication service.

2. Invoke method *CIM_ReplicationService.ModifySettingsDefineState()* to promote a non-promoted snapshot of a LUN group or a promoted group snapshot, by passing following parameters:

Operation – Modify operation; it should be set to 5 (Copy To Target).

SettingsState – Reference of the SettingsDefineState associating the ManagedElement (LUN group or promoted group snapshot) and the SettingData (non-promoted group snapshot).

3. Pull status of the Job output from the method to check status of the restoration process.

Use case: Setup Standalone-LUN Mirror

This use case describes how to setup LUN mirror via Replication Services.

1. From Block Server (Top-level Array ComputerSystem), traverse *CIM_HostedService* to *CIM_ReplicationService* to locate hosted replication service.
2. Invoke method *CIM_ReplicationService.CreateElementReplica()* to setup LUN mirror, by passing following parameters:

SyncType – Type of the copy to be made; it should be set to 6 (Mirror).

SourceElement – Reference of the source StorageVolume.

TargetElement – Reference of the target StorageVolume

3. Pull status of the Job output from the method to check status of the replication process.

Use case: Detach Standalone-LUN Mirror

This use case describes how to Detach LUN mirror via Replication Services.

1. From Block Server (Top-level Array ComputerSystem), traverse *CIM_HostedService* to *CIM_ReplicationService* to locate hosted replication service.
2. Invoke method *CIM_ReplicationService.ModifyReplicaSynchronization()* to detach LUN mirror, by passing following parameters:

Operation – Modify operation; it should be set to 8 (Detach).

Synchronization – Reference of the StorageSynchronized associating the replica and the source object.

3. Pull status of the Job output from the method to check status of the detach process.

Use case: Split Standalone-LUN Mirror

This use case describes how to Split LUN mirror via Replication Services.

1. From Block Server (Top-level Array ComputerSystem), traverse *CIM_HostedService* to *CIM_ReplicationService* to locate hosted replication service.
2. Invoke method *CIM_ReplicationService.ModifyReplicaSynchronization()* to split LUN mirror, by passing following parameters:

Operation – Modify operation; it should be set to 21 (Split).

Synchronization – Reference of the StorageSynchronized associating the replica and the source object.

3. Pull status of the Job output from the method to check status of the split process.

Use case: Resync Standalone-LUN Mirror after Split

This use case describes how to Resync LUN mirror after Split via Replication Services.

1. From Block Server (Top-level Array ComputerSystem), traverse *CIM_HostedService* to *CIM_ReplicationService* to locate hosted replication service.
2. Invoke method *CIM_ReplicationService.ModifyReplicaSynchronization()* to resync LUN mirror after split, by passing following parameters:

Operation – Modify operation; it should be set to 14 (Resync Replica).
Synchronization – Reference of the StorageSynchronized associating the replica and the source object.
3. Pull status of the Job output from the method to check status of the resync process.

Use case: Failover Standalone-LUN Mirror

This use case describes how to Failover LUN mirror via Replication Services.

1. From **Target** Block Server (Top-level Array ComputerSystem hosting the target storage object in replication), traverse *CIM_HostedService* to *CIM_ReplicationService* to locate hosted replication service.
2. Invoke method *CIM_ReplicationService.ModifyReplicaSynchronization()* to failover LUN mirror, by passing following parameters:

Operation – Modify operation; it should be set to 10 (Failover).
Synchronization – Reference of the StorageSynchronized associating the replica and the source object.
3. Pull status of the Job output from the method to check status of the failover process.

Use case: Failback Standalone-LUN Mirror after Failover

This use case describes how to Failback LUN mirror after Failover via Replication Services.

1. From **Target** Block Server (Top-level Array ComputerSystem hosting the target storage object in replication), traverse *CIM_HostedService* to *CIM_ReplicationService* to locate hosted replication service.
2. Invoke method *CIM_ReplicationService.ModifyReplicaSynchronization()* to failback LUN mirror after failover, by passing following parameters:

Operation – Modify operation; it should be set to 11 (Failback).
Synchronization – Reference of the StorageSynchronized associating the replica and the source object.
3. Pull status of the Job output from the method to check status of the failback process.

Use case: Reverse Roles of Standalone-LUN Mirror

This use case describes how to reverse roles of replication via Replication Services.

If replication session is in OK state, Reverse Roles should be initiated on source side then a Resync operation is required on destination side.

1. From Block Server (Top-level Array ComputerSystem), traverse *CIM_HostedService* to *CIM_ReplicationService* to locate hosted replication service.
2. Invoke method *CIM_ReplicationService.ModifyReplicaSynchronization()* reverse roles of LUN mirror, by passing following parameters:

Operation – Modify operation; it should be set to 20 (Reverse Roles).

Synchronization – Reference of the StorageSynchronized associating the replica and the source object.

3. Pull status of the Job output from the method to check status of the reverse roles process.
4. Resync the replication session on destination side.

If replication session in FailOver state, Reverse Roles should be initiated on destination side.

Use case: Setup Replication Group Mirror

This use case describes how to setup LUNGroup mirror via Replication Services.

1. From Block Server (Top-level Array ComputerSystem), traverse *CIM_HostedService* to *CIM_ReplicationService* to locate hosted replication service.
2. Invoke method *CIM_ReplicationService.CreateElementReplica()* to setup LUNGroup mirror, by passing following parameters:

SyncType – Type of the copy to be made; it should be set to 6 (Mirror).

SourceGroup – Reference of the source ReplicationGroup.

TargetGroup – Reference of the target ReplicationGroup.

3. Pull status of the Job output from the method to check status of the replication process.

Use case: Detach Replication Group Mirror

This use case describes how to Detach LUNGroup mirror via Replication Services.

1. From Block Server (Top-level Array ComputerSystem), traverse *CIM_HostedService* to *CIM_ReplicationService* to locate hosted replication service.
2. Invoke method *CIM_ReplicationService.ModifyReplicaSynchronization()* to detach LUNGroup mirror, by passing following parameters:

Operation – Modify operation; it should be set to 8 (Detach).

Synchronization – Reference of the StorageSynchronized associating the replica and the source group.

3. Pull status of the Job output from the method to check status of the detach process.

Use case: Split Replication Group Mirror

This use case describes how to Split LUNGroup mirror via Replication Services.

1. From Block Server (Top-level Array ComputerSystem), traverse *CIM_HostedService* to *CIM_ReplicationService* to locate hosted replication service.
2. Invoke method *CIM_ReplicationService.ModifyReplicaSynchronization()* to split LUNGroup mirror, by passing following parameters:

Operation – Modify operation; it should be set to 21 (Split).

Synchronization – Reference of the StorageSynchronized associating the replica and the source group.

3. Pull status of the Job output from the method to check status of the split process.

Use case: Resync Replication Group Mirror after Split

This use case describes how to Resync LUNGroup mirror after Split via Replication Services.

1. From Block Server (Top-level Array ComputerSystem), traverse *CIM_HostedService* to *CIM_ReplicationService* to locate hosted replication service.
2. Invoke method *CIM_ReplicationService.ModifyReplicaSynchronization()* to resync LUNGroup mirror after split, by passing following parameters:

Operation – Modify operation; it should be set to 14 (Resync Replica).
Synchronization – Reference of the StorageSynchronized associating the replica and the source group.
3. Pull status of the Job output from the method to check status of the resync process.

Use case: Failover Replication Group Mirror

This use case describes how to Failover LUNGroup mirror via Replication Services.

1. From **Target** Block Server (Top-level Array ComputerSystem hosting the target storage object in replication), traverse *CIM_HostedService* to *CIM_ReplicationService* to locate hosted replication service.
2. Invoke method *CIM_ReplicationService.ModifyReplicaSynchronization()* to failover LUNGroup mirror, by passing following parameters:

Operation – Modify operation; it should be set to 10 (Failover).
Synchronization – Reference of the StorageSynchronized associating the replica and the source group.
3. Pull status of the Job output from the method to check status of the failover process.

Use case: Failback Replication Group Mirror after Failover

This use case describes how to Failback LUNGroup mirror after Failover via Replication Services.

1. From **Target** Block Server (Top-level Array ComputerSystem hosting the target storage object in replication), traverse *CIM_HostedService* to *CIM_ReplicationService* to locate hosted replication service.
2. Invoke method *CIM_ReplicationService.ModifyReplicaSynchronization()* to failback LUNGroup mirror after failover, by passing following parameters:

Operation – Modify operation; it should be set to 11 (Failback).
Synchronization – Reference of the StorageSynchronized associating the replica and the source group.
3. Pull status of the Job output from the method to check status of the failback process.

Use case: Reverse Roles of Replication Group Mirror

This use case describes how to reverse roles of replication after Failover via Replication Services.

If replication session is in OK state, Reverse Roles should be initiated on source side then a Resync operation is required on destination side.

1. From Block Server (Top-level Array ComputerSystem), traverse *CIM_HostedService* to *CIM_ReplicationService* to locate hosted replication service.
2. Invoke method *CIM_ReplicationService.ModifyReplicaSynchronization()* to reverse roles of LUNGroup mirror, by passing following parameters:

Operation – Modify operation; it should be set to 20 (Reverse Roles).

Synchronization – Reference of the StorageSynchronized associating the replica and the source group.

3. Pull status of the Job output from the method to check status of the reverse roles process.
4. Resync the replication session on destination side.

If replication session in FailOver state, Reverse Roles should be initiated on destination side.

CIM Element

The implemented classes and associations related to Replication Services Profile in the VNXe storage system are described as follows:

Table 308 - CIM Elements implemented in VNXe for Replication Services Profile

CIM Class	Implemented Class	Description
CIM_BindsTo	EMC_VNXe_IPProtocolEndpointForReplication_TCPProtocolEndpointReplication_BindsToAssocLeaf	Associates the tcp protocol endpoint and the ip protocol endpoints to which it binds.
CIM_BindsTo	EMC_VNXe_TCPProtocolEndpointForReplication_ReplicationProtocolEndpoint_BindsToAssocLeaf	Associates the replication protocol endpoint and the tcp protocol endpoints to which it binds.
CIM_DeviceSAPImplementation	EMC_VNXe_EthernetPort_IPProtocolEndpointForReplication_DeviceSAPImplementationAssocLeaf	Associates the ip protocol endpoint and the Ethernet port.
CIM_DeviceSAPImplementation	EMC_VNXe_EthernetPort_ReplicationProtocolEndpoint_DeviceSAPImplementationAssocLeaf	Associates the ip protocol endpoint and the Ethernet port.
CIM_ElementCapabilities	EMC_VNXe_ReplicationService_ReplicationServiceCapabilities_ElementCapabilitiesAssocLeaf	Associates the ReplicationService and its Capabilities.
CIM_GroupSynchronized	EMC_VNXe_ReplicationGroup_ReplicationGroup_GroupMirrorSynchronizedAssocLeaf	Associates the source element (ReplicationGroup) and target replica (ReplicationGroup).
CIM_GroupSynchronized	EMC_VNXe_ReplicationGroup_SnapGroup_GroupSynchronizedAssocLeaf	Associates the source element (ReplicationGroup) and target replica (SnapGroup).
CIM_HostedAccessPoint	EMC_VNXe_StorageSystem_IPProtocolEndpointForReplication_HostedAccessPointAssocLeaf	Associates the ip protocol endpoints and the system hosting the endpoints.
CIM_HostedAccessPoint	EMC_VNXe_StorageSystem_ReplicationProtocolEndpoint_HostedAccessPointAssocLeaf	Associates the replication protocol endpoints and the system hosting the endpoints.
CIM_HostedAccessPoint	EMC_VNXe_StorageSystem_TCPProtocolEndpointForReplication_HostedAccessPointAssocLeaf	Associates the tcp protocol endpoints and the system hosting the endpoints.

CIM Class	Implemented Class	Description
CIM_HostedCollection	EMC_VNXe_StorageSystem_RemoteReplicationCollection_HostedCollectionAssocLeaf	Associates the RemoteReplicationCollection and the system hosting the collection.
CIM_HostedCollection	EMC_VNXe_StorageSystem_ReplicationGroup_HostedCollectionAssocLeaf	Associates the ReplicationGroup and the system hosting the group.
CIM_HostedCollection	EMC_VNXe_StorageSystem_SnapGroup_HostedCollectionAssocLeaf	Associates the SnapGroup and the system hosting the group.
CIM_HostedService	EMC_VNXe_StorageSystem_ReplicationService_HostedServiceAssocLeaf	Associates the hosted ReplicationService and the hosting system.
CIM_IPProtocolEndpoint	EMC_VNXe_IPProtocolEndpointForReplicationLeaf	Represents an IP protocol endpoint provides access in replication operations.
CIM_MemberOfCollection	EMC_VNXe_RemoteReplicationCollection_ReplicationProtocolEndpoint_MemberOfCollectionAssocLeaf	Associates the replication protocol endpoints and the collection collecting the endpoints.
CIM_OrderedMemberOfCollection	EMC_VNXe_ReplicationGroup_RemoteStorageVolume_OrderedMemberOfCollectionAssocLeaf	Associates the ReplicationGroup and its member ReplicationEntity contained in the group.
CIM_OrderedMemberOfCollection	EMC_VNXe_ReplicationGroup_StorageVolume_OrderedMemberOfCollectionAssocLeaf	Associates the ReplicationGroup and its member StorageVolume contained in the group.
CIM_OrderedMemberOfCollection	EMC_VNXe_SnapGroup_SnapVolume_OrderedMemberOfCollectionAssocLeaf	Associates the SnapGroup and its member SnapVolume contained in the group.
CIM_ProtocolEndpoint	EMC_VNXe_ReplicationProtocolEndpointLeaf	Represents a replication protocol endpoint provides access in replication operations.
CIM_RemoteReplicationCollection	EMC_VNXe_RemoteReplicationCollectionLeaf	Represents a collection of all paths that provide access to a remote system for replication operations.
CIM_ReplicationEntity	EMC_VNXe_RemoteStorageVolumeLeaf	Represents a remote storage volume in replication session.
CIM_ReplicationGroup	EMC_VNXe_ReplicationGroupLeaf	Represents a LUN group.
CIM_ReplicationGroup	EMC_VNXe_SnapGroupLeaf	Represents a LUN Group Snapshot.
CIM_ReplicationService	EMC_VNXe_ReplicationServiceLeaf	Represents the Replication Service of the storage system.

CIM Class	Implemented Class	Description
CIM_ReplicationServiceCapabilities	EMC_VNXe_ReplicationServiceCapabilitiesLeaf	Represents the Capabilities of the ReplicationService.
CIM_ServiceAffectsElement	EMC_VNXe_ReplicationService_RemoteReplicationCollection_ServiceAffectsElementAssocLeaf	Associates the ReplicationService and RemoteReplicationCollection affected by the service.
CIM_ServiceAffectsElement	EMC_VNXe_ReplicationService_RemoteStorageVolume_ServiceAffectsElementAssocLeaf	Associates the ReplicationService and ReplicationEntity affected by the service.
CIM_ServiceAffectsElement	EMC_VNXe_ReplicationService_ReplicationGroup_ServiceAffectsElementAssocLeaf	Associates the ReplicationService and ReplicationGroup affected by the service.
CIM_ServiceAffectsElement	EMC_VNXe_ReplicationService_SnapGroup_ServiceAffectsElementAssocLeaf	Associates the ReplicationService and SnapGroup affected by the service.
CIM_SettingsDefineState	EMC_VNXe_ReplicationGroup_SynchronizationAspect_SettingsDefineStateAssocLeaf	Associates the ReplicationGroup and SynchronizationAspect.
CIM_SettingsDefineState	EMC_VNXe_SnapGroup_SynchronizationAspect_SettingsDefineStateAssocLeaf	Associates the SnapGroup and SynchronizationAspect.
CIM_SettingsDefineState	EMC_VNXe_SnapVolume_SynchronizationAspect_SettingsDefineStateAssocLeaf	Associates the SnapVolume and SynchronizationAspect.
CIM_SettingsDefineState	EMC_VNXe_StorageVolume_SynchronizationAspect_SettingsDefineStateAssocLeaf	Associates the StorageVolume and SynchronizationAspect.
CIM_StorageReplicationCapabilities	EMC_VNXe_MirrorReplicationCapabilitiesLeaf	Represents the Mirror Capabilities of the storage system.
CIM_StorageReplicationCapabilities	EMC_VNXe_SnapReplicationCapabilitiesLeaf	Represents the Snapshot Capabilities of the storage system.
CIM_StorageSetting	EMC_VNXe_SnapVolumeSettingLeaf	Represents the StorageSetting of the SnapVolume.
CIM_StorageSynchronized	EMC_VNXe_RemoteStorageVolume_StorageVolume_MirrorSynchronizedAssocLeaf	Associates the source element (remote StorageVolume) and the target replica (local StorageVolume).
CIM_StorageSynchronized	EMC_VNXe_StorageVolume_RemoteStorageVolume_MirrorSynchronizedAssocLeaf	Associates the source element (StorageVolume) and the target replica (remote StorageVolume).
CIM_StorageSynchronized	EMC_VNXe_StorageVolume_SnapVolume_StorageSynchronizedAssocLeaf	Associates the source element (StorageVolume) and the target replica (SnapVolume).
CIM_StorageSynchronized	EMC_VNXe_StorageVolume_StorageVolume_MirrorSynchronizedAssocLeaf	Associates the source element (StorageVolume) and the target

CIM Class	Implemented Class	Description
		replica (local StorageVolume).
CIM_StorageVolume	EMC_VNXe_SnapVolumeLeaf	Represents a LUN snapshot.
CIM_SynchronizationAspect	EMC_VNXe_SynchronizationAspectForSourceGroupLeaf	Used to keep track of the source group of a copy operation.
CIM_SynchronizationAspect	EMC_VNXe_SynchronizationAspectForSourceLeaf	keep track of the source object of a copy operation.
CIM_SystemDevice	EMC_VNXe_StorageSystem_SnapVolume_SystemDeviceAssocLeaf	Associates the LUN snapshot and the storage system.
CIM_TCPProtocolEndpoint	EMC_VNXe_TCPProtocolEndpointForReplicationLeaf	Represents a TCP protocol endpoint provides access in replication operations.

EMC_VNXe_ReplicationServiceLeaf

Table 309 - Referenced properties/methods for EMC_VNXe_ReplicationServiceLeaf

CIM property	Description/notes
SystemCreationClassName	Set to "EMC_VNXe_StorageSystemLeaf".
SystemName	Set as system name of VNXe.
CreationClassName	Set to "EMC_VNXe_ReplicationServiceLeaf".
Name	Set to 'ReplicationService'.

EMC_VNXe_ReplicationServiceCapabilitiesLeaf

Table 310 - Referenced properties/methods for EMC_VNXe_ReplicationServiceCapabilitiesLeaf

CIM property	Description/notes
InstanceID	Unique ID for this instance.
ElementName	User friendly name for this instance.
SupportedReplicationTypes	Set to [7: Asynchronous Snapshot Local].
SupportedStorageObjects	Set to [2: StorageVolume].
SupportedSynchronousActions	Set to [12: CreateGroup, 13: DeleteGroup, 14: AddMembers, 15: RemoveMembers].
SupportedAsynchronousActions	Set to [2: CreateElementReplica, 3: CreateGroupReplica, 5: ModifyReplicaSynchronization].

EMC_VNXe_SnapReplicationCapabilitiesLeaf

Table 311 - Referenced properties/methods for EMC_VNXe_SnapReplicationCapabilitiesLeaf

CIM property	Description/notes
InstanceID	Unique ID for this instance.
ElementName	User friendly name for this instance.
SupportedSynchronizationType	Set to [5: UnSyncAssoc Delta].
SupportedModifyOperations	Set to [5: Restore].
SupportedSynchronousActions	Set to empty array.
SupportedAsynchronousActions	Set to [2: Local Replica Creation, 4: Local Replica Modification].
MaximumReplicasPerSource	Maximum number of replicas that can be associated with one source element. Set to 256.
ReplicaHostAccessibility	Indicates host access restrictions for replicas with this capabilities. Set to '2: Not Accessible'.
HostAccessibleState	Set to '11: Idle'.
InitialReplicationState	Set to '4: Idle'.
DeltaReplicaPoolAccess	Indicates that a specialized pool is required as a container for delta replication. Set to '3: Exclusive'.

EMC_VNXe_MirrorReplicationCapabilitiesLeaf

Table 312 - Referenced properties/methods for EMC_VNXe_MirrorReplicationCapabilitiesLeaf

CIM property	Description/notes
InstanceID	Unique ID for this instance.
ElementName	User friendly name for this instance.
SupportedSynchronizationType	Set to [2: Async].
SupportedModifyOperations	Set to [2: Detach, 4: Resync, 12: Start Copy, 13: Stop Copy].
SupportedSynchronousActions	Set to empty array.
SupportedAsynchronousActions	Set to [2: Local Replica Creation, 3: Remote Replica Creation, 4: Local Replica Modification, 5: Remote Replica Modification].
MaximumReplicasPerSource	Maximum number of replicas that can be associated with one source element.

CIM property	Description/notes
	Set to 256.
MaximunRemoteReplicationDepth	Maximum remote mirror replication depth allowed by the service.
ReplicaHostAccessibility	Indicates host access restrictions for replicas with this capabilities. Set to '2: Not Accessible'.
InitialReplicationState	Set to '4: Idle'.

EMC_VNXe_StorageVolumeLeaf

For detailed information, refer to [EMC_VNXe_StorageVolumeLeaf](#) in Block Services Package.

EMC_VNXe_SnapVolumeLeaf

Table 313 - Referenced properties/methods for EMC_VNXe_SnapVolumeLeaf

CIM property	Description/notes
SystemCreationClassName	Set to "EMC_VNXe_StorageSystemLeaf"
SystemName	Set as system name of VNXe
CreationClassName	Set to "EMC_VNXe_SnapVolumeLeaf"
DeviceID	Friendly ID of the SnapVolume.
Access	Indicates the read and write support.
ElementName	Name of the SnapVolume.
Name	Friendly ID of the SnapVolume.
NameFormat	Set to '7: SNVM'
NameNamespace	Set to '7: SNVM'
OtherIdentifyingInfo	Additional data for DeviceID information.
IdentifyingDescription	Set to ['NAA;VPD83Type3']
OtherNameformat	Describe the format of Name property when NameFormat is set to '1: Other'
OtherNameNamespace	Describe the namespace of Name property when NameNamespace is set to '1: Other'
OperationalStatus	Current operation status of the SnapVolume.
StatusDescriptions	Set to 512
BlockSize	Block numbers of the SnapVolume.

CIM property	Description/notes
NumberOfBlocks	Consumable block numbers of the SnapVolume.
ConsumableBlocks	Set to FALSE.
Primordial	Set to ['1: Unknown].
ExtentStatus	Set to ['1: Unknown].
ExtentDiscriminator	Set to ['SNIA: Allocated].
ThinlyProvisioned	Indicates if the SnapVolume is thinly provisioned. Set to TRUE.
DataRedundancy	Number of complete copies of data currently maintained.
DeltaReservation	Percentage that specifies the amount of space that should be reserved in a replica for caching changes.
PackageRedundancy	Number of physical packages can currently fail without data loss.
ParityLayout	Parity layout of the volume.
IsBasedOnUnderlyingRedundancy	Set to TRUE.
HealthState	Current health state of the volume.
NoSinglePointOfFailure	Set to TRUE.
Usage	Usage of the volume.

EMC_VNXe_ReplicationGroupLeaf

Table 314 - Referenced properties/methods for EMC_VNXe_ReplicationGroupLeaf

CIM property	Description/notes
InstanceID	Unique ID for this instance.
ElementName	User friendly name for this instance.
Persistent	Set to TRUE.
DeleteOnEmptyElement	Set to FALSE.
DeleteOnUnassociated	Set to FALSE.

EMC_VNXe_SnapGroupLeaf

Table 315 - Referenced properties/methods for EMC_VNXe_SnapGroupLeaf

CIM property	Description/notes
--------------	-------------------

CIM property	Description/notes
InstanceID	Unique ID for this instance.
ElementName	User friendly name for this instance.
Persistent	Set to TRUE.
DeleteOnEmptyElement	Set to FALSE.
DeleteOnUnassociated	Set to FALSE.

EMC_VNXe_SynchronizationAspectForSourceLeaf

Table 316 - Referenced properties/methods for EMC_VNXe_SynchronizationAspectForSourceLeaf

CIM property	Description/notes
InstanceID	Unique ID for this instance.
ElementName	User friendly name for this instance.
WhenPointInTime	Specifies when point-in-time was created.
SourceElement	The source object.
SyncType	Type of association between source and target elements. Set to '7: Snapshot'

EMC_VNXe_SynchronizationAspectForSourceGroupLeaf

Table 317 - Referenced properties/methods for EMC_VNXe_SynchronizationAspectForSourceGroupLeaf

CIM property	Description/notes
InstanceID	Unique ID for this instance.
ElementName	User friendly name for this instance.
WhenPointInTime	Specifies when point-in-time was created.
SourceElement	The source group.
SyncType	Type of association between source and target elements. Set to '7: Snapshot'

EMC_VNXe_StorageSystem_ReplicationService_HostedServiceAssocLeaf

Table 318 - Referenced properties/methods for EMC_VNXe_StorageSystem_ReplicationService_HostedServiceAssocLeaf

CIM property	Description/notes
--------------	-------------------

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_StorageSystemLeaf
Dependent	Reference of EMC_VNXe_ReplicationServiceLeaf

EMC_VNXe_ReplicationService_ReplicationServiceCapabilities_ElementCapabilitiesAssocLeaf

Table 319 - Referenced properties/methods for EMC_VNXe_ReplicationService_ReplicationServiceCapabilities_ElementCapabilitiesAssocLeaf

CIM property	Description/notes
ManagedElement	Reference of EMC_VNXe_ReplicationServiceLeaf
Capabilities	Reference of EMC_VNXe_ReplicationServiceCapabilitiesLeaf

EMC_VNXe_StorageSystem_ReplicationGroup_HostedCollectionAssocLeaf

Table 320 - Referenced properties/methods for EMC_VNXe_StorageSystem_ReplicationGroup_HostedCollectionAssocLeaf

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_StorageSystemLeaf
Dependent	Reference of EMC_VNXe_ReplicationGroupLeaf

EMC_VNXe_StorageSystem_SnapGroup_HostedCollectionAssocLeaf

Table 321 - Referenced properties/methods for EMC_VNXe_StorageSystem_SnapGroup_HostedCollectionAssocLeaf

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_StorageSystemLeaf
Dependent	Reference of EMC_VNXe_SnapGroupLeaf

EMC_VNXe_StorageVolume_SnapVolume_StorageSynchronizedAssocLeaf

Table 322 - Referenced properties/methods for EMC_VNXe_StorageVolume_SnapVolume_StorageSynchronizedAssocLeaf

CIM property	Description/notes
SystemElement	Reference of EMC_VNXe_StorageVolumeLeaf
SyncedElement	Reference of EMC_VNXe_SnapVolumeLeaf
CopyMethodology	Specifies the copy methodology the service uses to create or maintain the target element.

CIM property	Description/notes
CopyState	Describes the state of the association with respect to Replication activity.
CopyType	Describes the copy type applied to this association.
Mode	Describes whether the target element will be updated synchronously or asynchronously.
WhenSynced	Specifies the time when the replication was synced.
PercentSynced	Specifies the percent of the work.
ProgressStatus	Describes the status of the association with respect to Replication activity.
ReplicaType	Specifies how the Replica is being maintained.
RequestedCopyState	Indicates the last requested or desired state for the association.
SyncState	Describes the state of the association with respect to Replication activity.
SyncType	Describes the intended outcome of the replication.

EMC_VNXe_StorageVolume_StorageVolume_MirrorSynchronizedAssocLeaf

**Table 323 - Referenced properties/methods for
EMC_VNXe_StorageVolume_StorageVolume_MirrorSynchronizedAssocLeaf**

CIM property	Description/notes
SystemElement	Reference of EMC_VNXe_StorageVolumeLeaf
SyncedElement	Reference of EMC_VNXe_StorageVolumeLeaf
CopyMethodology	Specifies the copy methodology the service uses to create or maintain the target element.
CopyState	Describes the state of the association with respect to Replication activity.
CopyType	Describes the copy type applied to this association.
Mode	Describes whether the target element will be updated synchronously or asynchronously.
WhenSynced	Specifies the time when the replication was synced.
PercentSynced	Specifies the percent of the work.
ProgressStatus	Describes the status of the association with respect to Replication activity.
ReplicaType	Specifies how the Replica is being maintained.

CIM property	Description/notes
RequestedCopyState	Indicates the last requested or desired state for the association.
SyncState	Describes the state of the association with respect to Replication activity.
SyncType	Describes the intended outcome of the replication.

EMC_VNXe_StorageVolume_RemoteStorageVolume_MirrorSynchronizedAssocLeaf

**Table 324 - Referenced properties/methods for
EMC_VNXe_StorageVolume_RemoteStorageVolume_MirrorSynchronizedAssocLeaf**

CIM property	Description/notes
SystemElement	Reference of EMC_VNXe_StorageVolumeLeaf
SyncedElement	Reference of EMC_VNXe_RemoteStorageVolumeLeaf
UndiscoveredElement	Specifies whether the source, the target or both elements involved in a copy operation are undiscovered.
CopyMethodology	Specifies the copy methodology the service uses to create or maintain the target element.
CopyState	Describes the state of the association with respect to Replication activity.
CopyType	Describes the copy type applied to this association.
Mode	Describes whether the target element will be updated synchronously or asynchronously.
WhenSynced	Specifies the time when the replication was synced.
PercentSynced	Specifies the percent of the work.
ProgressStatus	Describes the status of the association with respect to Replication activity.
ReplicaType	Specifies how the Replica is being maintained.
RequestedCopyState	Indicates the last requested or desired state for the association.
SyncState	Describes the state of the association with respect to Replication activity.
SyncType	Describes the intended outcome of the replication.

EMC_VNXe_StorageVolume_SynchronizationAspect_SettingsDefineStateAssocLeaf

**Table 325 - Referenced properties/methods for
EMC_VNXe_StorageVolume_SynchronizationAspect_SettingsDefineStateAssocLeaf**

CIM property	Description/notes
ManagedElement	Reference of EMC_VNXe_StorageVolumeLeaf
SettingData	Reference of EMC_VNXe_SynchronizationAspectForSourceLeaf

EMC_VNXe_RemoteStorageVolume_StorageVolume_MirrorSynchronizedAssocLeaf

**Table 326 - Referenced properties/methods for
EMC_VNXe_RemoteStorageVolume_StorageVolume_MirrorSynchronizedAssocLeaf**

CIM property	Description/notes
SystemElement	Reference of EMC_VNXe_RemoteStorageVolumeLeaf
SyncedElement	Reference of EMC_VNXe_StorageVolumeLeaf
UndiscoveredElement	Specifies whether the source, the target or both elements involved in a copy operation are undiscovered.
CopyMethodology	Specifies the copy methodology the service uses to create or maintain the target element.
CopyState	Describes the state of the association with respect to Replication activity.
CopyType	Describes the copy type applied to this association.
Mode	Describes whether the target element will be updated synchronously or asynchronously.
WhenSynced	Specifies the time when the replication was synced.
PercentSynced	Specifies the percent of the work.
ProgressStatus	Describes the status of the association with respect to Replication activity.
ReplicaType	Specifies how the Replica is being maintained.
RequestedCopyState	Indicates the last requested or desired state for the association.
SyncState	Describes the state of the association with respect to Replication activity.
SyncType	Describes the intended outcome of the replication.

EMC_VNXe_SnapVolume_SynchronizationAspect_SettingsDefineStateAssocLeaf

**Table 327 - Referenced properties/methods for
EMC_VNXe_SnapVolume_SynchronizationAspect_SettingsDefineStateAssocLeaf**

CIM property	Description/notes
ManagedElement	Reference of EMC_VNXe_SnapVolumeLeaf

CIM property	Description/notes
SettingData	Reference of EMC_VNXe_SynchronizationAspectForSourceLeaf

EMC_VNXe_ReplicationGroup_SnapGroup_GroupSynchronizedAssocLeaf

**Table 328 - Referenced properties/methods for
EMC_VNXe_ReplicationGroup_SnapGroup_GroupSynchronizedAssocLeaf**

CIM property	Description/notes
SystemElement	Reference of EMC_VNXe_ReplicationGroupLeaf
SyncedElement	Reference of EMC_VNXe_SnapGroupLeaf
RelationshipName	Unique name of the relationship.
CopyState	Describes the state of the association with respect to Replication activity.
Mode	Describes whether the target element will be updated synchronously or asynchronously.
WhenSynced	Specifies the time when the replication was synced.
PercentSynced	Specifies the percent of the work.
ProgressStatus	Describes the status of the association with respect to Replication activity.
ConsistencyEnabled	Indicates whether consistency is enabled.
ConsistencyType	Indicates the consistency type used by the source and its associated target group.
ConsistencyState	Indicates current state of the consistency.
ConsistencyStatus	Indicates current status of the consistency.
SyncType	Describes the intended outcome of the replication.

EMC_VNXe_ReplicationGroup_ReplicationGroup_GroupMirrorSynchronizedAssocLeaf

**Table 329 - Referenced properties/methods for
EMC_VNXe_ReplicationGroup_ReplicationGroup_GroupMirrorSynchronizedAssocLeaf**

CIM property	Description/notes
SystemElement	Reference of EMC_VNXe_ReplicationGroupLeaf
SyncedElement	Reference of EMC_VNXe_ReplicationGroupLeaf
RelationshipName	Unique name of the relationship.
CopyState	Describes the state of the association with respect to Replication

CIM property	Description/notes
	activity.
Mode	Describes whether the target element will be updated synchronously or asynchronously.
WhenSynced	Specifies the time when the replication was synced.
PercentSynced	Specifies the percent of the work.
ProgressStatus	Describes the status of the association with respect to Replication activity.
ConsistencyEnabled	Indicates whether consistency is enabled.
ConsistencyType	Indicates the consistency type used by the source and its associated target group.
ConsistencyState	Indicates current state of the consistency.
ConsistencyStatus	Indicates current status of the consistency.
SyncType	Describes the intended outcome of the replication.

EMC_VNXe_ReplicationGroup_SynchronizationAspect_SettingsDefineStateAssocLeaf

**Table 330 - Referenced properties/methods for
EMC_VNXe_ReplicationGroup_SynchronizationAspect_SettingsDefineStateAssocLeaf**

CIM property	Description/notes
ManagedElement	Reference of EMC_VNXe_ReplicationGroupLeaf
SettingData	Reference of EMC_VNXe_SynchronizationAspectForSourceGroupLeaf

EMC_VNXe_SnapGroup_SynchronizationAspect_SettingsDefineStateAssocLeaf

**Table 331 - Referenced properties/methods for
EMC_VNXe_SnapGroup_SynchronizationAspect_SettingsDefineStateAssocLeaf**

CIM property	Description/notes
ManagedElement	Reference of EMC_VNXe_SnapGroupLeaf
SettingData	Reference of EMC_VNXe_SynchronizationAspectForSourceGroupLeaf

EMC_VNXe_ReplicationGroup_StorageVolume_OrderedMemberOfCollectionAssocLeaf

**Table 332 - Referenced properties/methods for
EMC_VNXe_ReplicationGroup_StorageVolume_OrderedMemberOfCollectionAssocLeaf**

CIM property	Description/notes
--------------	-------------------

CIM property	Description/notes
Collection	Reference of EMC_VNXe_ReplicationGroupLeaf
Member	Reference of EMC_VNXe_StorageVolumeLeaf
AssignedSequence	Indicates the relative position of members with in the collection.

EMC_VNXe_ReplicationGroup_RemoteStorageVolume_OrderedMemberOfCollectionAssocLeaf

Table 333 - Referenced properties/methods for EMC_VNXe_ReplicationGroup_RemoteStorageVolume_OrderedMemberOfCollectionAssocLeaf

CIM property	Description/notes
Collection	Reference of EMC_VNXe_ReplicationGroupLeaf
Member	Reference of EMC_VNXe_RemoteStorageVolumeLeaf
AssignedSequence	Indicates the relative position of members with in the collection.

EMC_VNXe_SnapGroup_SnapVolume_OrderedMemberOfCollectionAssocLeaf

Table 334 - Referenced properties/methods for EMC_VNXe_SnapGroup_SnapVolume_OrderedMemberOfCollectionAssocLeaf

CIM property	Description/notes
Collection	Reference of EMC_VNXe_SnapGroupLeaf
Member	Reference of EMC_VNXe_SnapVolumeLeaf
AssignedSequence	Indicates the relative position of members with in the collection.

Masking and Mapping Subprofile

Overview

Masking and Mapping Subprofile provides an interface for the administrator to specify which initiators can access what volumes through which target ports. The effect is that the given volume is only visible to SCSI commands that originate from the specified initiators through specific sets of target ports. The ability to limit access is called LUN masking; the ability to specify the device address seen by particular initiators is called LUN mapping.

Given a storage system with no LUN masking or mapping, all hosts/initiators see the same elements when they discover a storage system. In a storage system supporting LUN masking, logical units are masked (hidden) from SCSI initiators (Host Bus Adaptors) by default. The administrator uses the Masking and Mapping Subprofile to determine which logical units are visible (exposed) to specific initiators through which target ports. The LUN masking and mapping interfaces allow an administrator to customize the “view” of elements that are discovered. The effect is that the real storage system appears to be a number of subsets - each subset exposing a view customized for a particular set of initiators.

The management model is built on these “views” of a storage system - each view is a subset of components the administrator exposes to certain hosts - and the classes that model the authorization and access rights.

The key concepts for masking and mapping are view and path. A “view” is a list of logical units exposed to a list of initiators through a list of target ports, modeled as SCSIProtocolController (SPC) with associated LogicalDevices, StorageHardwareIDs, and SCSIProtocolEndpoints. The logical devices have logical unit numbers and access permissions relative to the view, modeled as DeviceNumber and DeviceAccess properties of the ProtocolControllerForUnit association. A full “path” is a combination of one each logical unit, initiator port, and target port - the concept of path is independent from a CIM model, but a view expresses a combinations of paths that comply with SCSI rules. In essence, an SPC serves as a collection of paths - each initiator ID is granted access to each logical unit through each target port.

Parameter name	Qualifier	Type	Description/note
Job	OUT	CIM_ConcreteJob Ref	Reference to the concrete job. NOTE: <ul style="list-style-type: none"> Client should not use this output since this method is supported in synchronous mode.
LUNames	IN, NULL allowed	string []	Array of IDs of logical unit instances (StorageVolume).
InitiatorPortIDs	IN, NULL allowed	string []	Array of IDs of initiator ports.
TargetPortIDs	IN, NULL allowed	string []	Array of IDs of target ports.
DeviceNumbers	IN, NULL allowed	string []	List of logical unit numbers to assign to the corresponding logical unit in the LUNames parameter.
DeviceAccesses	IN, NULL allowed	uint16 []	List of permissions to assign to the corresponding logical unit in the LUNames parameter. NOTE: <ul style="list-style-type: none"> Only support '2: Read Write'.
ProtocolControllers	IN, OUT, NULL allowed	CIM_SCSIProtocolControllerRef []	Array of references to SCSIProtocolControllers (SPCs).

- Parameter Constraints**

Table below shows valid use cases of ExposePaths supported by VNXe system and corresponding parameter constraints.

Table 336 - ExposePaths Use Cases and Input Parameter Constraints

Input Parameters / Use Cases	LUNNames	InitiatorPortIDs	TargetPortIDs	DeviceNumbers	DeviceAccesses	ProtocolControllers
Create a new view (Create a new Host)	See note 1	See note 1	NULL See note 2	See note 3	See note 4	NULL
Add LUs to a view (Expose LUNs to Host)	Mandatory	NULL	NULL	See note 3	Mandatory See note 4	Should only contain one SPC reference.

Input Parameters / Use Cases	LUNNames	InitiatorPortIDs	TargetPortIDs	DeviceNumbers	DeviceAccesses	ProtocolControllers
Add initiator port IDs to a view (Register Initiators to Host)	NULL	Mandatory	NULL	NULL	NULL	Should only contain one SPC reference.
Add target port IDs to a view (Add Targets to Host)	Not supported, See note 2					
<ol style="list-style-type: none"> 1. Depends on values of <i>SPCAAllowNoLUs</i> and <i>SPCAAllowNoInitiators</i> in <i>ProtocolControllerMaskingCapabilities</i>. 2. <i>TargetPortIDs</i> shall be null and <i>TargetPortIDs</i> manipulation is not supported since <i>ProtocolControllerMaskingCapabilities.PortsPerView</i> is set to "All ports share same view". 3. <i>DeviceNumbers</i> shall either be null or have same number of elements as <i>LUNNames</i> since <i>ProtocolControllerMaskingCapabilities.ClientSelectableDeviceNumbers</i> is set to true. 4. <i>DeviceAccesses</i> must have the same number of elements as <i>LUNNames</i>. 						

- **Return Results**

Table 337 - Possible return code of EMC_VNXe_ControllerConfigurationServiceLeaf.ExposePaths

Return code	Type	Description
0	uint32	Success
4	uint32	Failed: <ul style="list-style-type: none"> • Create a new view with initiators already added into an existing view. • Add initiators to a view but initiators already added into an existing view. • Any error that occurs or exception thrown when calling the underlying component.
5	uint32	Invalid Parameter: <ul style="list-style-type: none"> • Parameter combination conflicts with constraints (Refer to Parameters Constraints).
4097	uint32	Invalid logical unit ID
4098	uint32	Invalid initiator port ID
4100	uint32	Invalid permission
NA	CIM_Error	Exceptions will be thrown when any of the below cases is met:

Return code	Type	Description
		<ul style="list-style-type: none"> Any of the required parameters is NULL. Any of input parameters is not of the right CIM/SNIA type defined in the MOF files. Any of the input references cannot be found in the system.

Method: *ControllerConfigurationService.HidePaths*

• **Description**

This method allows the client to hide a list of SCSI logical units (*StorageVolume*) to a list of initiators (*StorageHardwareID*) on a SPC (*LunMaskingProtocolController*).

• **Parameters**

Table 338 - Signature and Parameters of EMC_VNXe_ControllerConfigurationServiceLeaf.ExposePaths

Parameter name	Qualifier	Type	Description/note
Job	OUT	CIM_ConcreteJob Ref	Reference to the concrete job. NOTE: <ul style="list-style-type: none"> Client should not use this output since this method is supported in synchronous mode.
LUNames	IN, NULL allowed	string []	Array of IDs of logical unit instances (<i>StorageVolume</i>).
InitiatorPortIDs	IN, NULL allowed	string []	Array of IDs of initiator ports.
TargetPortIDs	IN, NULL allowed	string []	Array of IDs of target ports.
ProtocolControllers	IN, OUT, NULL allowed	CIM_SCSIProtocolControllerRef []	Array of references to SCSIProtocolControllers (SPCs).

• **Parameter Constraints**

Table below shows valid use cases of *HidePaths* supported by VNXe system and corresponding parameter constraints.

Table 339 - HidePaths Use Cases and Input Parameter Constraints

Input Parameters / Use Cases	LUNNames	InitiatorPortIDs	TargetPortIDs	ProtocolControllers (on input)
Remove LUs from a view (Detach LUNs from Host)	Mandatory	NULL	NULL	Mandatory

Input Parameters / Use Cases	LUNNames	InitiatorPortIDs	TargetPortIDs	ProtocolControllers (on input)
Remove initiator IDs from a view (Unregister Initiators from Host)	NULL	Mandatory	NULL	Mandatory
Remove target ports to a view (Remove Target from Host)	Not Supported, See note 1			
Hide full paths from a view	Not Supported			
1. <i>TargetPortIDs</i> manipulation is not supported since <i>ProtocolControllerMaskingCapabilities.PortsPerView</i> is set to "All ports share same view".				

- **Return Results**

Table 340 - Possible return code of EMC_VNXe_ControllerConfigurationServiceLeaf.HidePaths

Return code	Type	Description
0	uint32	Success
4	uint32	Failed: <ul style="list-style-type: none"> • Any error that occurs or exception thrown when calling the underlying component.
5	uint32	Invalid Parameter: <ul style="list-style-type: none"> • Parameter combination conflicts with constraints (Refer to Parameter Constraints).
4097	uint32	Invalid logical unit ID
4098	uint32	Invalid initiator port ID
NA	CIM_Error	Exceptions will be thrown when any of the below cases is met: <ul style="list-style-type: none"> • Any of the required parameters is NULL. • Any of input parameters is not of the right CIM/SNIA type defined in the MOF files. • Any of the input references cannot be found in the system.

Method: ControllerConfigurationService.DeleteProtocolController

- **Description**

This method allows the client to delete an instance of *ProtocolController* and all associations in which this *ProtocolController* is referenced.

- **Parameters**

Table 341 - Signature and Parameters of EMC_VNXe_ControllerConfigurationServiceLeaf.ExposePaths

Parameter name	Qualifier	Type	Description/note
ProtocolController	IN	CIM_ProtocolControllerRef	Reference to the ProtocolController to be deleted.
DeleteChildrenProtocolControllers	IN, NULL allowed, Ignored	boolean	
DeleteUnits	IN, NULL allowed, Not Supported	boolean	NOTE: <ul style="list-style-type: none"> Client should leave this parameter NULL or set it to FALSE. Current SMI-S Provider does not support delete LogicalDevice at the same time.

- **Return Results**

Table 342 - Possible return code of EMC_VNXe_ControllerConfigurationServiceLeaf.HidePaths

Return code	Type	Description
0	uint32	Success
4	uint32	Failed: <ul style="list-style-type: none"> Any error that occurs or exception thrown when calling the underlying component.
5	uint32	Invalid Parameter: <ul style="list-style-type: none"> ProtocolController is invalid.
NA	CIM_Error	Exceptions will be thrown when any of the below cases is met: <ul style="list-style-type: none"> Any of the required parameters is NULL. Any of input parameters is not of the right CIM/SNIA type defined in the MOF files. Any of the input references cannot be found in the system.

Extrinsic Methods on StorageHardwareIDManagementService

None of extrinsic methods is supported on *StorageHardwareIDManagementService*.

Extrinsic Methods on PrivilegeManagementService

None of extrinsic methods is supported on *PrivilegeManagementService*.

Client considerations

Model Specification

- Host Access Support

Masking and Mapping (host access) configuration through VNXe SMI-S API can **ONLY** apply to standalone-LUN and LUN in LUN Group.

- Target Port Manipulation
Explicitly managing target port is **NOT** supported by VNXe SMI-S Provider.
Target ports will be automatically connected to corresponding SPCs per paths reported by underlying components.

Use case: Create a New Host

This use case describes how to create a new Host (SPC).

1. From Block Server (Top-level Array ComputerSystem), traverse *CIM_HostedServices* to *CIM_ControllerConfigurationService* to locate hosted controller configuration service.
2. Invoke method *CIM_ControllerConfigurationService.ExposePaths()* to create a new Host, by passing following parameters:
(Refer to [ExposePaths Parameter Constraints](#) for detailed information.)
3. Check output parameter *ProtocolController* to get the newly created Host.

Use case: Register Initiators to a Host

This use case describes how to register initiators to an existing host.

1. From Block Server (Top-level Array ComputerSystem), traverse *CIM_HostedServices* to *CIM_ControllerConfigurationService* to locate hosted controller configuration service.
2. Invoke method *CIM_ControllerConfigurationService.ExposePaths()* to register Initiators to the Host, by **ONLY** passing following parameters:
ProtocolControllers – A one instance list containing the Host to which the Initiators are to be registered.
InitiatorPortIDs – ID list of Initiators to be registered.
(Refer to [ExposePaths Parameter Constraints](#) for detailed information.)
3. If succeed, new *CIM_AssociatedPrivilege* associations will be built between the Host and Initiators.

Use case: Expose LUs to a Host

This use case describes how to expose LUNs to an existing host.

1. From Block Server (Top-level Array ComputerSystem), traverse *CIM_HostedServices* to *CIM_ControllerConfigurationService* to locate hosted controller configuration service.
2. Invoke method *CIM_ControllerConfigurationService.ExposePaths()* to expose LUNs (StorageVolume) to the Host, by passing following parameters:
ProtocolControllers – A one instance list containing the Host to which the LUNs are to be exposed.
LUNames – ID list of LUNs to be exposed.
DeviceAccesses – List of access privileges corresponding with *LUNames*.
(Refer to [ExposePaths Parameter Constraints](#) for detailed information.)
3. If succeed, *CIM_ProtocolControllerForUnit* associations will be built between the Host and LUNs.

Use case: Unregister InitiatorPortIDs from a Host

This use case describes how to unregister initiators from a host.

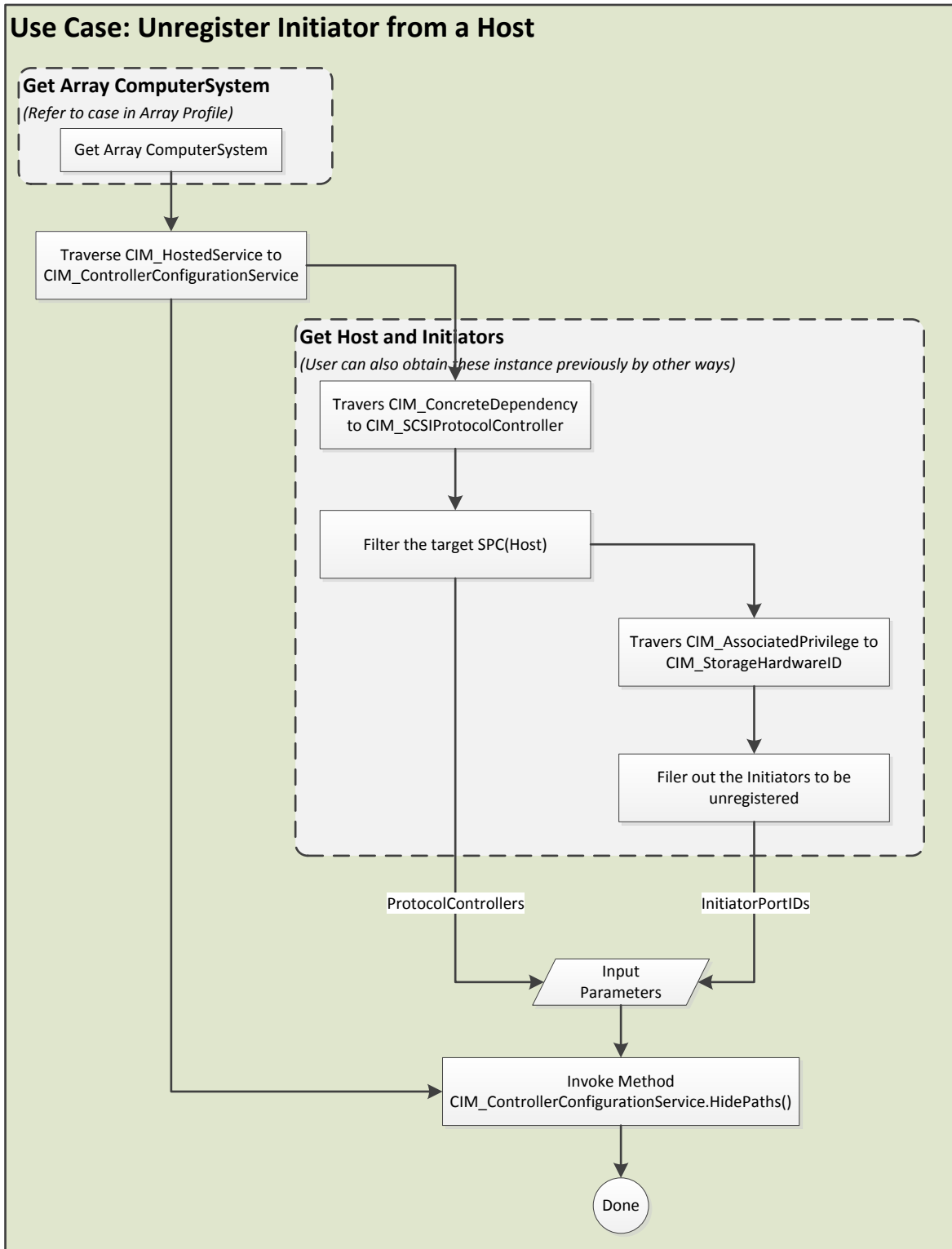


Figure 35 - Flowchart of Unregistering Initiators from Host

Use case: Detach LUs from a Host

This use case describes how to detach LUNs from a host.

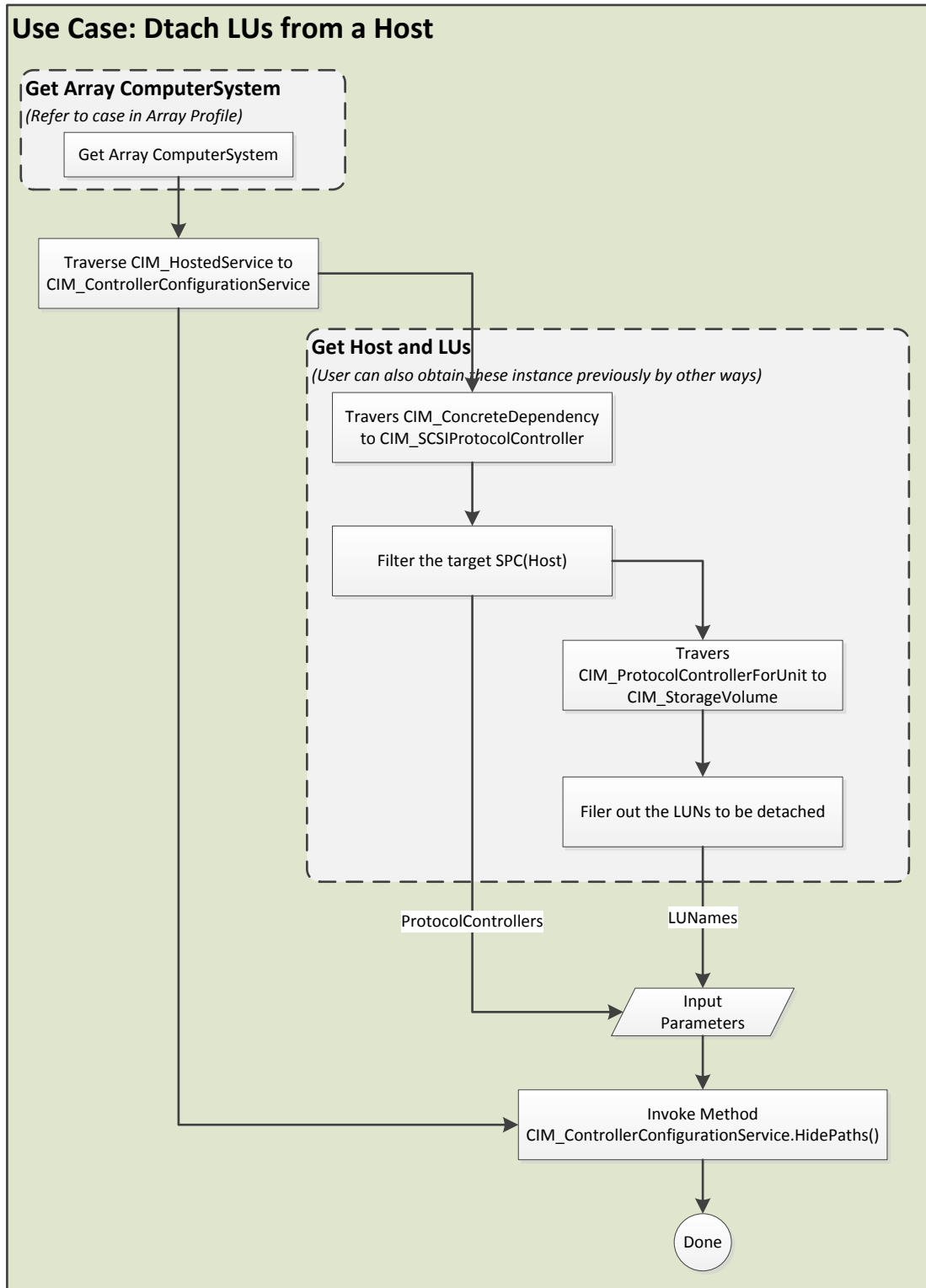


Figure 36 - Flowchart of Detaching LUs from Host

Use case: Delete a Host

This use case describes how to delete a host.

1. From Block Server (Top-level Array ComputerSystem), traverse *CIM_HostedServices* to *CIM_ControllerConfigurationService* to locate hosted controller configuration service.
2. Invoke method *CIM_ControllerConfigurationService.DeleteProtocolController()* to delete the Host, by passing following parameters:

ProtocolController – Reference to the Host to be deleted.

Use case: Rename a Host

Rename Host is supported via intrinsic method *ModifyInstance*. User can modify *ElementName* of *EMC_VNXe_LunMaskingProtocolControllerLeaf* to rename a Host.

CIM Element

The implemented classes and associations related to Masking and Mapping Subprofile in VNXe Storage System are described as follows:

Table 343 - CIM Elements implemented in VNXe for Masking and Mapping Subprofile

CIM Class	Implemented Class	Description
CIM_AssociatedPrivilege	EMC_VNXe_StorageHardwareID_LunMaskingProtocolController_AssociatedPrivilegeAssocLeaf	Associates the Host and Initiator registered in the Host.
CIM_AuthorizedPrivilege	EMC_VNXe_AuthorizedPrivilegeLeaf	Represents the privilege authorized. Deprecated
CIM_AuthorizedSubject	EMC_VNXe_StorageHardwareID_AuthorizedPrivilege_AuthorizedSubjectAssocLeaf	Associates the Initiator and the AuthorizedPrivilege granted or denied to the Initiator. Deprecated
CIM_AuthorizedTarget	EMC_VNXe_LunMaskingProtocolController_AuthorizedPrivilege_AuthorizedTargetAssocLeaf	Associates the TargetPort and the AuthorizedPrivilege applied to the TargetPort. Deprecated
CIM_ConcreteDependency	EMC_VNXe_ControllerConfigurationService_LunMaskingProtocolController_ConcreteDependencyAssocLeaf	Associates the controller configuration service and the Host(SPC).
CIM_ConcreteDependency	EMC_VNXe_PrivilegeManagementService_AuthorizedPrivilege_ConcreteDependencyAssocLeaf	Associates the privilege management service and the privileges.
CIM_ConcreteDependency	EMC_VNXe_StorageHardwareIDManagementService_StorageHardwareID_ConcreteDependencyAssocLeaf	Associates the storage hardware ID management service and the Initiators.

CIM Class	Implemented Class	Description
CIM_ConcreteDependency	EMC_VNXe_StorageHardwareIDManagementService_SystemSpecificCollection_ConcreteDependencyAssocLeaf	Associates the storage hardware ID management service and the collection of Initiators
CIM_ControllerConfigurationService	EMC_VNXe_ControllerConfigurationServiceLeaf	Represents the controller configuration service.
CIM_ElementCapabilities	EMC_VNXe_StorageSystem_ProtocolControllerMaskingCapabilities_ElementCapabilitiesAssocLeaf	Associates the system and its masking capabilities.
CIM_ElementSettingData	EMC_VNXe_StorageSystem_StorageClientSettingData_ElementSettingDataAssocLeaf	Associates the system and its storage client setting data.
CIM_ElementSettingData	EMC_VNXe_StorageHardwareID_StorageClientSettingData_ElementSettingDataAssocLeaf	Associates the Initiator and its storage client setting data
CIM_HostedCollection	EMC_VNXe_StorageSystem_SystemSpecificCollection_HostedCollectionAssocLeaf	Associates the system and the collection of Initiators hosted by the system.
CIM_HostedService	EMC_VNXe_StorageSystem_ControllerConfigurationService_HostedServiceAssocLeaf	Associates the hosting system and the controller configuration service hosted by the system.
CIM_HostedService	EMC_VNXe_StorageSystem_StorageHardwareIDManagementService_HostedServiceAssocLeaf	Associates the hosting system and the storage hardware ID management service hosted by the system.
CIM_HostedService	EMC_VNXe_StorageSystem_PrivilegeManagementService_HostedServiceAssocLeaf	Associates the hosting system and the privilege management service hosted by the system.
CIM_MemberOfCollection	EMC_VNXe_SystemSpecificCollection_StorageHardwareID_MemberOfCollectionAssocLeaf	Associates the Initiators and the collection of the Initiators.
CIM_PrivilegeManagementService	EMC_VNXe_PrivilegeManagementServiceLeaf	Represents the system provided privilege management service
CIM_ProtocolControllerForUnit	EMC_VNXe_LunMaskingProtocolController_StorageVolume_ProtocolControllerForUnitAssocLeaf	Associates the LUNs and the Host to which the LUNs are exposed.
CIM_ProtocolControllerMaskingCapabilities	EMC_VNXe_ProtocolControllerMaskingCapabilitiesLeaf	Represents the capabilities of masking.
CIM_SAPAvailableForElement	EMC_VNXe_iSCSIProtocolEndpoint_LunMaskingProtocolController_SAPAvailableForElementAssocLeaf	Associates the TargetPort (iSCSI) and the Host.
CIM_SAPAvailableForElement	EMC_VNXe_FCSCSIProtocolEndpoint_LunMaskingProtocolController_SAPAvailableFo	Associates the TargetPort (FCSCSI) and the Host.

CIM Class	Implemented Class	Description
	rElementAssocLeaf	
CIM_SCSIProtocolController	EMC_VNXe_LunMaskingProtocolControllerLeaf	Represents the Host.
CIM_SCSIProtocolEndpoint	EMC_VNXe_FCSCSIProtocolEndpointLeaf	Represents the TargetPort (FCSCSI).
CIM_SCSIProtocolEndpoint	EMC_VNXe_iSCSIProtocolEndpointLeaf	Represents the TargetPort (iSCSI).
CIM_StorageClientSettingData	EMC_VNXe_StorageClientSettingDataLeaf	Represents the storage client setting data.
CIM_StorageHardwareID	EMC_VNXe_StorageHardwareIDLeaf	Represents the Initiator.
CIM_StorageHardwareIDManagementService	EMC_VNXe_StorageHardwareIDManagementServiceLeaf	Represents the storage hardware ID management service.
CIM_SystemSpecificCollection	EMC_VNXe_SystemSpecificCollectionLeaf	Represents the collection of Initiators.

EMC_VNXe_ControllerConfigurationServiceLeaf

Table 344 - Referenced properties/methods for EMC_VNXe_ControllerConfigurationServiceLeaf

CIM property	Description/notes
SystemCreationClassName	Set to "EMC_VNXe_StorageSystemLeaf"
SystemName	Set as system name of VNXe
CreationClassName	Set to "EMC_VNXe_ControllerConfigurationServiceLeaf"
Name	Set to 'ControllerConfigurationService'

EMC_VNXe_PrivilegeManagementServiceLeaf

Table 345 - Referenced properties/methods for EMC_VNXe_PrivilegeManagementServiceLeaf

CIM property	Description/notes
SystemCreationClassName	Set to "EMC_VNXe_StorageSystemLeaf"
SystemName	Set as system name of VNXe
CreationClassName	Set to "EMC_VNXe_PrivilegeManagementService"
Name	Set to 'PrivilegeManagementService'

EMC_VNXe_StorageHardwareIDManagementServiceLeaf

**Table 346 - Referenced properties/methods for
EMC_VNXe_StorageHardwareIDManagementServiceLeaf**

CIM property	Description/notes
SystemCreationClassName	Set to "EMC_VNXe_StorageSystemLeaf"
SystemName	Set as system name of VNXe
CreationClassName	Set to "EMC_VNXe_StorageHardwareIDManagementService"
Name	Set to 'StorageHardwareIDManagementService'

EMC_VNXe_ProtocolControllerMaskingCapabilitiesLeaf

**Table 347 - Referenced properties/methods for
EMC_VNXe_ProtocolControllerMaskingCapabilitiesLeaf**

CIM property	Description/notes
InstanceID	Unique ID for this instance.
ElementName	User friendly name for this instance.
ValidHardwareIDTypes	A list of the valid values for StorageHardwareID.IDType. Set to [2: Port WWN, 5: iSCSI Name]
PortsPerView	Indicates the way that ports per view (ProtocolController) are handled by the underlying storage system. Set to '4: All Ports share the same View'
ClientSelectableDeviceNumbers	Indicates if this storage system allows the client to specify the DeviceNumber parameter when calling ControllerConfigurationService.AttachDevice() or specify the DeviceNumbers parameter when calling ControllerConfigurationService.ExposePaths(). Set to TRUE.
OneHardwareIDPerView	Indicates if this storage system limits configurations to a single subject hardware ID per view. Set to FALSE.
PrivilegeDeniedSupported	Indicates if this storage system allows a client to create a Privilege instance with PrivilegeGranted set to FALSE. Set to FALSE.
UniqueUnitNumbersPerPort	When set to false, different ProtocolContollers attached to a LogicalPort can expose the same unit numbers. If true, then this storage system requires unique unit numbers across all the ProtocolControllers connected to a LogicalPort. Set to FALSE.

CIM property	Description/notes
MaximumMapCount	The maximum number of ProtocolControllerForUnit associations that can be associated with a single LogicalDevice. Set to 0.
SPCAllowsNoLUs	Indicates if the instrumentation allows a client to create a configuration where an SPC has no LogicalDevices associated. Set to TRUE.
SPCAllowsNoTargets	Indicates if the instrumentation allows a client to create a configuration where an SPC has no target SCSIProtocolEndpoints associated. Set to TRUE.
SPCAllowsNoInitiators	Indicates if the instrumentation allows a client to create a configuration where an SPC has no StorageHardwareIDs associated. Set to TRUE.
SPCSupportsDefaultViews	Indicates if the instrumentation supports 'default view' SPCs that exposes logical units to all initiators. Set to FALSE.
CreateProtocolControllerSupported	Set to FALSE.
ExposePathsSupported	Set to TRUE.

EMC_VNXe_LunMaskingProtocolControllerLeaf

Table 348 - Referenced properties/methods for EMC_VNXe_LunMaskingProtocolControllerLeaf

CIM property	Description/notes
SystemCreationClassName	Set to 'EMC_VNXe_StorageSystemLeaf'
SystemName	Set as name of the system.
CreationClassName	Set to 'EMC_VNXe_LunMaskingProtocolControllerLeaf'
DeviceID	FriendlyID of the Host (SPC).
Name	FriendlyID of the Host (SPC).
ElementName	Name of the Host (SPC).
NameFormat	Set to ': Other'

EMC_VNXe_StorageHardwareIDLeaf

Table 349 - Referenced properties/methods for EMC_VNXe_StorageHardwareIDLeaf

CIM property	Description/notes
InstanceID	Unique ID for this instance.

CIM property	Description/notes
StorageID	The hardware worldwide unique ID.
IDType	The type of the ID property.

EMC_VNXe_AuthorizedPrivilegeLeaf

Table 350 - Referenced properties/methods for EMC_VNXe_AuthorizedPrivilegeLeaf

CIM property	Description/notes
InstanceID	Unique ID for this instance.
PrivilegeGranted	Indicates whether the Privilege is granted or denied.
Activities	Indicates the activities that are granted or denied.

EMC_VNXe_FCSCSIProtocolEndpointLeaf

For detailed information, refer to [EMC_VNXe_FCSCSIProtocolEndpointLeaf](#) in Block Services Package.

EMC_VNXe_iSCSIProtocolEndpointLeaf

For detailed information, refer to [EMC_VNXe_iSCSIProtocolEndpointLeaf](#) in Block Services Package.

EMC_VNXe_SystemSpecificCollectionLeaf

Table 351 - Referenced properties/methods for EMC_VNXe_SystemSpecificCollectionLeaf

CIM property	Description/notes
InstanceID	Unique ID for this instance.
ElementName	User friendly name for this instance.

EMC_VNXe_StorageClientSettingDataLeaf

Table 352 - Referenced properties/methods for EMC_VNXe_StorageClientSettingDataLeaf

CIM property	Description/notes
InstanceID	Unique ID for this instance.
ElementName	User friendly name for this instance.
ClientTypes	Defines operating system, version, driver, and other host environment factors that influence the behavior exposed by storage systems.
OtherClientTypeDescriptions	Set to 'Operation Systems are not differentiated'

EMC_VNXe_StorageSystem_ControllerConfigurationService_HostedServiceAssocLeaf

Table 353 - Referenced properties/methods for EMC_VNXe_StorageSystem_ControllerConfigurationService_HostedServiceAssocLeaf

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_StorageSystemLeaf
Dependent	Reference of EMC_VNXe_ControllerConfigurationServiceLeaf

EMC_VNXe_StorageSystem_PrivilegeManagementService_HostedServiceAssocLeaf

**Table 354 - Referenced properties/methods for
EMC_VNXe_StorageSystem_PrivilegeManagementService_HostedServiceAssocLeaf**

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_StorageSystemLeaf
Dependent	Reference of EMC_VNXe_PrivilegeManagementServiceLeaf

EMC_VNXe_StorageSystem_StorageHardwareIDManagementService_HostedServiceAssocLeaf

**Table 355 - Referenced properties/methods for
EMC_VNXe_StorageSystem_StorageHardwareIDManagementService_HostedServiceAssocLeaf**

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_StorageSystemLeaf
Dependent	Reference of EMC_VNXe_StorageHardwareIDManagementServiceLeaf

EMC_VNXe_StorageSystem_ProtocolControllerMaskingCapabilities_ElementCapabilitiesAssocLeaf

**Table 356 - Referenced properties/methods for
EMC_VNXe_StorageSystem_ProtocolControllerMaskingCapabilities_ElementCapabilitiesAssocLeaf**

CIM property	Description/notes
ManagedElement	Reference of EMC_VNXe_StorageSystemLeaf
Capabilities	Reference of EMC_VNXe_ProtocolControllerMaskingCapabilitiesLeaf

EMC_VNXe_StorageSystem_SystemSpecificCollection_HostedCollectionAssocLeaf

**Table 357 - Referenced properties/methods for
EMC_VNXe_StorageSystem_SystemSpecificCollection_HostedCollectionAssocLeaf**

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_StorageSystemLeaf
Dependent	Reference of EMC_VNXe_SystemSpecificCollectionLeaf

EMC_VNXe_StorageSystem_StorageClientSettingData_ElementSettingDataAssocLeaf

**Table 358 - Referenced properties/methods for
EMC_VNXe_StorageSystem_StorageClientSettingData_ElementSettingDataAssocLeaf**

CIM property	Description/notes
ManagedElement	Reference of EMC_VNXe_StorageSystemLeaf
SettingData	Reference of EMC_VNXe_StorageClientSettingDataLeaf

EMC_VNXe_SystemSpecificCollection_StorageHardwareID_MemberOfCollectionAssocLeaf

**Table 359 - Referenced properties/methods for
EMC_VNXe_SystemSpecificCollection_StorageHardwareID_MemberOfCollectionAssocLeaf**

CIM property	Description/notes
Collection	Reference of EMC_VNXe_SystemSpecificCollectionLeaf
Member	Reference of EMC_VNXe_StorageHardwareIDLeaf

EMC_VNXe_LunMaskingProtocolController_AuthorizedPrivilege_AuthorizedTargetAssocLeaf

**Table 360 - Referenced properties/methods for
EMC_VNXe_LunMaskingProtocolController_AuthorizedPrivilege_AuthorizedTargetAssocLeaf**

CIM property	Description/notes
Privilege	Reference of EMC_VNXe_AuthorizedPrivilegeLeaf
TargetElement	Reference of EMC_VNXe_LunMaskingProtocolControllerLeaf

EMC_VNXe_LunMaskingProtocolController_StorageVolume_ProtocolControllerForUnitAssocLeaf

**Table 361 - Referenced properties/methods for
EMC_VNXe_LunMaskingProtocolController_StorageVolume_ProtocolControllerForUnitAssocLeaf**

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_LunMaskingProtocolControllerLeaf
Dependent	Reference of EMC_VNXe_StorageVolumeLeaf
DeviceNumber	Address of the associated StorageVolume in the context of the ProtocolController.
DeviceAccess	The access rights granted to the referenced StorageVolume as exposed through the ProtocolController.

EMC_VNXe_StorageHardwareID_LunMaskingProtocolController_AssociatedPrivilegeAssocLeaf

Table 362 - Referenced properties/methods for EMC_VNXe_StorageHardwareID_LunMaskingProtocolController_AssociatedPrivilegeAssocLeaf

CIM property	Description/notes
Subject	Reference of EMC_VNXe_StorageHardwareIDLeaf
Target	Reference of EMC_VNXe_LunMaskingProtocolControllerLeaf
UseKey	Used to distinguish instances in case multiple instances of this association exist between the same Subject and Target. Set to 'Default'
PrivilegeGranted	Indicates whether the Privilege is granted or denied. Set to TRUE.
Activities	Activities granted.

EMC_VNXe_StorageHardwareID_AuthorizedPrivilege_AuthorizedSubjectAssocLeaf

Table 363 - Referenced properties/methods for EMC_VNXe_StorageHardwareID_AuthorizedPrivilege_AuthorizedSubjectAssocLeaf

CIM property	Description/notes
PrivilegedElement	Reference of EMC_VNXe_StorageHardwareIDLeaf
Privilege	Reference of EMC_VNXe_AuthorizedPrivilegeLeaf

EMC_VNXe_StorageHardwareID_StorageClientSettingData_ElementSettingDataAssocLeaf

Table 364 - Referenced properties/methods for EMC_VNXe_StorageHardwareID_StorageClientSettingData_ElementSettingDataAssocLeaf

CIM property	Description/notes
ManagedElement	Reference of EMC_VNXe_StorageHardwareIDLeaf
SettingData	Reference of EMC_VNXe_StorageClientSettingDataLeaf

EMC_VNXe_iSCSIProtocolEndpoint_LunMaskingProtocolController_SAPAvailableForElementAssocLeaf

Table 365 - Referenced properties/methods for EMC_VNXe_iSCSIProtocolEndpoint_LunMaskingProtocolController_SAPAvailableForElementAssocLeaf

CIM property	Description/notes
ManagedElement	Reference of EMC_VNXe_LunMaskingProtocolControllerLeaf

CIM property	Description/notes
AvailableSAP	Reference of EMC_VNXe_iSCSIProtocolEndpointLeaf

EMC_VNXe_FCSCSIProtocolEndpoint_LunMaskingProtocolController_SAPAvailableForElementAssocLeaf

Table 366 - Referenced properties/methods for EMC_VNXe_FCSCSIProtocolEndpoint_LunMaskingProtocolController_SAPAvailableForElementAssocLeaf

CIM property	Description/notes
ManagedElement	Reference of EMC_VNXe_LunMaskingProtocolControllerLeaf
AvailableSAP	Reference of EMC_VNXe_FCSCSIProtocolEndpointLeaf

EMC_VNXe_ControllerConfigurationService_LunMaskingProtocolController_ConcreteDependencyAssocLeaf

Table 367 - Referenced properties/methods for EMC_VNXe_ControllerConfigurationService_LunMaskingProtocolController_ConcreteDependencyAssocLeaf

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_ControllerConfigurationServiceLeaf
Dependent	Reference of EMC_VNXe_LunMaskingProtocolControllerLeaf

EMC_VNXe_PrivilegeManagementService_AuthorizedPrivilege_ConcreteDependencyAssocLeaf

Table 368 - Referenced properties/methods for EMC_VNXe_PrivilegeManagementService_AuthorizedPrivilege_ConcreteDependencyAssocLeaf

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_PrivilegeManagementServiceLeaf
Dependent	Reference of EMC_VNXe_AuthorizedPrivilegeLeaf

EMC_VNXe_StorageHardwareIDManagementService_StorageHardwareID_ConcreteDependencyAssocLeaf

Table 369 - Referenced properties/methods for EMC_VNXe_StorageHardwareIDManagementService_StorageHardwareID_ConcreteDependencyAssocLeaf

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_StorageHardwareIDManagementServiceLeaf

CIM property	Description/notes
Dependent	EMC_VNXe_StorageHardwareIDLeaf

[EMC_VNXe_StorageHardwareIDManagementService_SystemSpecificCollection_ConcreteDependencyAssocLeaf](#)

Table 370 - Referenced properties/methods for EMC_VNXe_StorageHardwareIDManagementService_SystemSpecificCollection_ConcreteDependencyAssocLeaf

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_StorageHardwareIDManagementServiceLeaf
Dependent	Reference of EMC_VNXe_SystemSpecificCollectionLeaf

NAS Head Profile

Overview

The NAS Head Profile exports file elements from a file system as file shares. The storage for the file system is obtained from external SAN storage. For example, a storage array that exports StorageVolumes as LUNs. The storage array may also provide storage to the other hosts or devices (or the other NAS Heads). The storage on the array might be visible to other external management tools, and may be managed actively and independently.

This profile models the necessary file system and NAS concepts, and defines how the connections to the underlying storage are managed.

The SMI-S Provider for the VNXe storage system implements the following subprofiles of the NAS Head Profile:

- Multiple Computer System
- Physical Package
- Software
- Health Package
- Block Services
- Extent Composition
- NAS Network Port
- File Storage
- File System
- File Export
- File Server Manipulation
- Filesystem Manipulation
- File Export Manipulation
- Simple Identity Management (DSP1034)

NOTE: For more details, refer to [Clause 13: NAS Head Profile in Storage Management Technical Specification, Part 4 Filesystems Version 1.6.0, Revision 4](#).

Class diagram

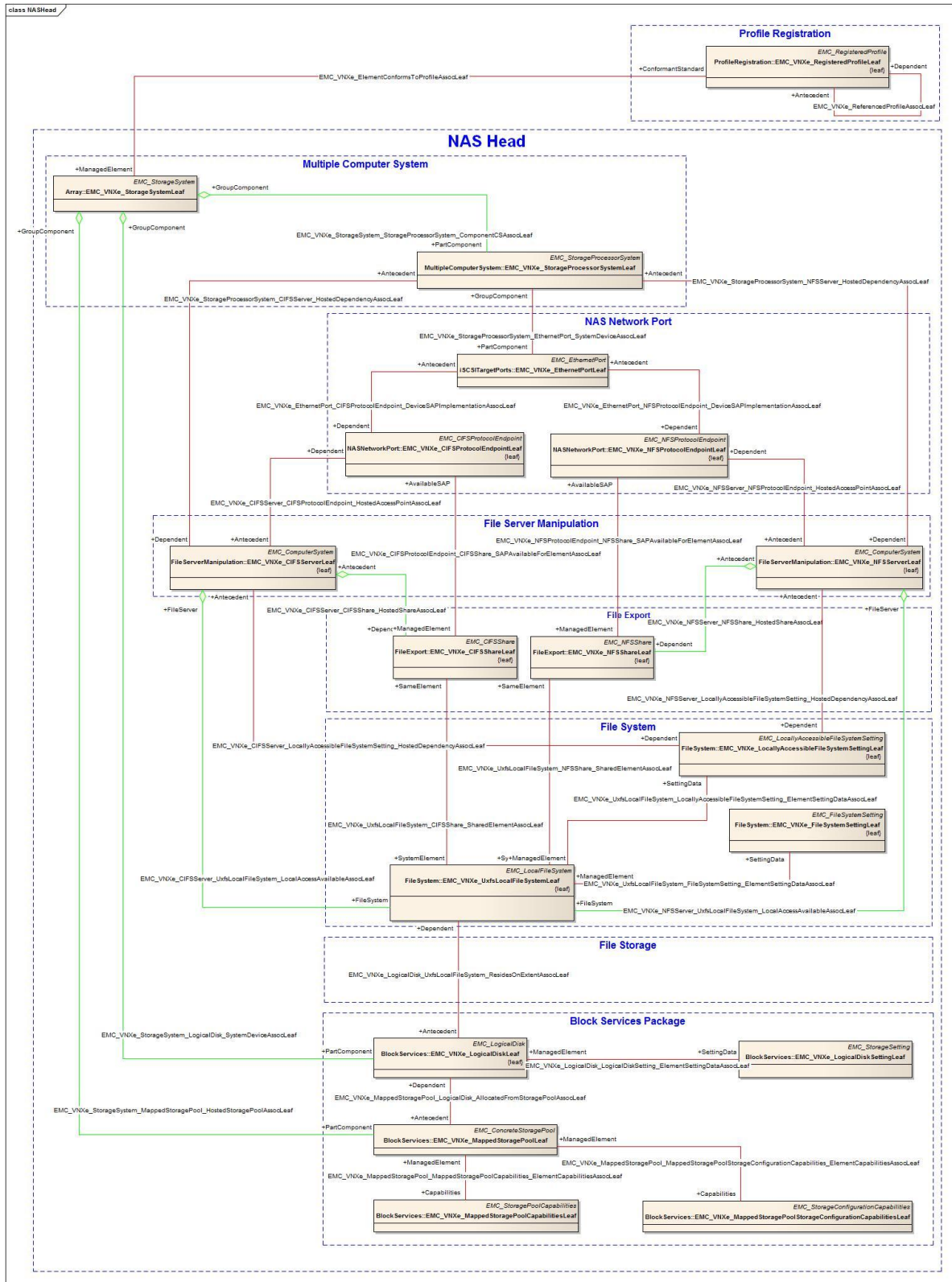


Figure 37 - NAS Head Profile class diagram

Method of the Profile

This profile does not include any extrinsic methods.

Client considerations

Use case: Discover NAS Head

This use case describes how to discover the NAS Head ComputerSystem and to retrieve the LogicalDisk, StoragePool, and Virtual File Server of a VNXe storage system.

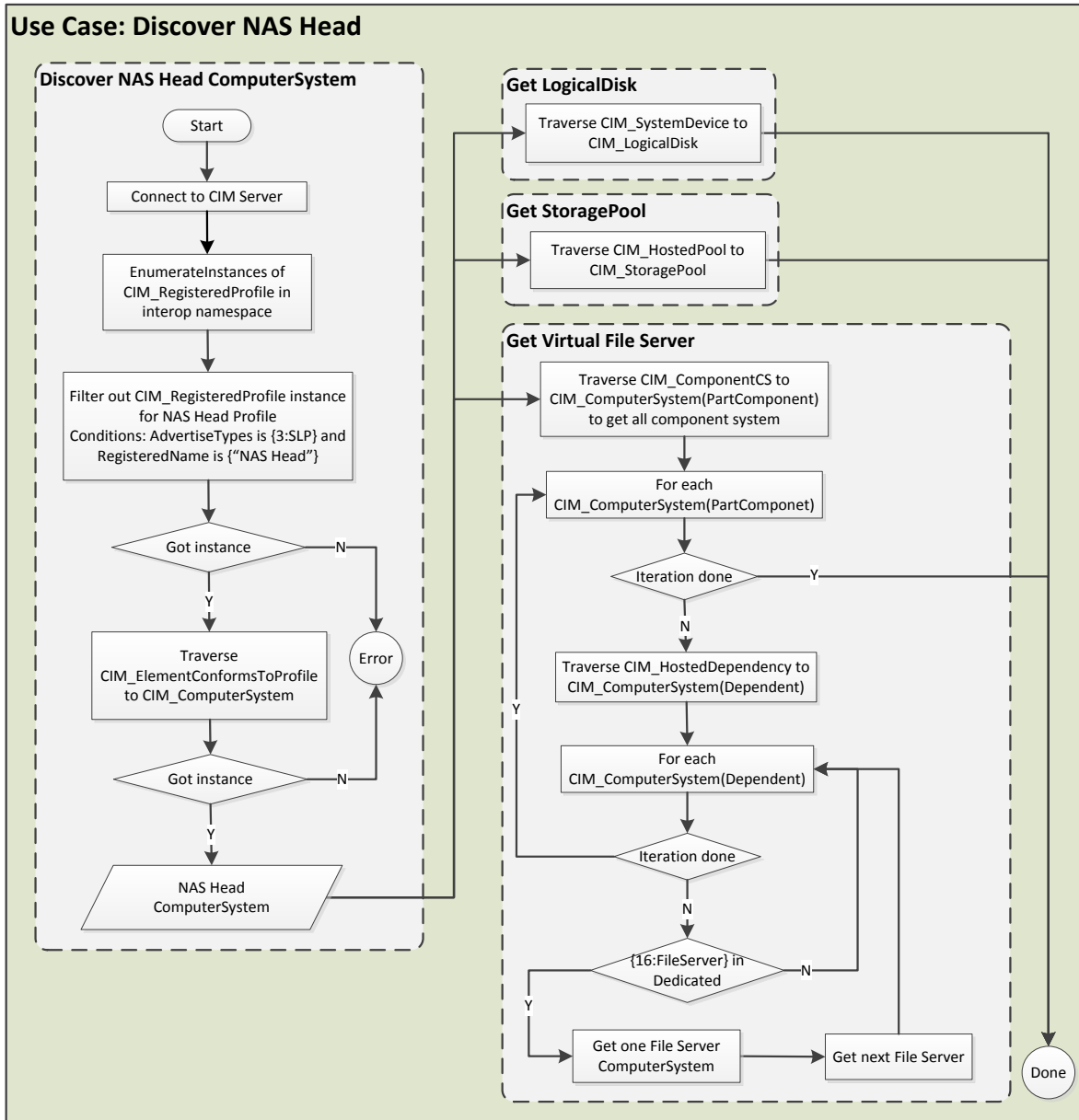


Figure 38 - Flowchart of NAS Head discovery

CIM Element

The implemented classes and associations related to the NAS Head Profile in the VNXe storage system are described as follows:

Table 371 - CIM Elements for NAS Head Profile

CIM Class	Implemented Class	Description
CIM_ComputerSystem (Top-level)	EMC_VNXe_StorageSystemLeaf	Represents the top-level ComputerSystem of NAS Head Profile.
CIM_ComputerSystem (Component)	EMC_VNXe_StorageProcessorSystemLeaf	Represents the storage processor system.
CIM_ComputerSystem (Virtual File Server)	EMC_VNXe_CIFSServerLeaf	Represents the virtual file server that supports the CIFS protocol.
CIM_ComputerSystem (Virtual File Server)	EMC_VNXe_NFSServerLeaf	Represents the virtual file server that supports the NFS protocol.
CIM_ComponentCS	EMC_VNXe_StorageSystem_StorageProcessorSystem_ComponentCSAssocLeaf	Associates the top-level and component ComputerSystems.
CIM_FileSystem	EMC_VNXe_UxfsLocalFileSystemLeaf	Represents the local file system.
CIM_HostedDependency	EMC_VNXe_StorageProcessorSystem_CIFSServer_HostedDependencyAssocLeaf	Associates the virtual CIFS file server and the component ComputerSystem that hosts the file server.
CIM_HostedDependency	EMC_VNXe_StorageProcessorSystem_NFSServer_HostedDependencyAssocLeaf	Associates the virtual NFS file server and the component ComputerSystem that hosts the file server.
CIM_HostedFileSystem	EMC_VNXe_StorageSystem_UxfsLocalFileSystem_HostedFileSystemAssocLeaf	Associates the local Filesystem and the Filesystem Host (NAS Head).
CIM_HostedStoragePool (To Concrete Pool)	EMC_VNXe_StorageSystem_MappedStoragePool_HostedStoragePoolAssocLeaf	Associates the concrete storage pool and the system hosting the pool.
CIM_HostedStoragePool (To Primordial Pool)	EMC_VNXe_StorageSystem_PrimordialPool_HostedStoragePoolAssocLeaf	Associates the primordial storage pool and the system hosting the pool.
CIM_LogicalDisk	EMC_VNXe_LogicalDiskLeaf	Represents the logical disk on which the local file system resides.
CIM_StorageExtent (Primordial)	EMC_VNXe_DiskExtentLeaf	Represents the primordial storage extent that makes up the storage pool.
CIM_StoragePool	EMC_VNXe_MappedStoragePool	Represents the concrete storage pool.

CIM Class	Implemented Class	Description
(Concrete)	olLeaf	
CIM_StoragePool (Primordial)	EMC_VNXe_PrimordialPoolLeaf	Represents the primordial storage pool.
CIM_SystemDevice (To Logical Disks)	EMC_VNXe_StorageSystem_LogicalDisk_SystemDeviceAssocLeaf	Associates the LogicalDisk and the system hosting it.
CIM_SystemDevice (To Primordial Extents)	EMC_VNXe_StorageSystem_DiskExtent_SystemDeviceAssocLeaf	Associates the primordial StorageExtent and the system hosting it.

[EMC_VNXe_StorageSystemLeaf](#)

For further information, refer to [EMC_VNXe_StorageSystemLeaf](#) in Multiple Computer System Subprofile.

[EMC_VNXe_StorageProcessorSystemLeaf](#)

For further information, refer to [EMC_VNXe_StorageProcessorSystemLeaf](#) in Multiple Computer System Subprofile.

[EMC_VNXe_CIFSServerLeaf](#)

For further information, refer to [EMC_VNXe_CIFSServerLeaf](#) in File Server Manipulation Subprofile.

[EMC_VNXe_NFSServerLeaf](#)

For further information, refer to [EMC_VNXe_NFSServerLeaf](#) in File Server Manipulation Subprofile.

[EMC_VNXe_LogicalDiskLeaf](#)

For further information, refer to [EMC_VNXe_LogicalDiskLeaf](#) in Block Services Package.

[EMC_VNXe_DiskExtentLeaf](#)

For further information, refer to [EMC_VNXe_DiskExtentLeaf](#) in Disk Drive Lite Subprofile.

[EMC_VNXe_PrimordialPoolLeaf](#)

For further information, refer to [EMC_VNXe_PrimordialPoolLeaf](#) in Block Services Package.

[EMC_VNXe_MappedStoragePoolLeaf](#)

For further information, refer to [EMC_VNXe_MappedStoragePoolLeaf](#) in Block Services Package.

[EMC_VNXe_UxfsLocalFileSystemLeaf](#)

For further information, refer to [EMC_VNXe_UxfsLocalFileSystemLeaf](#) in File System Profile.

NAS Network Port Profile

Overview

The NAS Network Port Profile models ProtocolEndpoints for file access (CIFS and NFS), TCP, IP and LAN endpoints. It also models the Network Port supported by the protocol endpoints. This profile provides basic information in the NAS models for addressing paths for accessing the NAS implementations for the purpose of data access (front-end ports).

The methods for manipulating these elements are covered by other profiles.

NOTE: For more details, refer to [Clause 15: NAS Network Port Profile in Storage Management Technical Specification, Part 4 Filesystems Version 1.6.0, Revision 4](#).

Class diagram

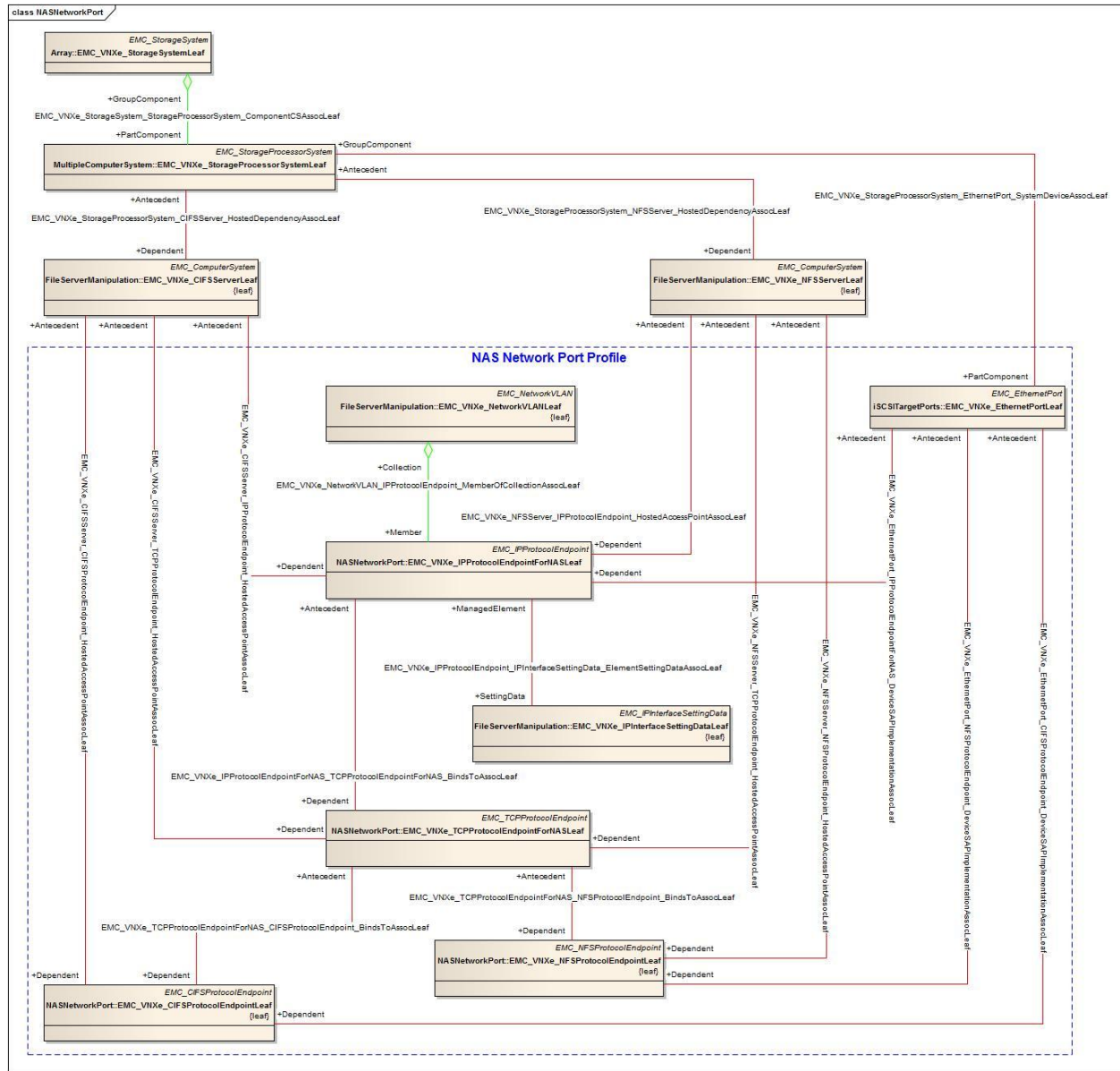


Figure 39 - NAS Network Port Profile class diagram

Methods of the Profile

This profile does not include any extrinsic methods.

Client considerations

N/A

CIM Elements

The implemented classes and associations related to NAS Network Port Profile in the VNXe storage system are described as follows:

Table 372 - CIM Elements for NAS Network Port Profile

CIM Class	Implemented Class	Description
CIM_BindsTo	EMC_VNXe_TCIPProtocolEndpointForNAS_CIFSProtocolEndpoint_BindsToAssocLeaf	Represents the association between a CIFS ProtocolEndpoint to an underlying TCPProtocolEndpoint.
CIM_BindsTo	EMC_VNXe_TCIPProtocolEndpointForNAS_NFSProtocolEndpoint_BindsToAssocLeaf	Represents the association between a NFS ProtocolEndpoint to an underlying TCPProtocolEndpoint.
CIM_BindsTo	EMC_VNXe_IPProtocolEndpointForNAS_TCIPProtocolEndpointForNAS_BindsToAssocLeaf	Represents the association between an IP ProtocolEndpoint to an underlying TCPProtocolEndpoint.
CIM_DeviceSAPImplementation	EMC_VNXe_EthernetPort_CIFSProtocolEndpoint_DeviceSAPImplementationAssocLeaf	Represents the association between an CIFS ProtocolEndpoint and the NetworkPort that it supports.
CIM_DeviceSAPImplementation	EMC_VNXe_EthernetPort_NFSProtocolEndpoint_DeviceSAPImplementationAssocLeaf	Represents the association between an NFS ProtocolEndpoint and the NetworkPort that it supports.
CIM_DeviceSAPImplementation	EMC_VNXe_EthernetPort_IPProtocolEndpointForNAS_DeviceSAPImplementationAssocLeaf	Represents the Ethernet Port an IP ProtocolEndpoint uses.
CIM_ElementSettingData	EMC_VNXe_IPProtocolEndpoint_IPInterfaceSettingData_ElementSettingDataAssocLeaf	Represents the association between an IP ProtocolEndpoint to its setting data.
CIM_HostedAccessPoint	EMC_VNXe_CIFSServer_CIFSProtocolEndpoint_HostedAccessPointAssocLeaf	Represents the CIFS ProtocolEndpoint on a CIFS server.
CIM_HostedAccessPoint	EMC_VNXe_NFSServer_NFSProtocolEndpoint_HostedAccessPointAssocLeaf	Represents the NFS ProtocolEndpoint on a NFS server.
CIM_HostedAccessPoint	EMC_VNXe_CIFSServer_TCIPProtocolEndpoint_HostedAccessPointAssocLeaf	Represents the TCP Protocol Endpoint a CIFS server uses.
CIM_HostedAccessPoint	EMC_VNXe_NFSServer_TCIPProtocolEndpoint_HostedAccessPointAssocLeaf	Represents the TCP Protocol Endpoint a NFS server uses.
CIM_HostedAccessPoint	EMC_VNXe_CIFSServer_IPProtocolEndpoint_HostedAccessPointAssocLeaf	Represents the association between a front end IPProtocolEndpoint and the CIFS server that hosts it.
CIM_HostedAccessPoint	EMC_VNXe_NFSServer_IPProtocolEndpoint_HostedAccessPoint	Represents the association between a front end IPProtocolEndpoint and the NFS server

CIM Class	Implemented Class	Description
	AssocLeaf	that hosts it.
CIM_MemberOfCollection	EMC_VNXe_NetworkVLAN_IPProtocolEndpoint_MemberOfCollectionAssocLeaf	Represents the association between a IP ProtocolEndpoint to its VLAN settings.
CIM_NetworkVLAN	EMC_VNXe_NetworkVLANLeaf	Represents the virtual LAN tag setting for an IP interface.
CIM_ProtocolEndpoint	EMC_VNXe_CIFSProtocolEndpointLeaf	Represents the front-end ProtocolEndpoint used to supported CIFS service.
CIM_ProtocolEndpoint	EMC_VNXe_NFSProtocolEndpointLeaf	Represents the front-end ProtocolEndpoint used to supported NFS service.
CIM_ProtocolEndpoint	EMC_VNXe_TCIPProtocolEndpointForNASLeaf	Represents the front-end ProtocolEndpoint used to supported TCP service.
CIM_ProtocolEndpoint	EMC_VNXe_IPProtocolEndpointForNASLeaf	Represents the front-end ProtocolEndpoint used to support the IP protocol services.
SNIA_IPInterfaceSettingData	EMC_VNXe_IPInterfaceSettingDataLeaf	Represents the setting data of an IP interface.

EMC_VNXe_IPProtocolEndpointForNASLeaf

Table 373 - Referenced properties/methods for EMC_VNXe_IPProtocolEndpointForNASLeaf

CIM property	Description/notes
SystemCreationClassName	Set to 'EMC_VNXe_CIFSServerLeaf' or 'EMC_VNXe_NFSServerLeaf'
SystemName	System name of the scoping FileServer.
CreationClassName	Set to 'EMC_VNXe_IPProtocolEndpointForNASLeaf'
Name	Name used to identify the protocol endpoint.
Description	Set to 'IP Protocol Endpoint'
OperationalStatus	Current operational status of the protocol endpoint.
HealthState	Current health stats of the protocol endpoint.
ProtocolIFType	Set to {4098: IPv4_v6}
IPv4Address	IPv4 address.
NameFormat	Set to '<servername>:if_<id>'
SubnetMask	Subnet mask of IPv4.
IPv6Address	IPv6 address

CIM property	Description/notes
PrefixLength	Prefix length of IPv6.

EMC_VNXe_TCPProtocolEndpointForNASLeaf

Table 374 - Referenced properties/methods for EMC_VNXe_TCPProtocolEndpointForNASLeaf

CIM property	Description/notes
SystemCreationClassName	Set to 'EMC_VNXe_CIFSServerLeaf' or 'EMC_VNXe_NFSServerLeaf'
SystemName	System name of the scoping FileServer.
CreationClassName	Set to 'EMC_VNXe_TCPProtocolEndpointForNASLeaf'
Name	Name used to identify the protocol endpoint.
Description	Set to 'TCP Protocol Endpoint'
OperationalStatus	Current operational status of the protocol endpoint.
ProtocolIFType	Set to {4111: TCP}
PortNumber	Access port for upper level protocol.
NameFormat	Set to '<serverName>:if_<id>:<PortNumber>'

EMC_VNXe_CIFSProtocolEndpointLeaf

Table 375 - Referenced properties/methods for EMC_VNXe_CIFSProtocolEndpointLeaf

CIM property	Description/notes
SystemCreationClassName	Set to 'EMC_VNXe_CIFSServerLeaf'
SystemName	System name of the scoping FileServer.
CreationClassName	Set to 'EMC_VNXe_CIFSProtocolEndpointLeaf'
Name	Name used to identify the protocol endpoint.
Description	Set to 'NFS Protocol Endpoint'
OperationalStatus	Current operational status of the protocol endpoint.
ProtocolIFType	Set to {4201: CIFS}
NameFormat	Set to 'if_<id>'

EMC_VNXe_NFSProtocolEndpointLeaf

Table 376 - Referenced properties/methods for EMC_VNXe_NFSProtocolEndpointLeaf

CIM property	Description/notes
SystemCreationClassName	Set to 'EMC_VNXe_NFSServerLeaf'
SystemName	System name of the scoping FileServer.
CreationClassName	Set to 'EMC_VNXe_NFSProtocolEndpointLeaf'
Name	Name used to identify the protocol endpoint.
Description	Set to 'NFS Protocol Endpoint'
OperationalStatus	Current operational status of the protocol endpoint.
ProtocolIFType	Set to {4200: CIFS}
NameFormat	Set to 'if_<id>'

[EMC_VNXe_IPInterfaceSettingDataLeaf](#)

For further information, refer to [EMC_VNXe_IPInterfaceSettingDataLeaf](#) in File Server Manipulation Subprofile.

[EMC_VNXe_NetworkVLANLeaf](#)

Table 377 - Referenced properties/methods for EMC_VNXe_NetworkVLANLeaf

CIM property	Description/notes
InstanceID	Unique ID for the instance.
VLANId	A 12-bit VLAN ID used in the VLAN tag header.

[EMC_VNXe_IPProtocolEndpoint_IPInterfaceSettingData_ElementSettingDataAssocLeaf](#)

Table 378 - Referenced properties/methods for EMC_VNXe_IPProtocolEndpoint_IPInterfaceSettingData_ElementSettingDataAssocLeaf

CIM property	Description/notes
ManagedElement	Reference of EMC_VNXe_IPProtocolEndpointForNASLeaf
SettingData	Reference of EMC_VNXe_IPInterfaceSettingDataLeaf

[EMC_VNXe_NetworkVLAN_IPProtocolEndpoint_MemberOfCollectionAssocLeaf](#)

Table 379 - Referenced properties/methods for EMC_VNXe_NetworkVLAN_IPProtocolEndpoint_MemberOfCollectionAssocLeaf

CIM property	Description/notes
Collection	Reference of EMC_VNXe_NetworkVLANLeaf
Member	Reference of EMC_VNXe_IPProtocolEndpointForNASLeaf

EMC_VNXe_TCPProtocolEndpointForNAS_CIFSProtocolEndpoint_BindsToAssocLeaf

**Table 380 - Referenced properties/methods for
EMC_VNXe_TCPProtocolEndpointForNAS_CIFSProtocolEndpoint_BindsToAssocLeaf**

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_TCPProtocolEndpointForNASLeaf
Dependent	Reference of EMC_VNXe_CIFSProtocolEndpointLeaf

EMC_VNXe_TCPProtocolEndpointForNAS_NFSProtocolEndpoint_BindsToAssocLeaf

**Table 381 - Referenced properties/methods for
EMC_VNXe_TCPProtocolEndpointForNAS_NFSProtocolEndpoint_BindsToAssocLeaf**

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_TCPProtocolEndpointForNASLeaf
Dependent	Reference of EMC_VNXe_NFSProtocolEndpointLeaf

EMC_VNXe_IPProtocolEndpointForNAS_TCPProtocolEndpointForNAS_BindsToAssocLeaf

**Table 382 - Referenced properties/methods for
EMC_VNXe_IPProtocolEndpointForNAS_TCPProtocolEndpointForNAS_BindsToAssocLeaf**

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_IPProtocolEndpointForNASLeaf
Dependent	Reference of EMC_VNXe_TCPProtocolEndpointForNASLeaf

EMC_VNXe_EthernetPort_CIFSProtocolEndpoint_DeviceSAPImplementationAssocLeaf

**Table 383 - Referenced properties/methods for
EMC_VNXe_EthernetPort_CIFSProtocolEndpoint_DeviceSAPImplementationAssocLeaf**

CIM property	Description/notes
Antecedent	Reference of EMC_EthernetPort
Dependent	Reference of EMC_VNXe_CIFSProtocolEndpointLeaf

EMC_VNXe_EthernetPort_NFSProtocolEndpoint_DeviceSAPImplementationAssocLeaf

**Table 384 - Referenced properties/methods for
EMC_VNXe_EthernetPort_NFSProtocolEndpoint_DeviceSAPImplementationAssocLeaf**

CIM property	Description/notes
Antecedent	Reference of EMC_EthernetPort
Dependent	Reference of EMC_VNXe_NFSProtocolEndpointLeaf

EMC_VNXe_EthernetPort_IPProtocolEndpointForNAS_DeviceSAPImplementationAssocLeaf

**Table 385 - Referenced properties/methods for
EMC_VNXe_EthernetPort_IPProtocolEndpointForNAS_DeviceSAPImplementationAssocLeaf**

CIM property	Description/notes
Antecedent	Reference of EMC_EthernetPort
Dependent	Reference of EMC_VNXe_IPProtocolEndpointLeaf

EMC_VNXe_CIFSServer_CIFSProtocolEndpoint_HostedAccessPointAssocLeaf

**Table 386 - Referenced properties/methods for
EMC_VNXe_CIFSServer_CIFSProtocolEndpoint_HostedAccessPointAssocLeaf**

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_CIFSServerLeaf
Dependent	Reference of EMC_VNXe_CIFSProtocolEndpointLeaf

EMC_VNXe_NFSServer_NFSProtocolEndpoint_HostedAccessPointAssocLeaf

**Table 387 - Referenced properties/methods for
EMC_VNXe_NFSServer_NFSProtocolEndpoint_HostedAccessPointAssocLeaf**

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_NFSServerLeaf
Dependent	Reference of EMC_VNXe_NFSProtocolEndpointLeaf

EMC_VNXe_CIFSServer_TCPProtocolEndpoint_HostedAccessPointAssocLeaf

**Table 388 - Referenced properties/methods for
EMC_VNXe_CIFSServer_TCPProtocolEndpoint_HostedAccessPointAssocLeaf**

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_CIFSServerLeaf
Dependent	Reference of EMC_VNXe_TCPProtocolEndpointLeaf

EMC_VNXe_NFSServer_TCPProtocolEndpoint_HostedAccessPointAssocLeaf

**Table 389 - Referenced properties/methods for
EMC_VNXe_NFSServer_TCPProtocolEndpoint_HostedAccessPointAssocLeaf**

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_NFSServerLeaf
Dependent	Reference of EMC_VNXe_TCPProtocolEndpointLeaf

EMC_VNXe_CIFSServer_IPProtocolEndpoint_HostedAccessPointAssocLeaf

**Table 390 - Referenced properties/methods for
EMC_VNXe_CIFSServer_IPProtocolEndpoint_HostedAccessPointAssocLeaf**

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_CIFSServerLeaf
Dependent	Reference of EMC_VNXe_IPProtocolEndpointLeaf

EMC_VNXe_NFSServer_IPProtocolEndpoint_HostedAccessPointAssocLeaf

**Table 391 - Referenced properties/methods for
EMC_VNXe_NFSServer_IPProtocolEndpoint_HostedAccessPointAssocLeaf**

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_NFSServerLeaf
Dependent	Reference of EMC_VNXe_IPProtocolEndpointLeaf

File Server Manipulation Subprofile

Overview

The File Server Manipulation Subprofile is a subprofile of autonomous profiles that support file systems. It makes use of elements of the file system subprofiles and supports creation, deletion and modification of file servers.

A file server is a computer system that is attached to a network and provides resources to allow client systems access to file system resources in the form of CIFS shares and/or NFS exports. A file server can be either a physical computer system or a virtual system that is hosted by a physical computer system. A physical file server can neither be created nor deleted but may have properties that can be modified via configuration actions. A virtual file server can be created, deleted, and modified via configuration actions. The number of virtual file servers that may be created is system dependent. This profile models both physical and virtual file servers.

NOTE: In a VNXe storage system, supported file servers are all virtual servers.

Extrinsic methods are provided for the creation, deletion and modification of virtual file servers. According to SNIA SMI-S specification, the implementation can provide either a read-only view of the file servers or may provide extrinsic methods for configuring existing and/or new file servers. A client can determine if a read-only implementation is provided by inspecting the two `FileServerConfigurationCapabilities` arrays `SynchronousMethodsSupported` and `AsynchronousMethodsSupported`. If they are both empty or null, then the implementation is read-only.

This profile supports viewing and configuring the following property “areas” of a file server:

- NFS exports
- CIFS shares
- Ethernet port properties including VLAN tagging.
- DNS settings
- NIS settings

NOTE: In VNXe, the File Server Manipulation Profile is implemented from a “configuration” perspective. Through this profile, a client can create a file server, modify a file server, delete a file server, add an IP interface for a file server, modify an IP interface combined to a file server and delete an IP interface from a file server. The indications of creating and deleting file server are also supported.

NOTE: For more details, refer to [Clause 6: File Server Manipulation Profile in Storage Management Technical Specification, Part 4 Filesystems Version 1.6.0, Revision 4](#).

Class diagram

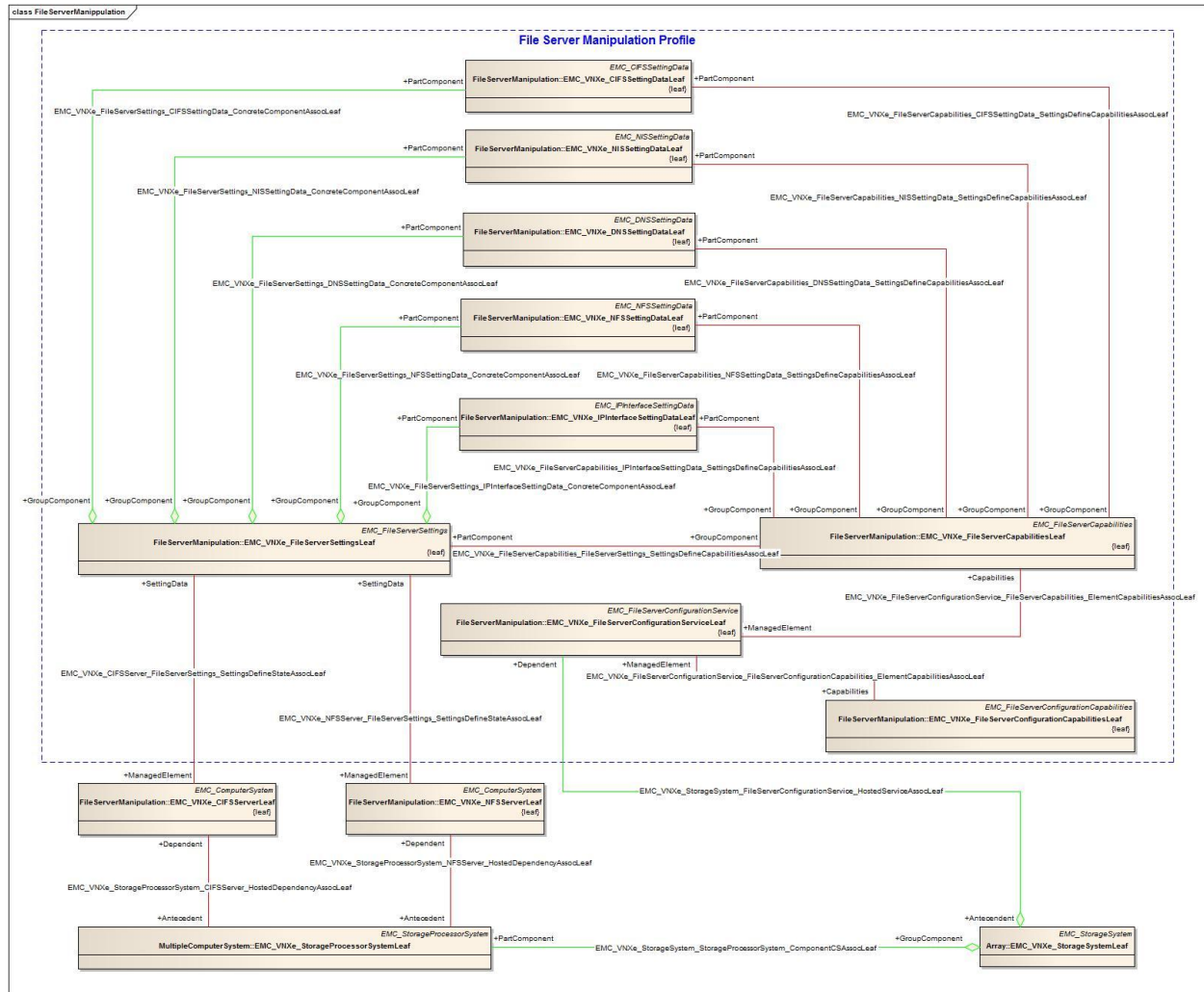


Figure 40 - File Server Manipulation Subprofile class diagram

Methods of the Profile

Extrinsic Methods on Capabilities

Method: *EMC_VNXe_FileServerCapabilitiesLeaf.CreateGoalSettings*

- **Description**

This extrinsic method of the FileServerCapabilities class validates support for caller-proposed Settings passed as the TemplateGoalSettings parameter.

NOTE: This method must meet the following specifications:

- *SNIA_FileServerSettings* (index 0 of *TemplateGoalSettings*) does not support setting properties, that is, any value set for the properties will be ignored.

- If the IPAddress in *SNIA_IPInterfaceSettingData* (index 1) is not set in *TemplateGoalSettings*, a default instance will be returned in index 1 of *SupportedGoalSettings*. This default value of IPAddress may be useless. Client needs to modify it before using it.
- Which type of IPAddress will be returned is decided by the properties of *SNIA_IPInterfaceSettingData* in *TemplateGoalSettings*. If AddressType is IPv6 or valid IPv6PrefixLength (0~128), the IPv6 default instance will be returned; otherwise the IPv4 default instance will be returned.
- The Port property in *SNIA_NFSSettingData* cannot be set. It will always be 2049. Any value set for it will be ignored.
- Table below shows the default setting data instances in this method.

**Table 392 - Default setting data instances used in
EMC_VNXe_FileServerCapabilities.CreateGoalSetting**

Class Name	Property Name	Property Value
SNIA_FileServerSettings	InstanceID	SupportedFileServerSettings
SNIA_IPInterfaceSettingData (IPv4)	InstanceID	SupportedIPInterfaceSettingData
	IPAddress	0.0.0.0
	AddressType	{1: IPv4}
	SubnetMask	255.255.255.255
	VLANId	0
	MTU	1500
SNIA_IPInterfaceSettingData (IPv6)	InstanceID	SupportedIPInterfaceSettingData
	IPAddress	0:0:0:0:0:0:0
	AddressType	{2: Ipv6}
	IPv6PrefixLength	128
	VLANId	0
	MTU	1500
SNIA_CIFSSettingData	InstanceID	SupportedCIFSSettingData
	Enabled	true
	AuthenticationDomain	WORKGROUP
	AuthenticationMode	{0: NTLM}
	CASupported	true
	ProtocolVersions	["SMB3.0"]
	NETBIOSName	NB_DDHHMMSS

Class Name	Property Name	Property Value
	MultiChannelSupported	false
SNIA_NFSSettingData	InstanceID	SupportedNFSSettingData
	Enabled	true
	Port	2049
SNIA_NISSettingData	InstanceID	SupportedNISSettingData
SNIA_DNSSettingData	InstanceID	SupportedDNSSettingData

- Parameters

Table 393 - Signature and parameters of EMC_VNXe_FileServerCapabilitiesLeaf.CreateGoalSettings

Parameter	Qualifiers	Type	Description
TemplateGoalSettings[]	IN, EmbeddedInstance, NULL Allowed	CIM_SettingDataRef[]	<p>Array of 6 elements, each of which is an EmbeddedInstance of a CIM_Setting subclass.</p> <p>Each of the array elements contains either an empty string to represent a "NULL" entry, or an EmbeddedInstance.</p> <p>Each array element contains a specific CIM_Setting subclass as follows:</p> <p>0:EmbeddedInstance("SNIA_FileServerSettings") 1:EmbeddedInstance("SNIA_IPInterfaceSettingData") 2:EmbeddedInstance("SNIA_CIFSSettingData") 3:EmbeddedInstance("SNIA_NFSSettingData") 4:EmbeddedInstance("SNIA_NISSettingData") 5:EmbeddedInstance("SNIA_DNSSettingData")</p>
SupportedGoalSettings[]	IN, Out, EmbeddedInstance	CIM_SettingDataRef[]	<p>On input, each of the array elements contains either an empty string to represent a "NULL" entry, or an EmbeddedInstance. If it contains an EmbeddedInstance, then this instance specifies a previously returned CIM_Setting that the implementation could support.</p> <p>On output, it specifies a new CIM_Setting that the implementation can support.</p> <p>The mappings between array index and EmbeddedInstance in this parameter are the same as the ones in TemplateGoalSettings.</p>

- Return Results

Table 394 - Possible return code of EMC_VNXe_FileServerCapabilitiesLeaf.CreateGoalSettings

Return code	Type	Description
-------------	------	-------------

Return code	Type	Description
0	uint32	Success
4	Uint32	Failed: This error code will be returned when any of the below cases is met: <ul style="list-style-type: none"> Settings specified in <i>TemplateGoalSettings</i> or <i>SupportedGoalSettings</i> conflict.
5	uint32	Invalid Parameter: This error code will be returned when any of the below cases is met: <ul style="list-style-type: none"> Size of <i>TemplateGoalSettings</i> or <i>SupportedGoalSettings</i> is not 6. Any Embedded instances in <i>TemplateGoalSettings</i> or <i>SupportedGoalSettings</i> does not follow the type. Cannot determine IP address type from <i>SNIA_IPInterfaceSettingData</i> in <i>TemplateGoalSettings</i> or <i>SupportedGoalSettings</i>.
6	uint32	Alternative Proposed: This code will be returned when any of the below cases is met: <ul style="list-style-type: none"> Any element in <i>TemplateGoalSettings</i> is a "NULL" entry. Any element in <i>SupportedGoalSettings</i> as input is NOT a "NULL" entry, and it has been modified when output .
NA	CIM_Error	Exceptions will be thrown when any of the below cases is met: <ul style="list-style-type: none"> Any of the required parameters in SNIA Spec is NULL. Any of the input parameters is not of the right CIM/SNIA type defined in the MOF files.

Extrinsic Methods on FileServerConfigurationService

Method: EMC_VNXe_FileServerConfigurationServiceLeaf.CreateFileServer

- Description**

This extrinsic method allows the client to create a new FileServer with several "goal" parameters that represent different configurable aspects of the FileServer.

NOTE: This method must meet the following specifications:

- ElementName is a conditional parameter. User should be aware of following aspects:
 - ElementName should be NULL if AuthenticationMode in CIFSSettingData is set to {0: NTLM} which means to create a Standalone CIFS Server. NETBIOSName in CIFSSettingData will be selected as the server name.
 - ElementName should NOT be empty if AuthenticationMode in CIFSSettingData is set to {1: ActiveDirectory} which means to create a Domain CIFS Server.
 - ElementName should NOT be empty if NFSSettingData is provided which means to create a NFS Server.
- User can NOT create a File Server to support both CIFS and NFS at the same time.

- 3) No property in SNIA_FileServerSettings is supported to set, that is any value set for the properties will be ignored.
- 4) The Port property in SNIA_NFSSettingData is not supported to set. It always is 2049. Any value set for it will be ignored.
- 5) The AuthenticationDomain and AuthenticationMode in SNIA_CIFSSettingData could cause the CreateFileServer fail if AuthenticationMode is "Active Directory". Reason is let CIFSServer join AuthenticationDomain needs the credential which cannot be passed in through SMI-S.
- 6) EthernetPort must be provided if IPInterfaceSettingData is provided.

- **Parameters**

**Table 395 - Signature and Parameters of
EMC_VNXe_FileServerConfigurationServiceLeaf.CreateFileServer**

Parameter	Qualifiers	Type	Description
ElementName	IN, Conditional	string	An end user relevant name for the file server being created.
Job	OUT	CIM_ConcreteJobRef	Reference of the concrete job if this method is asynchronously supported; otherwise it will be NULL.
TheElement	OUT	CIM_LogicalElementRef	Reference to the newly created File Server if this method is synchronously supported; otherwise it will be NULL and client needs to use Job to check the affected element.
FileServerSettings	IN, EmbeddedInstance, Ignored	SNIA_FileServerSettingsRef	The FileServerSettings for the newly created FileServer. NOTE: <ul style="list-style-type: none"> • This parameter will be ignored.
IPInterfaceSettingData	IN, EmbeddedInstance, NULL allowed	SNIA_IPInterfaceSettingDataRef	The IP Interface that the File Server will use for servicing all CIFS and NFS requests.
CIFSSettingData	IN, EmbeddedInstance, NULL allowed	SNIA_CIFSSettingDataRef	The CIFS settings for the File Server being created.
NFSSettingData	IN, EmbeddedInstance, NULL allowed	SNIA_NFSSettingDataRef	The NFS settings for the File Server being created.
DNSSettingData	IN, EmbeddedInstance, NULL allowed	SNIA_DNSSettingDataRef	The DNS settings for the File Server being created.
NISSettingData	IN,	SNIA_NISSettingDataRef	The NIS settings for the File Server being

Parameter	Qualifiers	Type	Description
	EmbeddedInstance, NULL allowed	ataRef	created.
NASComputerSystem	IN, NULL allowed	CIM_ComputerSystemRef	The NAS Head ComputerSystem. NOTE: <ul style="list-style-type: none"> Type must be EMC_VNXe_StorageSystemLeaf.
HostingComputerSystem	IN, Required	CIM_ComputerSystemRef	The ComputerSystem that hosts the File Server. NOTE: <ul style="list-style-type: none"> Type must be EMC_VNXe_StorageProcessorSystemLeaf.
EthernetPort	IN, Conditional	CIM_EthernetPortRef	The hardware port that the File Server will use for mount requests.

- Return Results**

**Table 396 - Possible return code of
EMC_VNXe_FileServerConfigurationServiceLeaf.CreateFileServer**

Return code	Type	Description
0	uint32	Job Completed with No Error
4096	Uint32	Method Parameters Checked – Job Started
1	uint32	Failed. This code will be returned when any of the below cases is met: <ul style="list-style-type: none"> Invalid parameters. Any of optional input 'Goal' settings has invalid properties. Any error occurs when calling underlying component.
NA	CIM_Error	Exceptions will be thrown when any of the below cases is met: <ul style="list-style-type: none"> Any of the required parameters in SNIA Spec is NULL. Any of input parameters is not of the right CIM/SNIA type defined in the MOF files. Any of input references cannot be found in system.

Method: EMC_VNXe_FileServerConfigurationServiceLeaf.ModifyFileServer

- Description**

This extrinsic method allows the client to modify the settings for an existing file server. All settings except `IPInterfaceSettingData`, `HostingComputerSystem`, and `EthernetPort` may be modified. To modify the `IPInterfaceSettingData`, `HostingComputerSystem`, and/or `EthernetPort` properties, use the extrinsic method `ModifyIPInterface`.

NOTE: This method must meet the following specifications:

- 1) `ElementName` can NOT be modified if `AuthenticationMode` in `CIFSSettingData` of the File Server is set to {0: NTLM} which means the File Server is a Standalone CIFS Server. User should use `NETBIOSName` in `CIFSSettingData` to modify name of the server.
- 2) Disable CIFS or NFS support via `CIFSSettingData.Enabled` or `NFSSettingData.Enabled` is NOT supported.
- 3) No property in `SNIA_FileServerSettings` is supported to set, that is any value set for the properties will be ignored.
- 4) The Port property in `SNIA_NFSSettingData` is not supported to set. It always is 2049. Any value set for it will be ignored.
- 5) The `AuthenticationDomain` and `AuthenticationMode` in `SNIA_CIFSSettingData` could cause the `ModifyFileServer` fail if `AuthenticationMode` is "Active Directory". Reason is that let CIFS server join `AuthenticationDomain` needs the credential which cannot be passed in through SMI-S.

- **Parameters**

**Table 397 - Signature and Parameters of
EMC_VNXe_FileServerConfigurationServiceLeaf.ModifyFileServer**

Parameter	Qualifiers	Type	Description
FileServer	IN, Required	CIM_ComputerSystemRef	Reference to the File Server to be modified. NOTE: <ul style="list-style-type: none"> • Type must be EMC_VNXe_CIFSServerLeaf or EMC_VNXe_NFSServerLeaf.
Job	OUT	CIM_ConcreteJobRef	Reference of the concrete job if this method is asynchronously supported; otherwise it will be NULL.
ElementName	IN, NULL Allowed	String	The new name of the File Server being modified.
FileServerSettings	IN, EmbeddedInstance, Ignored	SNIA_FileServerSettingsRef	The FileServerSettings for the FileServer to be modified. NOTE: <ul style="list-style-type: none"> • This parameter will be ignored.
CIFSSettingData	IN, EmbeddedInstance, NULL allowed	SNIA_CIFSSettingDataRef	The CIFS settings for the File Server being modified.

Parameter	Qualifiers	Type	Description
NFSSettingData	IN, EmbeddedInstance, NULL allowed	SNIA_NFSSettingDataRef	The NFS settings for the File Server being modified.
DNSSettingData	IN, EmbeddedInstance, NULL allowed	SNIA_DNSSettingDataRef	The DNS settings for the File Server being modified.
NISSettingData	IN, EmbeddedInstance, NULL allowed	SNIA_NISSettingDataRef	The NIS settings for the File Server being modified.

- **Return Results**

**Table 398 - Possible return code of
EMC_VNXe_FileServerConfigurationServiceLeaf.ModifyFileServer**

Return code	Type	Description
0	uint32	Job Completed with No Error
4096	Uint32	Method Parameters Checked – Job Started
1	uint32	Failed. This code will be returned when any of the below cases is met: <ul style="list-style-type: none"> • Invalid parameters. • Any of optional input 'Goal' settings has invalid properties. • Any error occurs when calling underlying component.
NA	CIM_Error	Exceptions will be thrown when any of the below cases is met: <ul style="list-style-type: none"> • Any of the required parameters in SNIA Spec is NULL. • Any of input parameters is not of the right CIM/SNIA type defined in the MOF files. • Any of input references cannot be found in system.

Method: EMC_VNXe_FileServerConfigurationServiceLeaf.DeleteFileServer

- **Description**

This extrinsic method allows client to delete a file server.

- **Parameters**

**Table 399 - Signature and Parameters of
EMC_VNXe_FileServerConfigurationServiceLeaf.DeleteFileServer**

Parameter	Qualifiers	Type	Description
FileServer	IN, Required	CIM_ComputerSystemRef	Reference to the File Server to be deleted. NOTE: <ul style="list-style-type: none"> Type must be EMC_VNXe_CIFSServerLeaf or EMC_VNXe_NFSServerLeaf.
Job	OUT	CIM_ConcreteJobRef	Reference of the concrete job if this method is asynchronously supported; otherwise it will be NULL.

- **Return Results**

**Table 400 - Possible return code of
EMC_VNXe_FileServerConfigurationServiceLeaf.DeleteFileServer**

Return code	Type	Description
0	uint32	Job Completed with No Error
4096	UInt32	Method Parameters Checked – Job Started
1	uint32	Failed. This code will be returned when any of the below cases is met: <ul style="list-style-type: none"> Invalid type of FileServer. Any error occurs when calling underlying component.
NA	CIM_Error	Exceptions will be thrown when any of the below cases is met: <ul style="list-style-type: none"> Any of the required parameters in SNIA Spec is NULL. Any of input parameters is not of the right CIM/SNIA type defined in the MOF files. Any of input references cannot be found in system.

Method: EMC_VNXe_FileServerConfigurationServiceLeaf.AddIPInterface

- **Description**

This extrinsic method allows client to add a new IPInterface to an existing file server.

- **Parameters**

**Table 401 - Signature and Parameters of
EMC_VNXe_FileServerConfigurationServiceLeaf.AddIPInterface**

Parameter	Qualifiers	Type	Description
FileServer	IN,	CIM_ComputerSystemRef	Reference to the File Server to which the IP address to be added.

Parameter	Qualifiers	Type	Description
	Required		NOTE: <ul style="list-style-type: none"> Type must be EMC_VNXe_CIFSServerLeaf or EMC_VNXe_NFSServerLeaf.
Job	OUT	CIM_ConcreteJobRef	Reference of the concrete job if this method is asynchronously supported; otherwise it will be NULL.
IPInterfaceSettingData	IN, EmbeddedInstance, Require	SNIA_IPInterfaceSettingDataRef	IP Interface that the File Server will use for servicing all CIFS and NFS requests.
HostingComputerSystem	IN, Required	CIM_ComputerSystemRef	ComputerSystem that hosts the File Server. NOTE: <ul style="list-style-type: none"> Type must be EMC_VNXe_StorageProcessorSystemLeaf.
EthernetPort	IN, Required	CIM_EthernetPortRef	EthernetPort identifies the hardware port that the File Server will use for mount requests on the new IPAddress.

- Return Results**

Table 402 - Possible return code of EMC_VNXe_FileServerConfigurationServiceLeaf.AddIPInterface

Return code	Type	Description
0	uint32	Job Completed with No Error
4096	Uint32	Method Parameters Checked – Job Started
1	uint32	Failed. This code will be returned when any of the below cases is met: <ul style="list-style-type: none"> Invalid parameters. Input IPInterfaceSettingData is not supported in system. Invalid properties of input IPInterfaceSettingData. Input FileServer is not hosted by input HostingComputerSystem. Input EthernetPort cannot be used by FileServer. Any error occurs when calling underlying component.
NA	CIM_Error	Exceptions will be thrown when any of the below cases is met: <ul style="list-style-type: none"> Any of the required parameters in SNIA Spec is NULL. Any of the input parameters is not of the right CIM/SNIA type defined in

		<p>the MOF files.</p> <ul style="list-style-type: none"> Any of input references cannot be found in system.
--	--	--

Method: EMC_VNXe_FileServerConfigurationServiceLeaf.ModifyIPInterface

- Description**

This extrinsic method allows the client to modify the existing IPInterface of the file server.

NOTE: This method must meet the following specification:

If HostingComputerSystem is non-NULL, it must be the one hosting the FileServer. SMI-S does NOT support using this method to modify HostingComputerSystem.

- Parameters**

Table 403 - Signature and Parameters of EMC_VNXe_FileServerConfigurationServiceLeaf.ModifyIPInterface

Parameter	Qualifiers	Type	Description
FileServer	IN, Required	CIM_ComputerSystemRef	Reference to the FileServer from which the IPInterface will be modified. NOTE: <ul style="list-style-type: none"> Type must be EMC_VNXe_CIFSServerLeaf or EMC_VNXe_NFSServerLeaf.
IPInterfaceSettingData	IN, EmbeddedInstance, Required	SNIA_IPInterfaceSettingDataRef	IPInterfaceSettingData that is to be modified. This is used to identify which IPInterfaceSettingData instance to modify.
Job	OUT	CIM_ConcreteJobRef	Reference of the concrete job if this method is asynchronously supported; otherwise it will be NULL.
NewIPInterfaceSettingData	IN, EmbeddedInstance, NULL allowed	SNIA_IPInterfaceSettingDataRef	If non-NULL, the IPInterfaceSettingData that will replace an existing IPInterfaceSettingData instance in the FileServer. If NULL, then the IPInterfaceSettingData will not be modified.
HostingComputerSystem	IN, NULL allowed	CIM_ComputerSystemRef	ComputerSystem that hosts the File Server. NOTE: <ul style="list-style-type: none"> Type must be EMC_VNXe_StorageProcessorSystemLeaf. If non-NULL, it must be the one hosting the FileServer.

Parameter	Qualifiers	Type	Description
EthernetPort	IN, NULL allowed	CIM_EthernetPortRef	If non-NULL, the EthernetPort identifies the new hardware port for the IPInterface. If NULL, the current EthernetPort setting will not be changed.

- **Return Results**

**Table 404 - Possible return code of
EMC_VNXe_FileServerConfigurationServiceLeaf.ModifyIPInterface**

Return code	Type	Description
0	uint32	Job Completed with No Error
4096	UInt32	Method Parameters Checked – Job Started
1	uint32	Failed. This code will be returned when any of the below cases is met: <ul style="list-style-type: none"> • Invalid parameters. • Input IPInterfaceSettingData doesn't identify an existing IP interface hosted by the FileServer. • Input NewIPInterfaceSettingData is not supported in the system. • Invalid properties of input NewIPInterfaceSettingData. • Input FileServer is not hosted by input HostingComputerSystem. • Any error occurs when calling underlying component.
NA	CIM_Error	Exceptions will be thrown when any of the below cases is met: <ul style="list-style-type: none"> • Any of the required parameters in SNIA Spec is NULL. • Any of the input parameters is not of the right CIM/SNIA type defined in the MOF files. • Any of the input references cannot be found in system.

Method: EMC_VNXe_FileServerConfigurationServiceLeaf.DeleteIPInterface

- **Description**

This extrinsic method allows the client to delete an existing IPInterface from the file server.

- **Parameters**

**Table 405 - Signature and Parameters of
EMC_VNXe_FileServerConfigurationServiceLeaf.DeleteIPInterface**

Parameter	Qualifiers	Type	Description
FileServer	IN,	CIM_ComputerSystemRef	Reference to the FileServer from

Parameter	Qualifiers	Type	Description
	Required		<p>which the IPInterface will be removed.</p> <p>NOTE:</p> <ul style="list-style-type: none"> Type must be EMC_VNXe_CIFSServerLeaf or EMC_VNXe_NFSServerLeaf.
IPInterfaceSettingData	IN, EmbeddedInstance, Required	SNIA_IPInterfaceSettingDataRef	<p>IPInterfaceSettingData that is to be removed.</p> <p>This is used to identify which IPInterfaceSettingData instance to modify.</p>
Job	OUT	CIM_ConcreteJobRef	Reference of the concrete job if this method is asynchronously supported; otherwise it will be NULL.

- Return Results**

**Table 406 - Possible return code of
EMC_VNXe_FileServerConfigurationServiceLeaf.DeleteIPInterface**

Return code	Type	Description
0	uint32	Job Completed with No Error
4096	UInt32	Method Parameters Checked – Job Started
1	uint32	<p>Failed.</p> <p>This code will be returned when any of the below cases is met:</p> <ul style="list-style-type: none"> Invalid parameters. Input IPInterfaceSettingData does not identify an existing IP interface hosted by the FileServer. Any error occurs when calling underlying component.
NA	CIM_Error	<p>Exceptions will be thrown when any of the below cases is met:</p> <ul style="list-style-type: none"> Any of the required parameters in SNIA Spec is NULL. Any of the input parameters is not of the right CIM/SNIA type defined in the MOF files. Any of the input references cannot be found in system.

Client considerations

Model Specification

- Goal Setting Data

NOTE: Before performing File Server management, user should negotiate with the provider to obtain supported setting data via `CreateGoalSettings()` and then pass them into corresponding extrinsic methods. There is no guarantee for any fabricated setting data.

- File Server Creation Prerequisite

Per the VNXe system design, metadata for FileServer is stored in an user-defined pool. There must be at least one pool before creating any FileServer.

In this implementation, the current largest user-defined pool will be selected to store the metadata of the FileServer be in the system.

Use case: Create Supported Settings for File Server

Series settings can be specified and passed by the client when creating a file server via SMI-S. This is a negotiation between client and provider to get system supported settings.

Refer to [Use Case: Create a File Server](#)

Use case: Create a File Server

This use case describes how to create a file server with user specified settings.

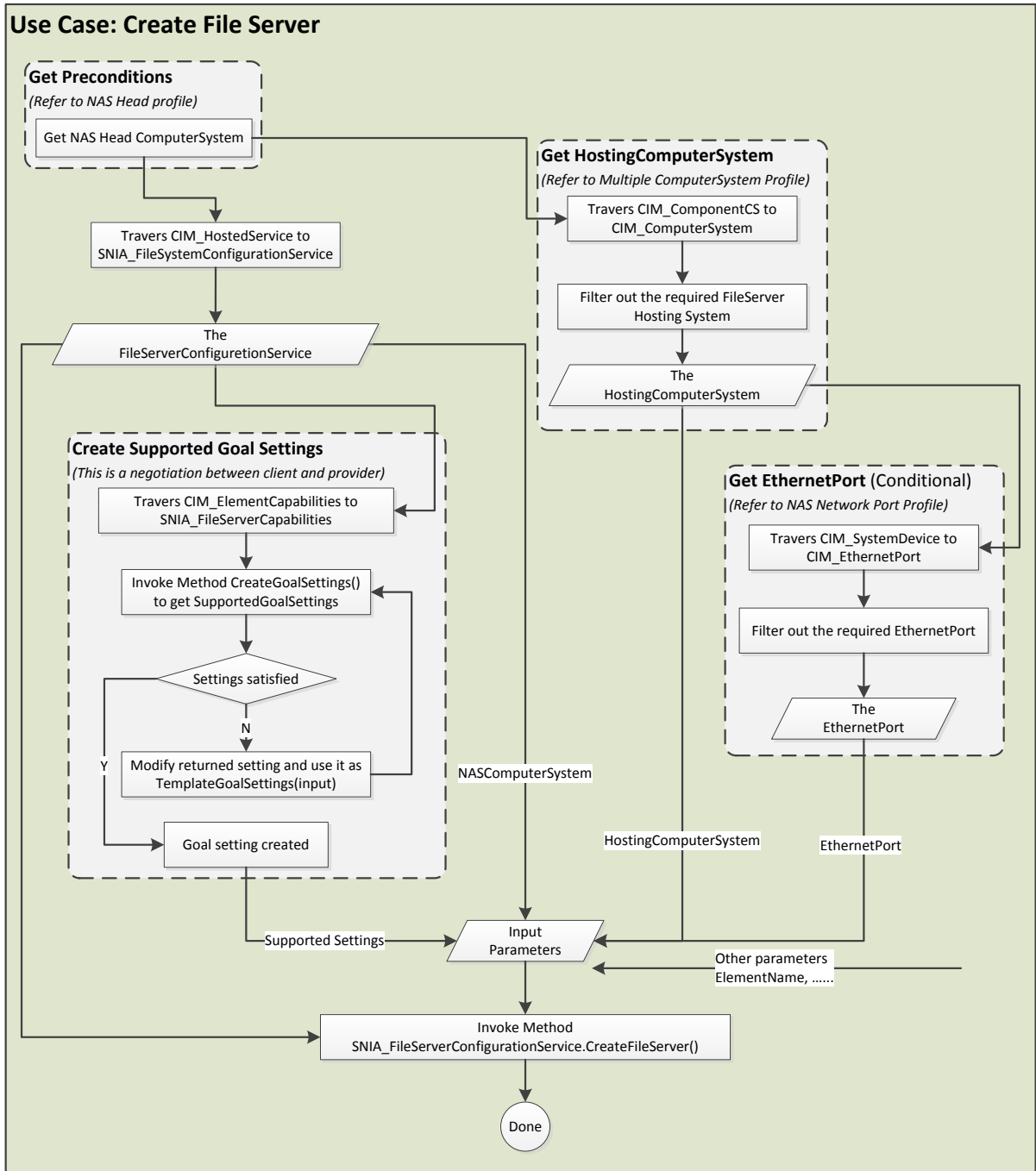


Figure 41 - Flowchart of Creating File Server

Use case: Modify a File Server

This use case describes how to modify aspects of a file server.

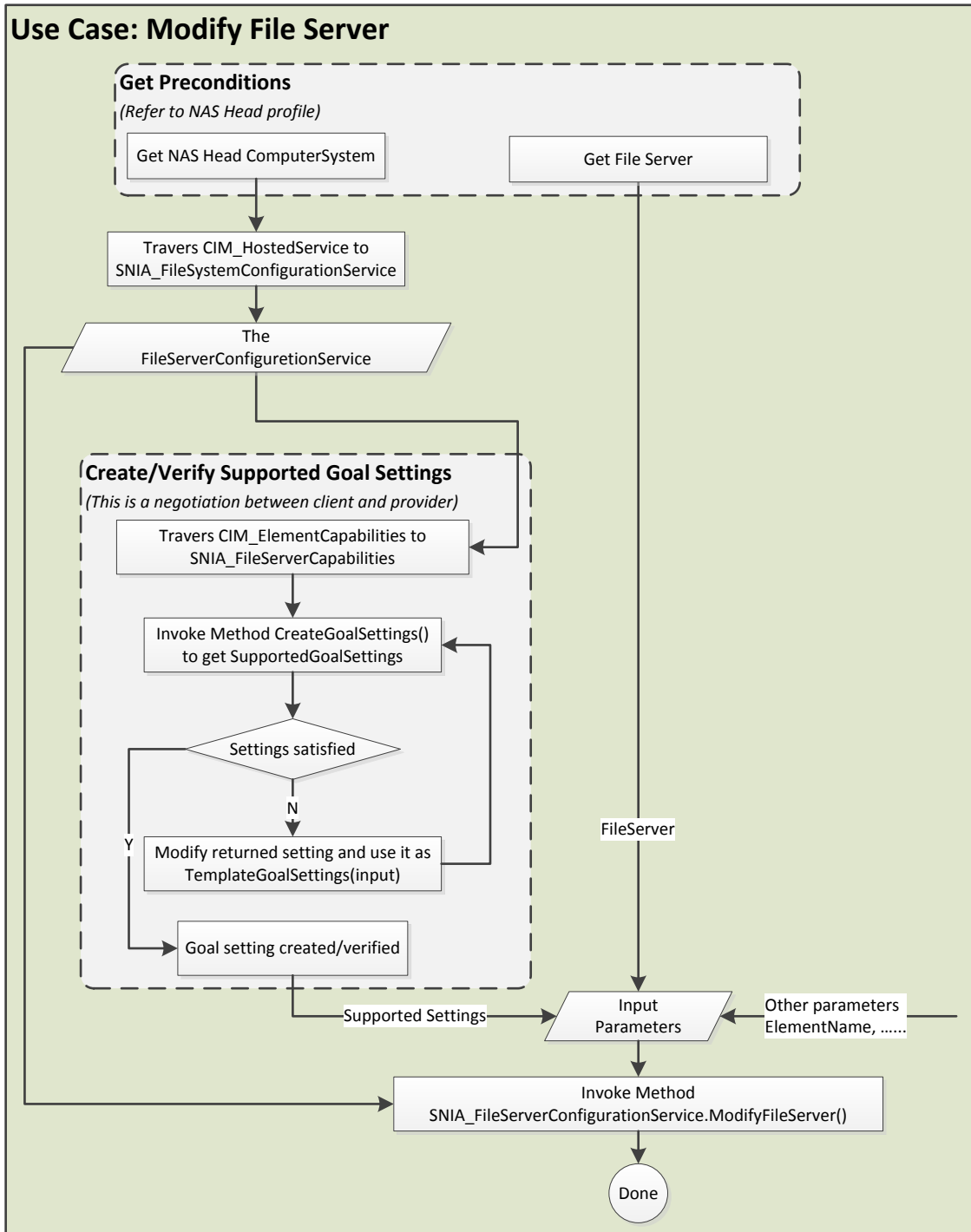


Figure 42 - Flowchart of Modifying File Server

Use case: Delete a File Server

Users can invoke `SNIA_FileServerConfigurationService.DeleteFileServer` to delete a file server.

Use case: Add a New IP Interface to an Existing CIFS File Server

This use case describes how to add a new IP interface to an existing file server.

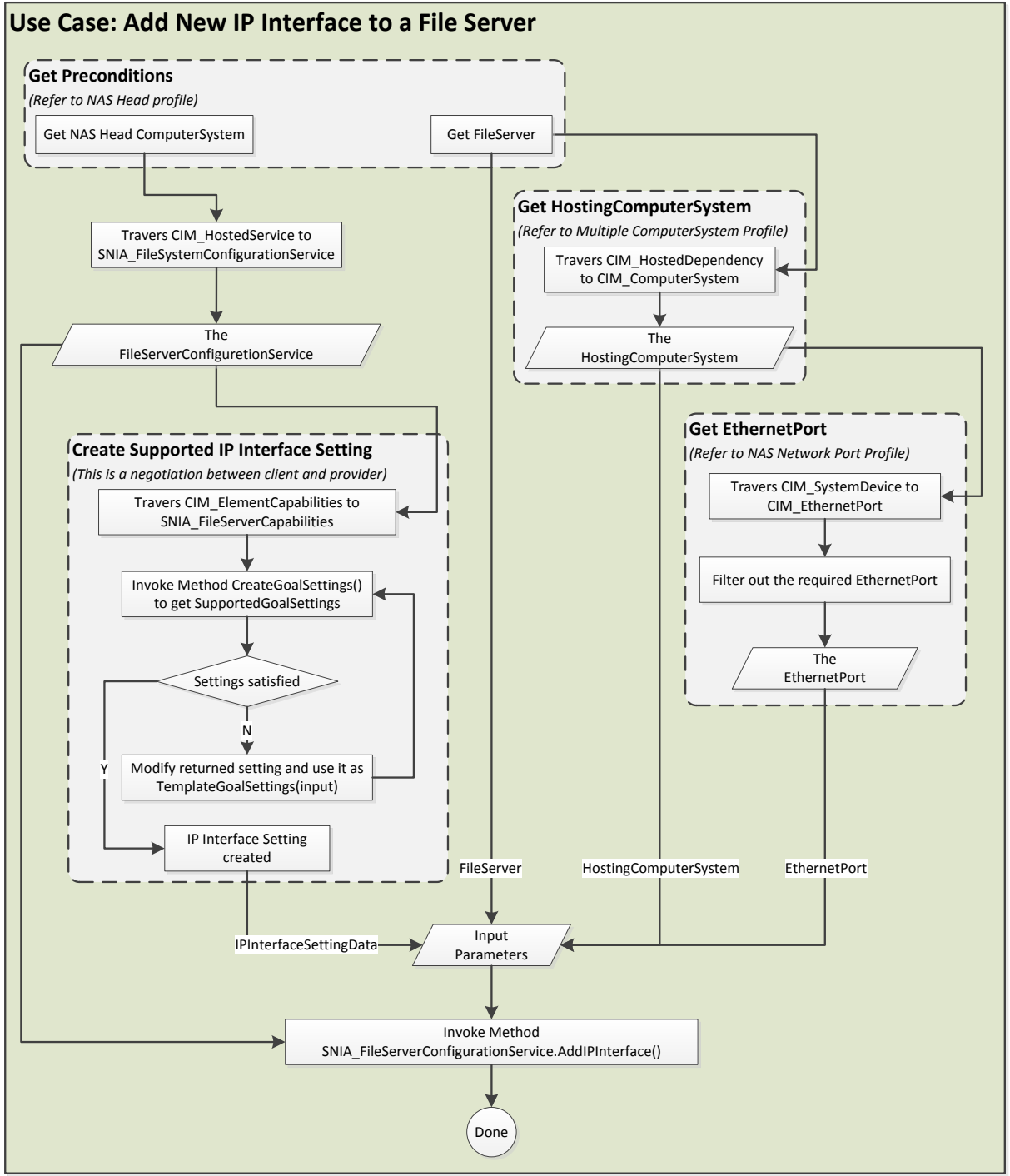


Figure 43 - Flowchart of Adding New IP Interface to a File Server

Use case: Modify an Existing IP Interface of a File Server

This use case describes how to modify an existing IP interface of a file server.

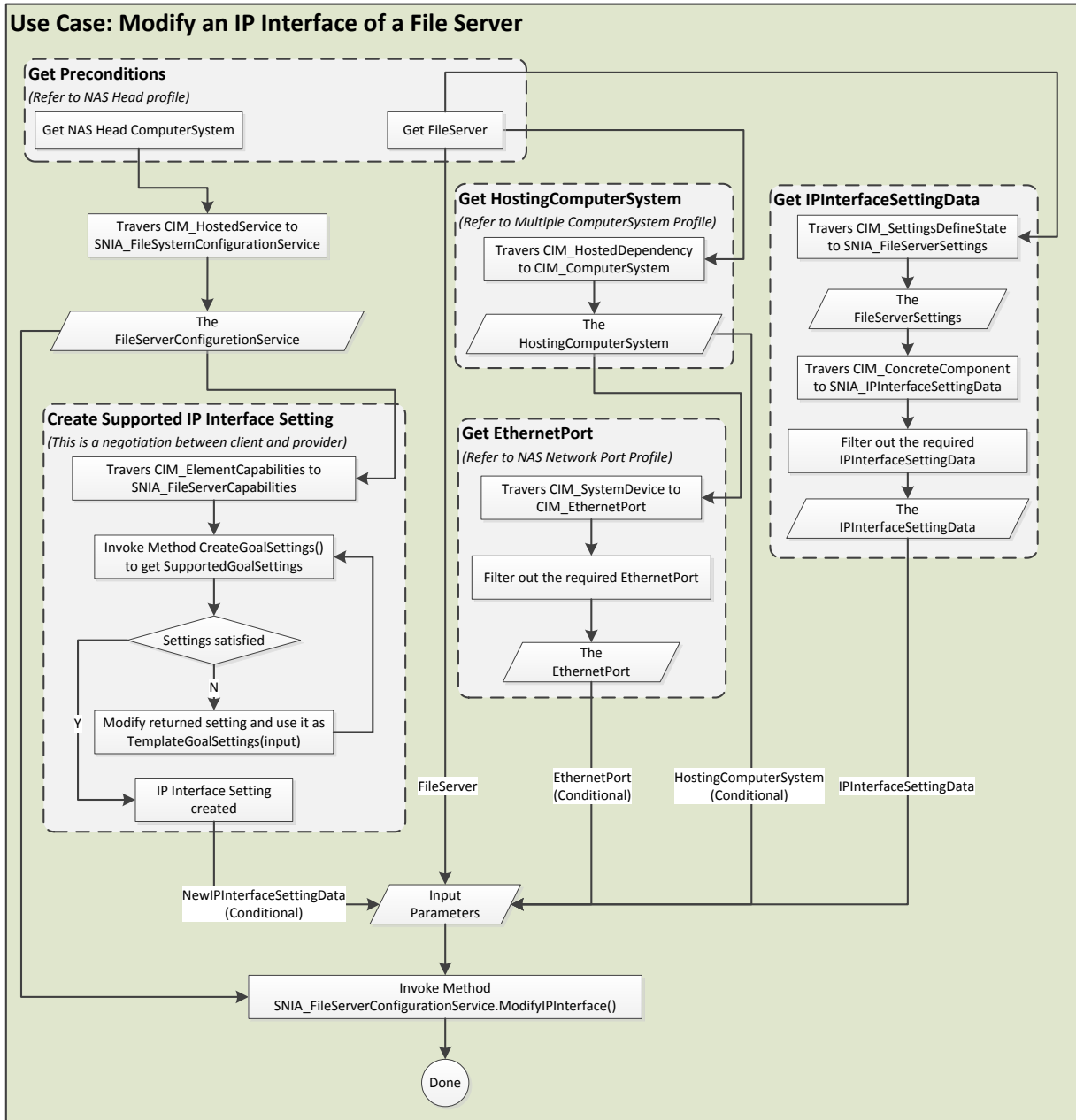


Figure 44 - Flowchart of Modifying an Existing IP Interface of the File Server

Use case: Delete an Existing IP Interface from a CIFS File Server

This use case describes how to delete an existing IP interface from a file server.

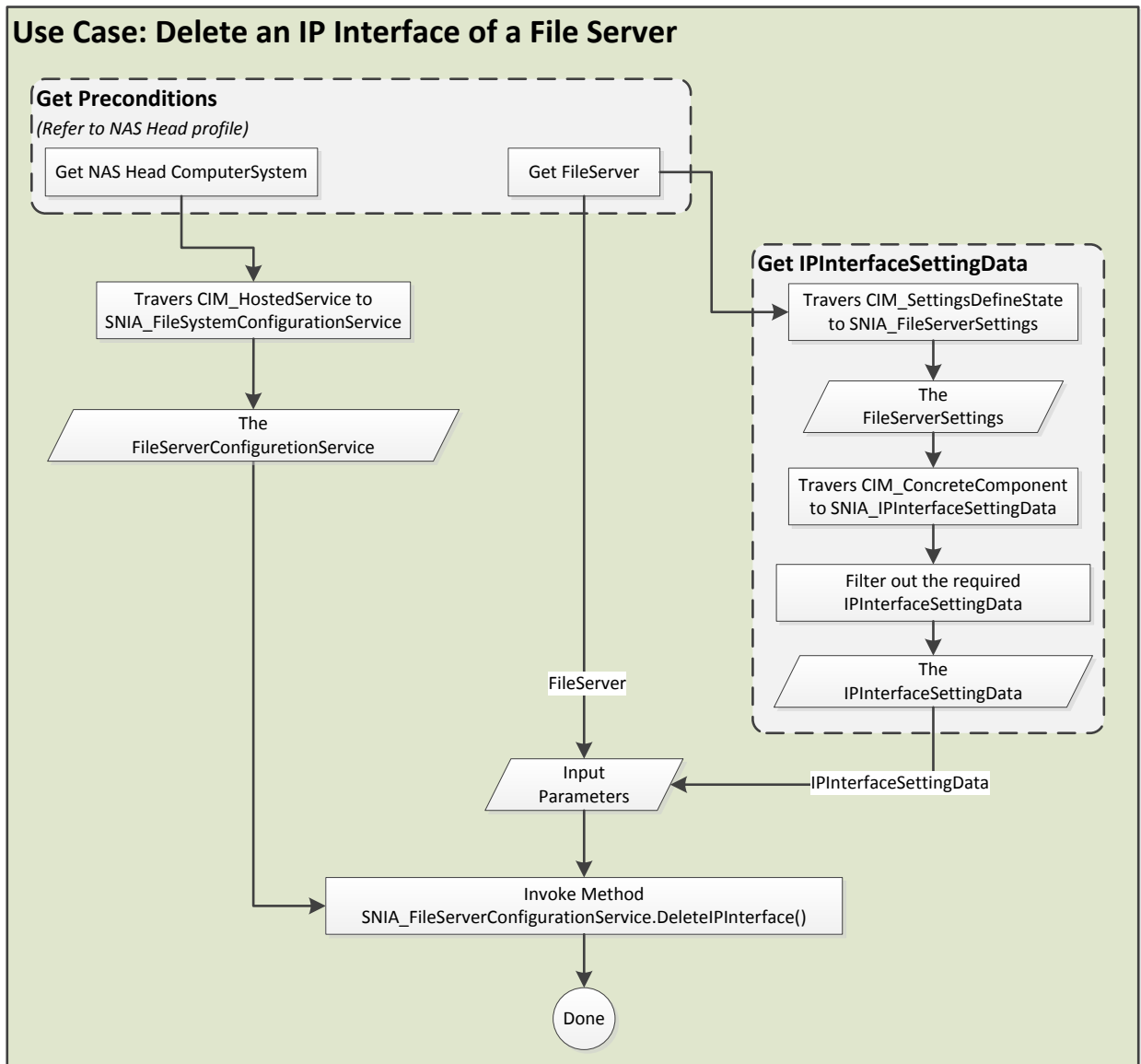


Figure 45 - Flowchart of Deleting an Existing IP Interface from a File Server

CIM Elements

The implemented classes and associations related to File Server Manipulation Subprofile in the VNxe storage system are described as follows:

Table 407 - CIM Elements for File Server Manipulation Subprofile

CIM Class	Implemented Class	Description
CIM_ComputerSystem	EMC_VNxe_CIFSServerLeaf	Represents the CIFS file server.
CIM_ComputerSystem	EMC_VNxe_NFSServerLeaf	Represents the NFS file server.
CIM_ConcreteComponent	EMC_VNxe_FileServerSettings_CIFSSettingData_ConcreteComp	Represents the association between the File Server Settings and its concrete CIFS

CIM Class	Implemented Class	Description
	onentAssocLeaf	settings.
CIM_ConcreteComponent	EMC_VNXe_FileServerSettings_DNSSettingData_ConcreteComponentAssocLeaf	Represents the association between the File Server Settings and its concrete DNS setting data.
CIM_ConcreteComponent	EMC_VNXe_FileServerSettings_IPInterfaceSettingData_ConcreteComponentAssocLeaf	Represents the association between the File Server Settings and its concrete IP interface settings.
CIM_ConcreteComponent	EMC_VNXe_FileServerSettings_NFSSettingData_ConcreteComponentAssocLeaf	Represents the association between the File Server Settings and its concrete NFS settings.
CIM_ConcreteComponent	EMC_VNXe_FileServerSettings_NISSettingData_ConcreteComponentAssocLeaf	Represents the association between the File Server Settings and its concrete NIS server settings.
CIM_ElementCapabilities	EMC_VNXe_FileServerConfigurationService_FileServerCapabilities_ElementCapabilitiesAssocLeaf	Represents the association between the File Server Configuration Service and the Capabilities supported by all file servers.
CIM_ElementCapabilities	EMC_VNXe_FileServerConfigurationService_FileServerConfigurationCapabilities_ElementCapabilitiesAssocLeaf	Represents the association between the File Server Configuration Service and its supported capabilities.
CIM_HostedService	EMC_VNXe_StorageSystem_FileServerConfigurationService_HostedServiceAssocLeaf	Represents the association between the top level computer system and the file server configuration service hosed on it.
CIM_SettingsDefineState	EMC_VNXe_CIFSServer_FileServerSettings_SettingsDefineStateAssocLeaf	Represents the association between the CIFS file server computer system and its settings.
CIM_SettingsDefineState	EMC_VNXe_NFSServer_FileServerSettings_SettingsDefineStateAssocLeaf	Represents the association between the NFS file server computer system and its settings.
SNIA_CIFSSettingData	EMC_VNXe_CIFSSettingDataLeaf	Represents the CIFS settings of file server.
SNIA_DNSSettingData	EMC_VNXe_DNSSettingDataLeaf	Represents the DNS setting data used by file server.
SNIA_FileServerCapabilites	EMC_VNXe_FileServerCapabilitiesLeaf	Represents the capabilities of file server.
SNIA_FileServerConfigurationCapabilities	EMC_VNXe_FileServerConfigurationCapabilitiesLeaf	Represents the management capabilities of the File Server Configuration Service.
SNIA_FileServerConfigurationService	EMC_VNXe_FileServerConfigurationServiceLeaf	Represents the service providing the methods to manipulate file servers.
SNIA_FileServerSettings	EMC_VNXe_FileServerSettingsL	Represents the settings for the file server.

CIM Class	Implemented Class	Description
	eaf	
SNIA_IPInterfaceSettingData	EMC_VNXe_IPInterfaceSettingDataLeaf	Represents the settings for single IP interface.
SNIA_NFSSettingData	EMC_VNXe_NFSSettingDataLeaf	Represents the NFS settings of file server.
SNIA_NISSettingData	EMC_VNXe_NISSettingDataLeaf	Represents the settings of Network Information System configured for the file server.
SNIA_SettingsDefineCapabilities	EMC_VNXe_FileServerCapabilities_CIFSSettingData_SettingsDefineCapabilitiesAssocLeaf	Represents the association between the File Server Capabilities and its default CIFS settings.
SNIA_SettingsDefineCapabilities	EMC_VNXe_FileServerCapabilities_DNSSettingData_SettingsDefineCapabilitiesAssocLeaf	Represents the association between the File Server Capabilities and its default DNS setting data.
SNIA_SettingsDefineCapabilities	EMC_VNXe_FileServerCapabilities_FileServerSettings_SettingsDefineCapabilitiesAssocLeaf	Represents the association between the File Server Capabilities and its default File Server Settings.
SNIA_SettingsDefineCapabilities	EMC_VNXe_FileServerCapabilities_IPInterfaceSettingData_SettingsDefineCapabilitiesAssocLeaf	Represents the association between the File Server Capabilities and its default IP interface settings.
SNIA_SettingsDefineCapabilities	EMC_VNXe_FileServerCapabilities_NFSSettingData_SettingsDefineCapabilitiesAssocLeaf	Represents the association between the File Server Capabilities and its default NFS settings.
SNIA_SettingsDefineCapabilities	EMC_VNXe_FileServerCapabilities_NISSettingData_SettingsDefineCapabilitiesAssocLeaf	Represents the association between the File Server Capabilities and its default NIS server settings.

EMC_VNXe_FileServerConfigurationServiceLeaf

Table 408 - Referenced properties/methods for EMC_VNXe_FileServerConfigurationServiceLeaf

CIM property	Description/notes
SystemCreationClassName	Set to 'EMC_VNXe_StorageSystemLeaf'.
SystemName	Name of the storage system.
CreationClassName	Set to 'EMC_VNXe_FileServerConfigurationServiceLeaf'
Name	Same as SystemName
ElementName	Set as 'File Server Configuration Service'

EMC_VNXe_FileServerConfigurationCapabilitiesLeaf

Table 409 - Referenced properties/methods for EMC_VNXe_FileServerConfigurationCapabilitiesLeaf

CIM property	Description/notes
InstanceID	Unique ID for the instance.
ElementName	Set as 'File Server Configuration Capabilities'
SynchronousMethodsSupported	Set to [2: CreateFileServer, 3: DeleteFileServer, 4: ModifyFileServer, 5: AddIPInterface, 6: ModifyIPInterface, 7: DeleteIPInterface, 8: CreateGoalSettings]
AsynchronousMethodsSupported	Set to empty array.
CanConfigureCIFS	Indicates if CIFS settings can be configured. Set to TRUE.
CanConfigureNFS	Indicates if NFS settings can be configured. Set to TRUE.
CanConfigureNIS	Indicates if NIS settings can be configured. Set to TRUE.
CanConfigureDNS	Indicates if DNS settings can be configured. Set to TRUE.
CanConfigureNetworkVLAN	Indicates if Network VLAN tagging settings can be configured. Set to TRUE.

EMC_VNXe_FileServerCapabilitiesLeaf

Table 410 - Referenced properties/methods for EMC_VNXe_FileServerCapabilitiesLeaf

CIM property	Description/notes
InstanceID	Unique ID for the instance.
ElementName	User friendly name of the instance.
FileServerSettingsSupported	Indicates if there will be a FileServerSettings instance associated with the FileServer ComputerSystem. Set to TRUE.
CIFSSupported	Indicates if the File Server ComputerSystem supports CIFS. Set to TRUE.
NFSSupported	Indicates if the File Server ComputerSystem supports NFS. Set to TRUE.

CIM property	Description/notes
NISSupported	Indicates if the File Server ComputerSystem supports NIS. Set to TRUE.
DNSSupported	Indicates if the File Server ComputerSystem supports DNS. Set to TRUE.
NetworkVLANSupported	Indicates if the File Server ComputerSystem supports Scale Out. Set to TRUE.
ScaleOutSupported	Indicates if the File Server ComputerSystem supports Network VLAN tagging. Set to FALSE.

EMC_VNXe_CIFSServerLeaf

Table 411 - Referenced properties/methods for EMC_VNXe_CIFSServerLeaf

CIM property	Description/notes
CreationClassName	Set to 'EMC_VNXe_CIFSServerLeaf'
Name	FriendlyID of the File Server.
ElementName	Name or NETBIOSName of the File Server.
OperationalStatus	Current operational status of the File Server.
NameFormat	Set to 'Other'
Dedicated	Set to [16: FileServer]
PrimaryStatus	High level status of the File Server.
HealthState	Current health state of the File Server.

EMC_VNXe_NFSServerLeaf

Table 412 - Referenced properties/methods for EMC_VNXe_NFSServerLeaf

CIM property	Description/notes
CreationClassName	Set to 'EMC_VNXe_NFSServerLeaf'
Name	FriendlyID of the File Server.
ElementName	Name or NETBIOSName of the File Server.
OperationalStatus	Current operational status of the File Server.
NameFormat	Set to 'Other'

CIM property	Description/notes
Dedicated	Set to [16: FileServer]
PrimaryStatus	Hige level status of the File Server.
HealthState	Current health state of the File Server.

EMC_VNXe_FileServerSettingsLeaf

Table 413 - Referenced properties/methods for EMC_VNXe_FileServerSettingsLeaf

CIM property	Description/notes
InstanceID	Unique ID of the instance.

EMC_VNXe_IPInterfaceSettingDataLeaf

Table 414 - Referenced properties/methods for EMC_VNXe_IPInterfaceSettingDataLeaf

CIM property	Description/notes
InstanceID	Unique ID of the instance.
IPAddress	The IP address that this IP Interface represents.
AddressType	Format of the IPAddress. Set to {1: IPv4} or {2: IPv6}
SubnetMask	The mask for the IPv4 address if AddressType is set to {1; IPv4}.
IPv6PrefixLength	The prefix length for the IPv6 address if AddressType is set to {2; IPv6}.
VLANId	A 12-bit VLAN ID used in the VLAN Tag header.
MTU	The Maximum Transmission Unit that is to be used
RDMAcapable	Indicates if RDMA is capable.
RSSCapable	Indicates if RSS is capable.
LinkSpeed	Speed of the interface.

EMC_VNXe_DNSSettingDataLeaf

Table 415 - Referenced properties/methods for EMC_VNXe_DNSSettingDataLeaf

CIM property	Description/notes
InstanceID	Unique ID of the instance.
DomainName	Name of the domain.

CIM property	Description/notes
DNSServerAddresses	Addresses of the DNS Server.

EMC_VNXe_NISSettingDataLeaf

Table 416 - Referenced properties/methods for EMC_VNXe_NISSettingDataLeaf

CIM property	Description/notes
InstanceID	Unique ID of the instance.
DomainName	The name of the NIS domain to use for looking up hosts and users.
ServerIP	The IP address of a NIS server.

EMC_VNXe_CIFSSettingDataLeaf

Table 417 - Referenced properties/methods for EMC_VNXe_CIFSSettingDataLeaf

CIM property	Description/notes
InstanceID	Unique ID of the instance.
Enabled	Indicates if CIFS is enabled for the FileServer.
NETBIOSName	The NetBIOS name of the FileServer
AuthenticationDomain	Name of CIFS domain to which the File Server is joined. Represents either the NTLM domain or the ActiveDirectory domain.
AuthenticationMode	Specifies if authentication is to be performed against NTLM or ActiveDirectory domains.
CASupported	Indicates if Continuous Availability (CA) if supported.
ProtocolVersions	Supported SMB versions.
MultiChannelSupported	Indicates if MPIO is supported.

EMC_VNXe_NFSSettingDataLeaf

Table 418 - Referenced properties/methods for EMC_VNXe_NFSSettingDataLeaf

CIM property	Description/notes
InstanceID	Unique ID of the instance.
Enabled	Indicates if NFS is enabled for the FileServer.
Port	Access port of the protocol.

EMC_VNXe_StorageSystem_FileServerConfigurationService_HostedServiceAssocLeaf

**Table 419 - Referenced properties/methods for
EMC_VNXe_StorageSystem_FileServerConfigurationService_HostedServiceAssocLeaf**

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_StorageSystemLeaf
Dependent	Reference of EMC_VNXe_FileServerConfigurationServiceLeaf

EMC_VNXe_FileServerConfigurationService_FileServerCapabilities_ElementCapabilitiesAssocLeaf

**Table 420 - Referenced properties/methods for
EMC_VNXe_FileServerConfigurationService_FileServerCapabilities_ElementCapabilitiesAssocLeaf**

CIM property	Description/notes
ManagedElement	Reference of EMC_VNXe_FileServerConfigurationServiceLeaf
Capabilities	Reference of EMC_VNXe_FileServerConfigurationCapabilitiesLeaf

EMC_VNXe_FileServerConfigurationService_FileServerConfigurationCapabilities_ElementCapabilitiesAssocLeaf

**Table 421 - Referenced properties/methods for
EMC_VNXe_FileServerConfigurationService_FileServerConfigurationCapabilities_ElementCapabilitiesAssocLeaf**

CIM property	Description/notes
ManagedElement	Reference of EMC_VNXe_FileServerConfigurationServiceLeaf
Capabilities	Reference of EMC_VNXe_FileServerConfigurationServiceCapabilitiesLeaf

EMC_VNXe_CIFSServer_FileServerSettings_SettingsDefineStateAssocLeaf

**Table 422 - Referenced properties/methods for
EMC_VNXe_CIFSServer_FileServerSettings_SettingsDefineStateAssocLeaf**

CIM property	Description/notes
ManagedElement	Reference of EMC_VNXe_CIFSServerLeaf
SettingData	Reference of EMC_VNXe_FileServerSettingsLeaf

EMC_VNXe_NFSServer_FileServerSettings_SettingsDefineStateAssocLeaf

**Table 423 - Referenced properties/methods for
EMC_VNXe_NFSServer_FileServerSettings_SettingsDefineStateAssocLeaf**

CIM property	Description/notes
ManagedElement	Reference of EMC_VNXe_NFSServerLeaf
SettingData	Reference of EMC_VNXe_FileServerSettingsLeaf

EMC_VNXe_FileServerSettings_CIFSSettingData_ConcreteComponentAssocLeaf

**Table 424 - Referenced properties/methods for
EMC_VNXe_FileServerSettings_CIFSSettingData_ConcreteComponentAssocLeaf**

CIM property	Description/notes
GroupComponent	Reference of EMC_VNXe_FileServerSettingsLeaf
PartComponent	Reference of EMC_VNXe_CIFSSettingDataLeaf

EMC_VNXe_FileServerSettings_DNSSettingData_ConcreteComponentAssocLeaf

**Table 425 - Referenced properties/methods for
EMC_VNXe_FileServerSettings_DNSSettingData_ConcreteComponentAssocLeaf**

CIM property	Description/notes
GroupComponent	Reference of EMC_VNXe_FileServerSettingsLeaf
PartComponent	Reference of EMC_VNXe_DNSSettingDataLeaf

EMC_VNXe_FileServerSettings_IPInterfaceSettingData_ConcreteComponentAssocLeaf

**Table 426 - Referenced properties/methods for
EMC_VNXe_FileServerSettings_IPInterfaceSettingData_ConcreteComponentAssocLeaf**

CIM property	Description/notes
GroupComponent	Reference of EMC_VNXe_FileServerSettingsLeaf
PartComponent	Reference of EMC_VNXe_IPInterfaceSettingDataLeaf

EMC_VNXe_FileServerSettings_NFSSettingData_ConcreteComponentAssocLeaf

**Table 427 - Referenced properties/methods for
EMC_VNXe_FileServerSettings_NFSSettingData_ConcreteComponentAssocLeaf**

CIM property	Description/notes
GroupComponent	Reference of EMC_VNXe_FileServerSettingsLeaf
PartComponent	Reference of EMC_VNXe_NFSSettingDataLeaf

EMC_VNXe_FileServerSettings_NISSettingData_ConcreteComponentAssocLeaf

**Table 428 - Referenced properties/methods for
EMC_VNXe_FileServerSettings_NISSettingData_ConcreteComponentAssocLeaf**

CIM property	Description/notes
GroupComponent	Reference of EMC_VNXe_FileServerSettingsLeaf
PartComponent	Reference of EMC_VNXe_NISSettingDataLeaf

EMC_VNXe_FileServerCapabilities_FileServerSettings_SettingsDefineCapabilitiesAssocLeaf

**Table 429 - Referenced properties/methods for
EMC_VNXe_FileServerCapabilities_FileServerSettings_SettingsDefineCapabilitiesAssocLeaf**

CIM property	Description/notes
GroupComponent	Reference of EMC_VNXe_FileServerCapabilitiesLeaf
PartComponent	Reference of EMC_VNXe_FileServerSettingsLeaf

EMC_VNXe_FileServerCapabilities_CIFSSettingData_SettingsDefineCapabilitiesAssocLeaf

**Table 430 - Referenced properties/methods for
EMC_VNXe_FileServerCapabilities_CIFSSettingData_SettingsDefineCapabilitiesAssocLeaf**

CIM property	Description/notes
GroupComponent	Reference of EMC_VNXe_FileServerCapabilitiesLeaf
PartComponent	Reference of EMC_VNXe_CIFSSettingDataLeaf

EMC_VNXe_FileServerCapabilities_DNSSettingData_SettingsDefineCapabilitiesAssocLeaf

**Table 431 - Referenced properties/methods for
EMC_VNXe_FileServerCapabilities_DNSSettingData_SettingsDefineCapabilitiesAssocLeaf**

CIM property	Description/notes
GroupComponent	Reference of EMC_VNXe_FileServerCapabilitiesLeaf
PartComponent	Reference of EMC_VNXe_DNSSettingDataLeaf

EMC_VNXe_FileServerCapabilities_IPInterfaceSettingData_SettingsDefineCapabilitiesAssocLeaf

**Table 432 - Referenced properties/methods for
EMC_VNXe_FileServerCapabilities_IPInterfaceSettingData_SettingsDefineCapabilitiesAssocLeaf**

CIM property	Description/notes
GroupComponent	Reference of EMC_VNXe_FileServerCapabilitiesLeaf
PartComponent	Reference of EMC_VNXe_IPInterfaceSettingDataLeaf

EMC_VNXe_FileServerCapabilities_NFSSettingData_SettingsDefineCapabilitiesAssocLeaf

**Table 433 - Referenced properties/methods for
EMC_VNXe_FileServerCapabilities_NFSSettingData_SettingsDefineCapabilitiesAssocLeaf**

CIM property	Description/notes
GroupComponent	Reference of EMC_VNXe_FileServerCapabilitiesLeaf
PartComponent	Reference of EMC_VNXe_NFSSettingDataLeaf

EMC_VNXe_FileServerCapabilities_NISSettingData_SettingsDefineCapabilitiesAssocLeaf

**Table 434 - Referenced properties/methods for
EMC_VNXe_FileServerCapabilities_NISSettingData_SettingsDefineCapabilitiesAssocLeaf**

CIM property	Description/notes
GroupComponent	Reference of EMC_VNXe_FileServerCapabilitiesLeaf
PartComponent	Reference of EMC_VNXe_NISSettingDataLeaf

File Storage Profile

Overview

The File Storage Profile models the mapping of file systems to logical disks. For the NAS Head Profile, each file system shall be established on one logical disk. The relationship between the LocalFileSystem and the LogicalDisk is represented by the ResidesOnExtent association.

The File Storage Profile is a “read-only” profile. That is, the methods for creating, modifying or deleting a LocalFileSystem are external to the File Storage Profile. The SMI-S prescribed way of performing these functions are covered by the Filesystem Manipulation Profile.

NOTE: For more details, refer to *Clause 7: File Storage Profile in Storage Management Technical Specification, Part 4 Filesystems Version 1.6.0, Revision 4.*

Class diagram

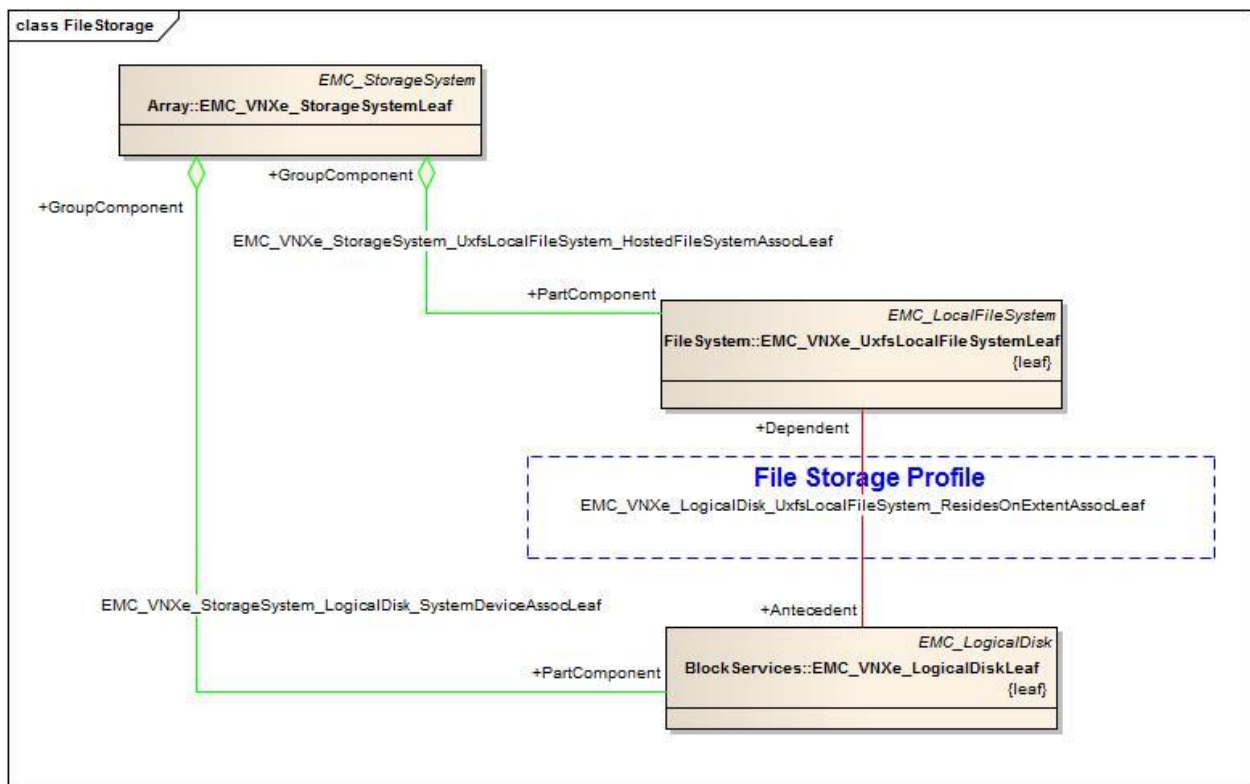


Figure 46 - File Storage Profile Class Diagram

Methods of the Profile

This profile does not include any extrinsic methods.

Client considerations

N/A

CIM Element

The implemented classes and associations related to File Storage Profile in the VNXe storage system are described as follows:

Table 435 - CIM Elements for File Storage Profile

CIM Class	Implemented Class	Description
CIM_ResidesOnExtent	EMC_VNXe_LogicalDisk_UxfsLocalFileSystem_ResidesOnExtentAssocLeaf	Associates the LocalFileSystem and the LogicalDisk where the file system resides.

EMC_VNXe_LogicalDisk_UxfsLocalFileSystem_ResidesOnExtentAssocLeaf

**Table 436 - Referenced properties/methods for
EMC_VNXe_LogicalDisk_UxfsLocalFileSystem_ResidesOnExtentAssocLeaf**

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_LogicalDisk Leaf
Dependent	Reference of EMC_VNXe_UxfsLocalFileSystem Leaf

File System Profile

Overview

The File System Profile is a subprofile for autonomous profiles that support file systems. Specifically, for SMI-S V 1.6, it is the NAS Head Profile.

A number of other profiles and subprofiles make use of elements of the File System Profile. These include but are not limited to the Filesystem Manipulation Subprofile, File Export Subprofile, File Export Manipulation Subprofile, and NAS Head Profile.

NOTE: For more details, refer to [Clause 8: Filesystem Profile in Storage Management Technical Specification, Part 4 Filesystems Version 1.6.0, Revision 4.](#)

Class diagram

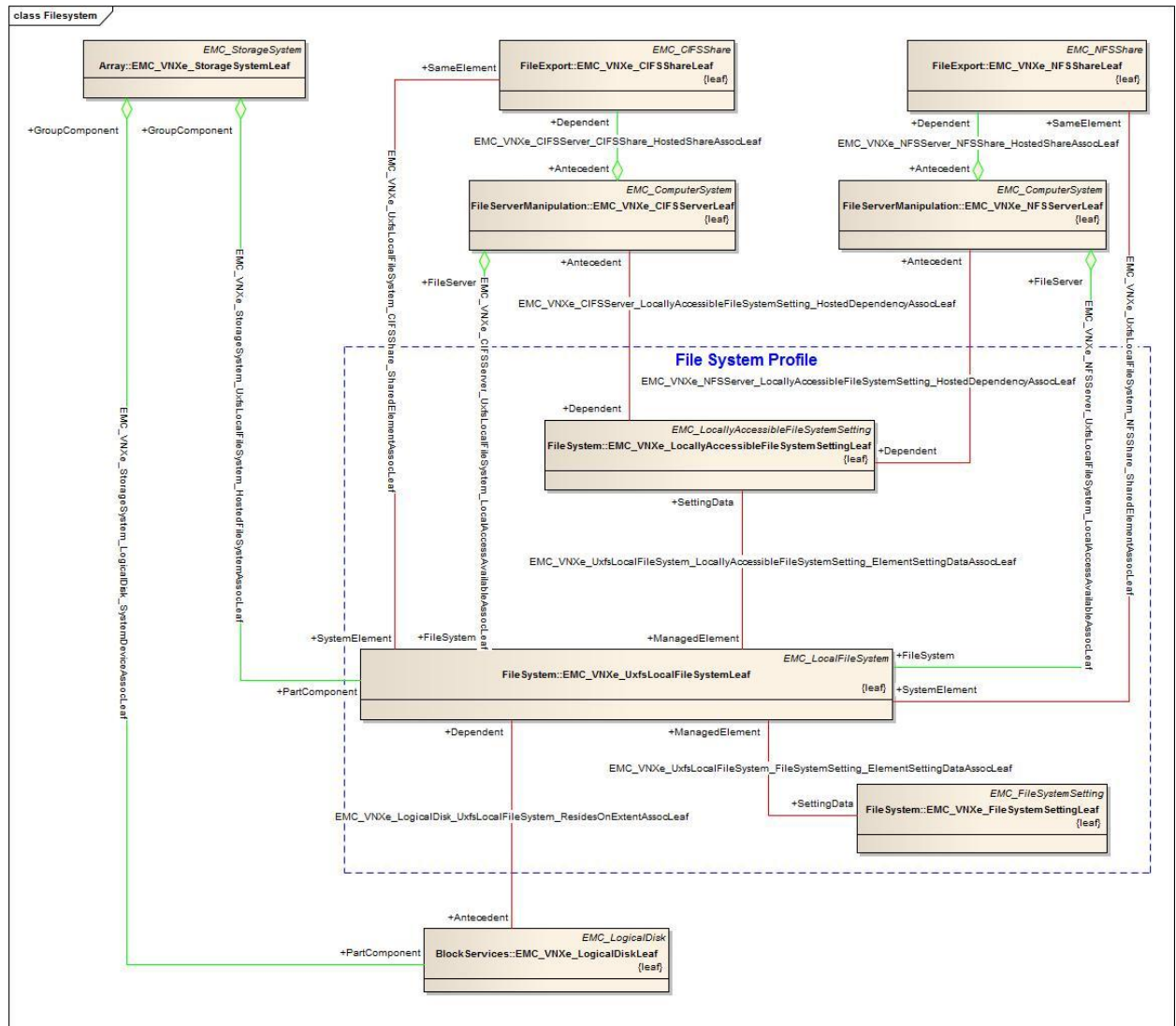


Figure 47 - File System Profile Class Diagram

Methods of the Profile

This profile does not include any extrinsic methods.

Client considerations

Model Specification

- Filesystem Type
Only user-defined UXFS file systems are exposed to the client by the SMI-S Provider in class *EMC_VNXe_UxfsLocalFileSystemLeaf*.
- Filesystem Access Path
Implemented in *LocalAccessPoint* of *ENC_VNXe_CIFSServer_UxfsLocalFileSystem_LocalAccessAvailableAssocLeaf* and *ENC_VNXe_NFSServer_UxfsLocalFileSystem_LocalAccessAvailableAssocLeaf*.
This property specifies the name used by the file server to identify the file system.
- Filesystem Read Write Policy
Implemented in *ReadWritePolicy* of *EMC_VNXe_LocallyAccessibleFileSystemSettingLeaf*.

Use case: Discover all File Systems Hosted by the Referencing Profile

This use case describes how to discover all file systems.

1. From *CIM_ComputerSystem* (the File system Host, also the NAS Head ComputerSystem), traverse *CIM_HostedFileSystem* to *CIM_LocalFileSystem* to discover all File systems hosted by the referencing profile (NAS Head Profile).

Use case: Discover all Exportable File Systems

This use case describes how to discover all exportable file systems.

An exportable file system is a file system that can be exported from a file server to be accessed as a file share.

File systems whose *OperationStatus* contains value 2 are exportable.

Use case: Get the Settings of a File System

This use case describes how to get the settings on a specified file system.

1. From the *CIM_LocalFileSystem* (the File system), traverse *CIM_ElementSettingData* to *CIM_FileSystemSetting* to get the setting of the file system.

Use case: Get the ComputerSystem that Hosts a File System

This use case describes how to get the ComputerSystem that hosts a specified file system.

1. From the *CIM_LocalFileSystem* (the File system), traverse *CIM_HostedFileSystem* to *CIM_ComputerSystem* to get the File system Host.

Use case: Get all File Servers that Have Local Access to a File System and their Access Paths

This use case describes how to get all file servers that have local access to the specified file system and their access paths.

1. From *CIM_LocalFileSystem* (the Filesystem), reference *SNIA_LocalAccessAvailable*.
2. For each result instance of *SNIA_LocalAccessAvailable*
Extract property *FileServer* to get File Server that has local access to the Filesystem.
Extract property *LocalAccessPoint* to get the mount point used by the File Server to access the Filesystem

Use case: Get the Access Path to a File System on the Specified File Server

This use case describes how to get the local access path from the specified file server to the specified file system.

1. From *CIM_LocalFileSystem* (the Filesystem), reference *SNIA_LocalAccessAvailable*.
2. For each result instance of *SNIA_LocalAccessAvailable*
Extract property *FileServer* to see if it references the specified File Server.
If so, extract property *LocalAccessPoint* to get the access point; otherwise check next instance.

Use case: Get the Local Access Settings for a File System on the Specified File Server

This use case describes how to get the local access settings for specified file system on the specified file server.

1. From *CIM_LocalFileSystem* (the Filesystem), traverse *CIM_ElementSettingData* to *SNIA_LocallyAccessibleFileSystemSettingavailable* to get all local access settings for the Filesystem.
2. For each result instance of *SNIA_LocallyAccessibleFileSystemSetting*, traverse *CIM_HostedDependency* to *CIM_ComputerSystem* to get File Server associated to the setting.
3. If the File Server is the one specified, current setting is the one required; otherwise check the next setting instance.

Use case: Get the File Shares of a given File System exported from the specified File Server

This use case describes how to get the file shares of a given file system exported from the specified file server.

1. From *CIM_LocalFileSystem* (the Filesystem), traverse *CIM_SharedElement* to *CIM_FileShare* to get all File Shares of the Filesystem.
2. For each result instance of *CIM_FileShare*, traverse *CIM_HostedShare* to *CIM_ComputerSystem* to get the File Server from which the File Share is exported.
3. If the File Server is the one specified, add current share to result.

CIM Elements

The implemented classes and associations related to File System Profile in the VNXe storage system are described as follows:

Table 437 - CIM Elements for File System Profile

CIM Class	Implemented Class	Description
CIM_ElementSettingData	EMC_VNXe_UxfsLocalFileSystem_FileSystemSetting_ElementSettingDataAssocLeaf	Represents the association between a Filesystem and its FileSystemSetting element.
CIM_ElementSettingData	EMC_VNXe_UxfsLocalFileSystem_LocallyAccessibleFileSystemSetting_ElementSettingDataAssocLeaf	Represents the association between a LocalFileSystem to LocallyAccessibleFileSystemSetting elements, one for each file server that has local access.
CIM_HostedDependency	EMC_VNXe_CIFSServer_LocallyAccessibleFileSystemSetting_HostedDependencyAssocLeaf	Represents the association between a locally accessible setting of a Filesystem and the CIFS File Server that can locally access the Filesystem.
CIM_HostedDependency	EMC_VNXe_NFSServer_LocallyAccessibleFileSystemSetting_HostedDependencyAssocLeaf	Represents the association between a locally accessible setting of a Filesystem and the NFS File Server that can locally access the Filesystem.
CIM_HostedFileSystem	EMC_VNXe_StorageSystem_UxfsLocalFileSystem_HostedFileSystemAssocLeaf	Represents the association between a Filesystem and the ComputerSystem that hosts it.
SNIA_FileSystemSetting	EMC_VNXe_FileSystemSettingLeaf	Represents the configuration settings of a Filesystem.
SNIA_LocalAccessAvailable	EMC_VNXe_CIFSServer_UxfsLocalFileSystem_LocalAccessAvailableAssocLeaf	Represents the association between a Filesystem and the CIFS File Server that can locally access the Filesystem.
SNIA_LocalAccessAvailable	EMC_VNXe_NFSServer_UxfsLocalFileSystem_LocalAccessAvailableAssocLeaf	Represents the association between a Filesystem and the NFS File Server that can locally access the Filesystem.
SNIA_LocalFileSystem	EMC_VNXe_UxfsLocalFileSystemLeaf	Represents a filesystem in a FileSystem-related profile.
SNIA_LocallyAccessibleFileSystemSetting	EMC_VNXe_LocallyAccessibleFileSystemSettingLeaf	Represents the configuration settings of a LocalFileSystem that can be made locally accessible from a File Server ComputerSystem. This Setting provides further details on the functionality supported and the parameters of that functionality when locally accessible.

EMC_VNXe_UxfsLocalFileSystemLeaf

Table 438 - Referenced properties/methods for EMC_VNXe_UxfsLocalFileSystemLeaf

CIM property	Description/notes
CSCreationClassName	Set to 'EMC_VNXe_StorageSystemLeaf'.
CSName	Name of the system.

CIM property	Description/notes
CreationClassName	Set to 'EMC_VNXe_UxfsLocalFileSystemLeaf'
Name	FriendlyID of the FileSystem.
ElementName	Name of the FileSystem.
OperationalStatus	Current operational status of this Local File System.
CaseSensitive	Whether this FileSystem is sensitive to case of characters in filenames. Set to TRUE.
CasePreserved	Whether this FileSystem implementation preserves the case of characters in filenames when saving and restoring. Set to TRUE.
MaxFileNameLength	Length of the longest filename supported in implementation.
FileSystemType	Set to 'EMC_UXFS'
LocalAccessDefinitionRequired	Whether local access definition is required. Set to {4: Not Required}
PathNameSeparatorString	Set to '/'
NumberOfFiles	Number of Files in the FileSystem.
BlockSize	FileSystem block size for data storage and retrieval.
AvailableSpace	Total amount of free space for the FileSystem in bytes.
FileSystemSize	Total size of the FileSystem in bytes.

EMC_VNXe_FileSystemSettingLeaf

Table 439 - Referenced properties/methods for EMC_VNXe_FileSystemSettingLeaf

CIM property	Description/notes
InstanceID	Unique ID of this instance.
ElementName	User friendly name of this instance.
ActualFileSystemType	Set to {21: EMC_UXFS}
FileNameCaseAttributes	Set to {1: Case Sensitive}
NumberOfObjects	Set to empty array.
NumberOfObjectsMax	Set to empty array.
NumberOfObjectsMin	Set to empty array.

CIM property	Description/notes
ObjectSize	Set to empty array.
ObjectSizeMax	Set to empty array.
ObjectSizeMin	Set to empty array.
ObjectTypes	Set to empty array.

EMC_VNXe_LocallyAccessibleFileSystemSettingLeaf

Table 440 - Referenced properties/methods for EMC_VNXe_LocallyAccessibleFileSystemSettingLeaf

CIM property	Description/notes
InstanceID	Unique ID of this instance.
ElementName	User friendly name of this instance.
ReadWritePolicy	Set to {3: Read/Write}

EMC_VNXe_UxfsLocalFileSystem_LocallyAccessibleFileSystemSetting_ElementSettingDataAssocLeaf

Table 441 - Referenced properties/methods for EMC_VNXe_UxfsLocalFileSystem_LocallyAccessibleFileSystemSetting_ElementSettingDataAssocLeaf

CIM property	Description/notes
ManagedElement	Reference of EMC_VNXe_UxfsLocalFileSystemLeaf
SettingData	Reference of EMC_VNXe_LocallyAccessibleFileSystemSettingLeaf

EMC_VNXe_UxfsLocalFileSystem_FileSystemSetting_ElementSettingDataAssocLeaf

Table 442 - Referenced properties/methods for EMC_VNXe_UxfsLocalFileSystem_FileSystemSetting_ElementSettingDataAssocLeaf

CIM property	Description/notes
ManagedElement	Reference of EMC_VNXe_UxfsLocalFileSystemLeaf
SettingData	Reference of EMC_VNXe_FileSystemSettingLeaf

EMC_VNXe_StorageSystem_UxfsLocalFileSystem_HostedFileSystemAssocLeaf

Table 443 - Referenced properties/methods for EMC_VNXe_StorageSystem_UxfsLocalFileSystem_HostedFileSystemAssocLeaf

CIM property	Description/notes
--------------	-------------------

CIM property	Description/notes
GroupComponent	Reference of EMC_VNXe_StorageSystemLeaf
PartComponent	Reference of EMC_VNXe_UxfsLocalFileSystemLeaf

EMC_VNXe_CIFSServer_UxfsLocalFileSystem_LocalAccessAvailableAssocLeaf

**Table 444 - Referenced properties/methods for
EMC_VNXe_CIFSServer_UxfsLocalFileSystem_LocalAccessAvailableAssocLeaf**

CIM property	Description/notes
FileServer	Reference of EMC_VNXe_CIFSServerLeaf
FileSystem	Reference of EMC_VNXe_UxfsLocalFileSystemLeaf

EMC_VNXe_NFSServer_UxfsLocalFileSystem_LocalAccessAvailableAssocLeaf

**Table 445 - Referenced properties/methods for
EMC_VNXe_NFSServer_UxfsLocalFileSystem_LocalAccessAvailableAssocLeaf**

CIM property	Description/notes
FileServer	Reference of EMC_VNXe_NFSServerLeaf
FileSystem	Reference of EMC_VNXe_UxfsLocalFileSystemLeaf

EMC_VNXe_CIFSServer_LocallyAccessibleFileSystemSetting_HostedDependencyAssocLeaf

**Table 446 - Referenced properties/methods for
EMC_VNXe_CIFSServer_LocallyAccessibleFileSystemSetting_HostedDependencyAssocLeaf**

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_CIFSServerLeaf
Dependent	Reference of EMC_VNXe_LocallyAccessibleFileSystemSettingLeaf

EMC_VNXe_NFSServer_LocallyAccessibleFileSystemSetting_HostedDependencyAssocLeaf

**Table 447 - Referenced properties/methods for
EMC_VNXe_NFSServer_LocallyAccessibleFileSystemSetting_HostedDependencyAssocLeaf**

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_NFSServerLeaf
Dependent	Reference of EMC_VNXe_LocallyAccessibleFileSystemSettingLeaf

File System Manipulation Subprofile

Overview

The File System Manipulation Profile is a subprofile that provides support for configuring and manipulating file systems in the context of file system profiles consisting of the NAS Head Profile. A number of other profiles and subprofiles make use of elements of the file system profiles.

NOTE: For more details, refer to *Clause 10: File System Manipulation Profile in Storage Management Technical Specification, Part 4 Filesystems Version 1.6.0, Revision 4.*

Class diagram

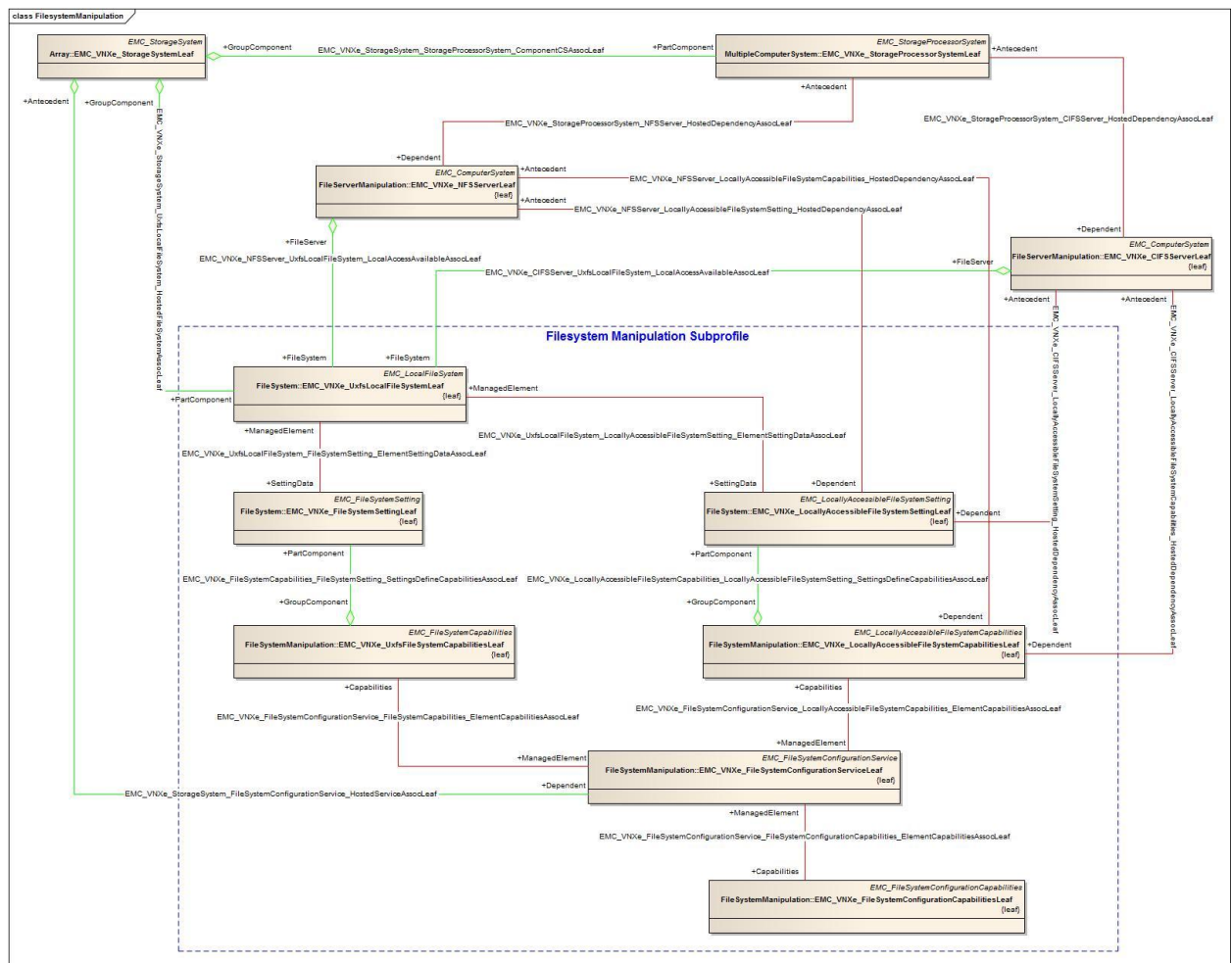


Figure 48 - File System Manipulation Subprofile Class Diagram

Methods of the Profile

Extrinsic Methods on Capabilities

Method: EMC_VNXe_UxfsFileSystemCapabilitiesLeaf.CreateGoalSettings

- Description**

This extrinsic method of the FileSystemCapabilities class validates support for a caller-proposed FileSystemSetting passed as the TemplateGoalSettings parameter.

NOTE:

- 1) This profile restricts the usage of this method to a single entry array for both TemplateGoalSettings and SupportedGoalSettings parameters.
- 2) The Supported Goal Settings output by this method is always the same settings, whatever client inputs in TemplateGoalSettings and SupportedGoalSettings. That is client cannot specify any properties in SNIA_FileSystemSettings.

- Parameters**

Table 448 - Signature and Parameters of EMC_VNXe_UxfsFileSystemCapabilitiesLeaf.CreateGoalSettings

Parameter	Qualifiers	Type	Description
TemplateGoalSettings	IN, EmbeddedInstance, NULL Allowed	CIM_SettingDataRef[]	String array containing embedded instances of class FileSystemSetting, or a derived class. Specifies the client's requirements and is used to locate matching settings that the implementation can support.
SupportedGoalSettings	IN, Out, EmbeddedInstance	CIM_SettingDataRef[]	String array containing embedded instances of class FileSystemSetting, or a derived class. On input, it specifies a previously returned set of Settings that the implementation could support. On output, it specifies a new set of Settings that the implementation can support. If the output set is identical to the input set, both client and implementation may conclude that this is the best match for the TemplateGoalSettings that is available..

- Return Results**

Table 449 - Possible return code of EMC_VNXe_UxfsFileSystemCapabilitiesLeaf.CreateGoalSettings

Return code	Type	Description
0	uint32	Success
NA	CIM_Error	Exceptions will be thrown when any of the below cases is met: <ul style="list-style-type: none"> Any of the required parameters in SNIA Spec is NULL.

Return code	Type	Description
		<ul style="list-style-type: none"> Any of the input parameters is not of the right CIM/SNIA type defined in the MOF files.

Method: EMC_VNXe_LocallyAccessibleFileSystemCapabilitiesLeaf.CreateGoalSettings

- Description**

This extrinsic method of the LocallyAccessibleFileSystemCapabilities class validates support for a caller-proposed LocallyAccessibleFileSystemSetting passed as the TemplateGoalSettings parameter.

NOTE:

- The property - ReadWritePolicy in supported SNIA_LocallyAccessibleFileSystemSetting is always "Read/Write".

- Parameters**

Table 450 - Signature and Parameters of EMC_VNXe_LocallyAccessibleFileSystemCapabilitiesLeaf.CreateGoalSettings

Parameter	Qualifiers	Type	Description
TemplateGoalSettings	IN, EmbeddedInstance, NULL Allowed	CIM_SettingDataRef[]	String array containing embedded instances of class LocallyAccessibleFileSystemSetting, or a derived class. Specifies the client's requirements and is used to locate matching settings that the implementation can support.
SupportedGoalSettings	IN, Out, EmbeddedInstance	CIM_SettingDataRef[]	String array containing embedded instances of class LocallyAccessibleFileSystemSetting, or a derived class. On input, it specifies a previously returned set of Settings that the implementation could support. On output, it specifies a new set of Settings that the implementation can support. If the output set is identical to the input set, both client and implementation may conclude that this is the best match for the TemplateGoalSettings that is available..

- Return Results**

Table 451 - Possible return code of EMC_VNXe_LocallyAccessibleFileSystemCapabilitiesLeaf.CreateGoalSettings

Return code	Type	Description
0	uint32	Success
NA	CIM_Error	Exceptions will be thrown when any of the below cases is met:

Return code	Type	Description
		<ul style="list-style-type: none"> Any of the required parameters in SNIA Spec is NULL. Any of the input parameters is not of the right CIM/SNIA type defined in the MOF files.

Extrinsic Methods on FileSystemConfigurationService

Method: EMC_VNXe_FileSystemConfigurationServiceLeaf.SNIA_CreateFileSystem

- Description**

This extrinsic method allows client to create Locally Accessible Filesystem on StoragePool with a set of caller-proposed Settings.

NOTE: This method must meet the following specifications::

- Filesystem is ONLY supported to be created from StoragePool rather than StorageExtent.
- Filesystem is ONLY supported to be created with capacity equal or more than 1GB and less than 16TB.
- Filesystem is ONLY supported to be created as a Locally Accessible Filesystem.
 - FileServer must be provided to setup local access attributes.
 - ReadWritePolice is only supported with {3: Read/Write}.
 - LocalAccessPoint cannot be specified by client. The input value for it will be ignored.
- Filesystem name specified in ElementName should follow the pattern:
 - A file system name CAN include upper and lowercase letters, numbers, hyphens (-), underscores (_), and periods (.).
 - A file system name can NOT begin with a hyphen, period, or the word root or include a blank character or a colon (:).
 - A file system name can NOT be a single integer or be comprised entirely of integers. Alphanumeric names are accepted.
 - A file system name is limited to 63 bytes.

- Parameters**

Table 452 - Signature and Parameters of EMC_VNXe_FileSystemConfigurationServiceLeaf.SNIA_CreateFileSystem

Parameter	Qualifiers	Type	Description
ElementName	IN, Required	String	End user relevant name for the FileSystem being created.
Job	OUT	CIM_ConcreteJobRef	Reference of the concrete job if this method is asynchronously supported; otherwise it will be NULL.
TheElement	OUT	CIM_LogicalElementRef	Reference to the newly created FileSystem if this method is synchronously supported; otherwise it will be NULL and client needs to use Job to check the affected element.
Goal	IN,	CIM_FileSystem	Setting FileSystemSettings is not supported..

	EmbeddedInst ance, Ignored	SettingRef	
InExtents	IN, Ignored	CIM_StorageExt entRef[]	Ignored because VNXe only supports creating Filesystem from StoragePool.
Pools	IN, Required	CIM_StoragePo olRef[]	References to concrete StoragePools from which LogicalDisk will be created automatically and on which the Filesystem being created will reside. NOTE: <ul style="list-style-type: none"> ONLY one StoragePool will be used when creating Filesystem. Therefore, whenever the client provides more than one pool in Pools, ONLY the 1st StoragePool will be used and rest of the StoragePools will be ignored.
Sizes	IN, OUT, Required	uint64[]	As input, this parameter specifies the desired size of the LogicalDisk storage element corresponding to the 1 st pool passed in Pools[]. The achieved Size is returned as the output parameter. NOTE: <ul style="list-style-type: none"> Only the 1st value in Sizes[] will be used since only the 1st pool is cared.
ExtentSettings	IN, Ignored	CIM_StorageSet tingRef[]	Ignored because VNXe only supports creating Filesystem from StoragePool.
FileServer	IN, Required	CIM_ComputerS ystemRef	Reference to the FileServer ComputerSystem that will have local access to the FileSystem and will be able to export shares from it.
LocalAccessPoint	IN, Ignored, OUT	String	Opaque string to use as a pathname in the name space of the ComputerSystem indicated by the FileServer parameter. NOTE: <ul style="list-style-type: none"> On input, this parameter will be ignored. Client cannot specify the mount point. On output, the actual local access path will be put in this parameter to output.
LocalAccessSetting	IN, OUT, EmbeddedInst ance, NULL allowed	SNIA_LocallyAc cessibleFileSyst emSettingRef	Settings to use for making the filesystem locally accessible from the File Server ComputerSystem. NOTE: <ul style="list-style-type: none"> If input is NULL or empty, default setting scoped to the FileServer will be returned.

DirectoryServer	IN, Ignored	CIM_ComputerSystemRef	Reference to Directory Server. Ignored.
-----------------	----------------	-----------------------	--

- **Return Results**

**Table 453 - Possible return code of
EMC_VNXe_FileSystemConfigurationServiceLeaf.SNIA_CreateFileSystem**

Return code	Type	Description
0	uint32	Job Completed with No Error
4096	Uint32	Method Parameters Checked – Job Started
1	Uint32	Failed: <ul style="list-style-type: none"> • Invalid Parameters. • There is insufficient space in the pool to create the filesystem. • Any error occurs when calling underlying component.
NA	CIM_Error	Exceptions will be thrown when any of the below cases is met: <ul style="list-style-type: none"> • Any of the required parameters in SNIA Spec is NULL. • Any of the input parameters is not of the right CIM/SNIA type defined in the MOF files. • Any of the input references cannot be found in the system.

Method: EMC_VNXe_FileSystemConfigurationServiceLeaf.SNIA_ModifyFileSystem

- **Description**

This extrinsic method allows the client to modify aspects of a Locally Accessible Filesystem.

NOTE: This method must meet the following specifications:

- 1) Filesystem is ONLY supported to be renamed or extended.
- 2) Filesystem is ONLY supported to be extended in the StoragePool in which it was created.

- **Parameters**

**Table 454 - Signature and Parameters of
EMC_VNXe_FileSystemConfigurationServiceLeaf.SNIA_ModifyFileSystem**

Parameter	Qualifiers	Type	Description
ElementName	IN, Optional	String	New name of the Filesystem.
Job	OUT	CIM_ConcreteJobRef	Reference of the concrete job if this method is asynchronously supported; otherwise it will be NULL.

TheElement	IN, Required	CIM_LogicalElementRef	Reference to the FileSystem to be modified.
Goal	IN, EmbeddedInstance, Ignored	CIM_FileSystemSettingRef	Setting FileSystemSettings is not supported..
InExtents	IN, Ignored	CIM_StorageExtentRef[]	Ignored because VNXe only supports creating Filesystem from StoragePool.
Pools	IN, Optional	CIM_StoragePoolRef[]	Reference to the concrete StoragePool in which the FileSystem will be extended. NOTE: <ul style="list-style-type: none"> ONLY one StoragePool will be used when extending the Filesystem. Therefore, whenever the client provides more than one pool in Pools, ONLY the 1st StoragePool will be used and rest of the StoragePools will be ignored.
Sizes	IN, OUT, Required if Pools[] provided	uint64[]	Size of the Filesystem that clients want to extent to. NOTE: <ul style="list-style-type: none"> Only the 1st value in Sizes[] will be used since only the 1st pool is cared.
ExtentSettings	IN, Ignored	CIM_StorageSettingRef[]	Ignored because VNXe only supports creating Filesystem from StoragePool.
FileServer	IN, Ignored	CIM_ComputerSystemRef	Modification of Local Access is not supported.
LocalAccessPoint	IN, Ignored	String	Modification of Local Access is not supported.
LocalAccessSetting	IN, EmbeddedInstance, Ignored	SNIA_LocallyAccessibleFileSystemSettingRef	Modification of Local Access is not supported.
InUseOptions	IN, Ignored	Uint16	Ignored.
WaitTime	IN, Ignored	Uint32	Ignored.

- Return Results**

**Table 455 - Possible return code of
EMC_VNXe_FileSystemConfigurationServiceLeaf.SNIA_ModifyFileSystem**

Return code	Type	Description
0	uint32	Job Completed with No Error
4096	UInt32	Method Parameters Checked – Job Started
1	UInt32	Failed: <ul style="list-style-type: none"> Invalid Parameters. There is insufficient space in the pool to extend the file system. Any error occurs when calling underlying component.
NA	CIM_Error	Exceptions will be thrown when any of the below cases is met: <ul style="list-style-type: none"> Any of the required parameters in SNIA Spec is NULL. Any of the input parameters is not of the right CIM/SNIA type defined in the MOF files. Any of the input references cannot be found in the system.

Method: EMC_VNXe_FileSystemConfigurationServiceLeaf.DeleteFileSystem

- Description**

This extrinsic method allows the client to delete a file system.

- Parameters**

**Table 456 - Signature and Parameters of
EMC_VNXe_FileSystemConfigurationServiceLeaf.DeleteFileSystem**

Parameter	Qualifiers	Type	Description
Job	OUT	CIM_Concrete JobRef	Reference of the concrete job if this method is asynchronously supported; otherwise it will be NULL.
TheElement	IN, Required	EMC_VNXe_U xfsLocalFileSy stemLeafRef	Reference to the FileSystem to be deleted.
InUseOptions	IN, Ignored	UInt16	Ignored.
WaitTime	IN, Ignored	UInt32	Ignored.

- Return Results**

**Table 457 - Possible return code of
EMC_VNXe_FileSystemConfigurationServiceLeaf.DeleteFileSystem**

Return code	Type	Description
0	uint32	Job Completed with No Error
4096	Uint32	Method Parameters Checked – Job Started
1	Uint32	Failed: <ul style="list-style-type: none"> • Invalid Parameters. • Any error occurs when calling underlying component.
NA	CIM_Error	Exceptions will be thrown when any of the below cases is met: <ul style="list-style-type: none"> • Any of the required parameters in SNIA Spec is NULL. • Any of the input parameters is not of the right CIM/SNIA type defined in the MOF files. • Any of the input references cannot be found in the system.

Client considerations

Model Specification

- Filesystem Exportation Support

Per the VNXe system design, one Filesystem can only support one kind of protocol – CIFS or NFS. It means that the user cannot create a CIFS Share and a NFS Share from a Filesystem or its sub components at the same time.

The type of protocol Filesystem supports is determined by the FileServer parameter when creating the Filesystem.

The type of protocol Filesystem supports cannot be modified via SMI-S.

- Filesystem Storage

The Filesystem created via SMI-S is thin. SMI-S does not support creating a non-thin Filesystem or modifying a Filesystem to non-thin.

Use case: Create Locally Accessible Setting for Filesystem

Locally accessible setting is required when creating a locally accessible filesystem via SMI-S.

Refer to [Use Case: Create a Locally Accessible Filesystem on a StoragePool](#).

Use case: Create a Locally Accessible Filesystem on a StoragePool

This use case describes how to create a file system on a StoragePool.

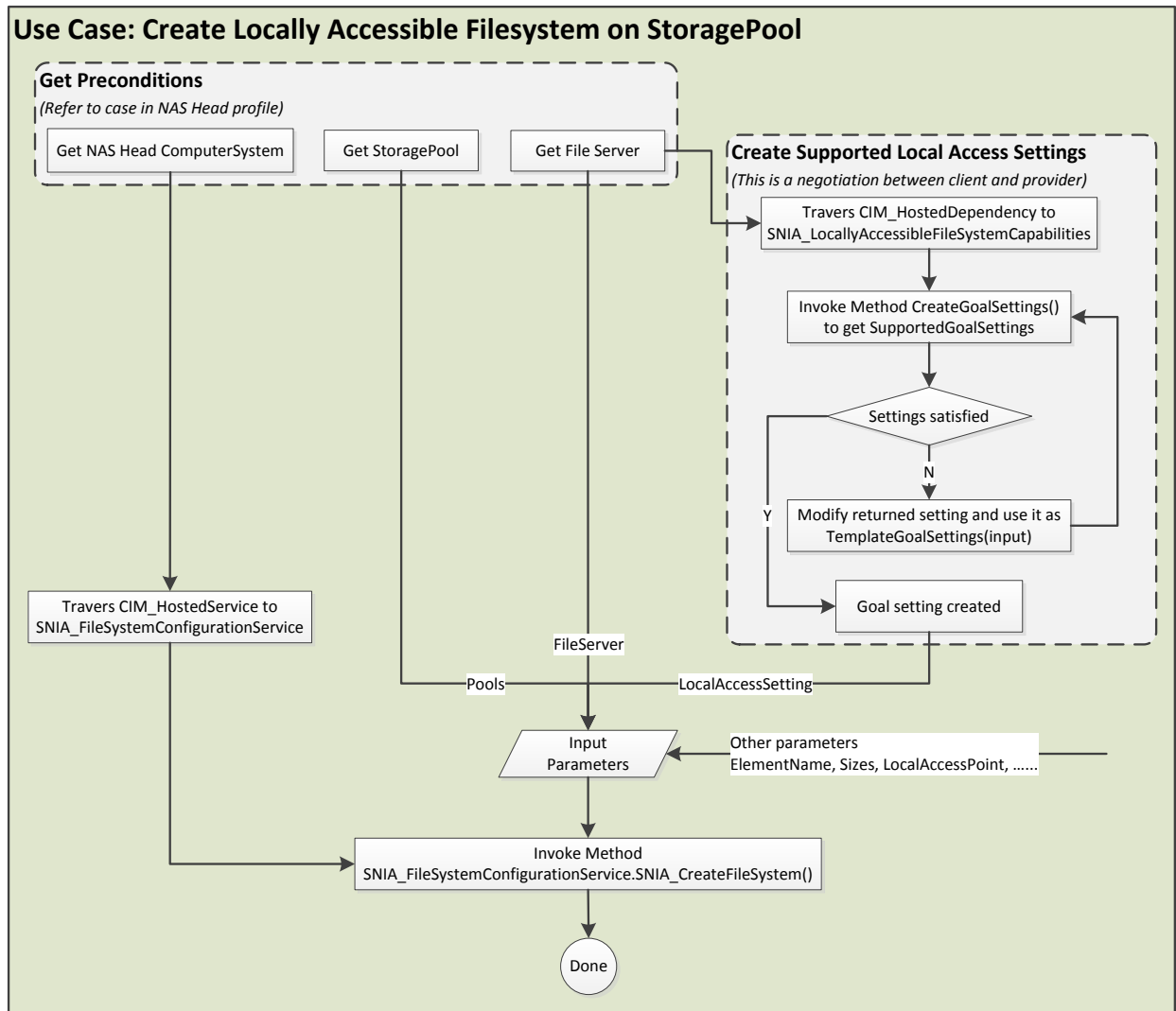


Figure 49 - Flowchart of Creating a File System

Use case: Modify the Name of a Filesystem

This use case describes how to modify the name of a file system.

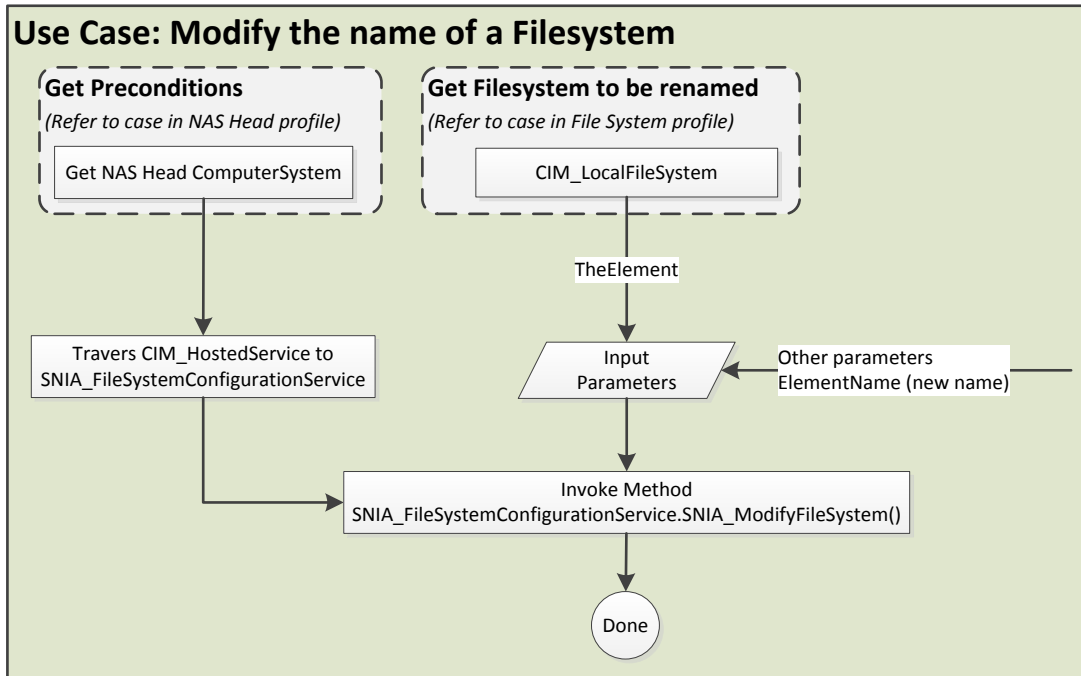


Figure 50 - Flowchart of Modifying Filesystem Name

Use case: Increase the size of a File System

This use case describes how to increase the size of a file system.

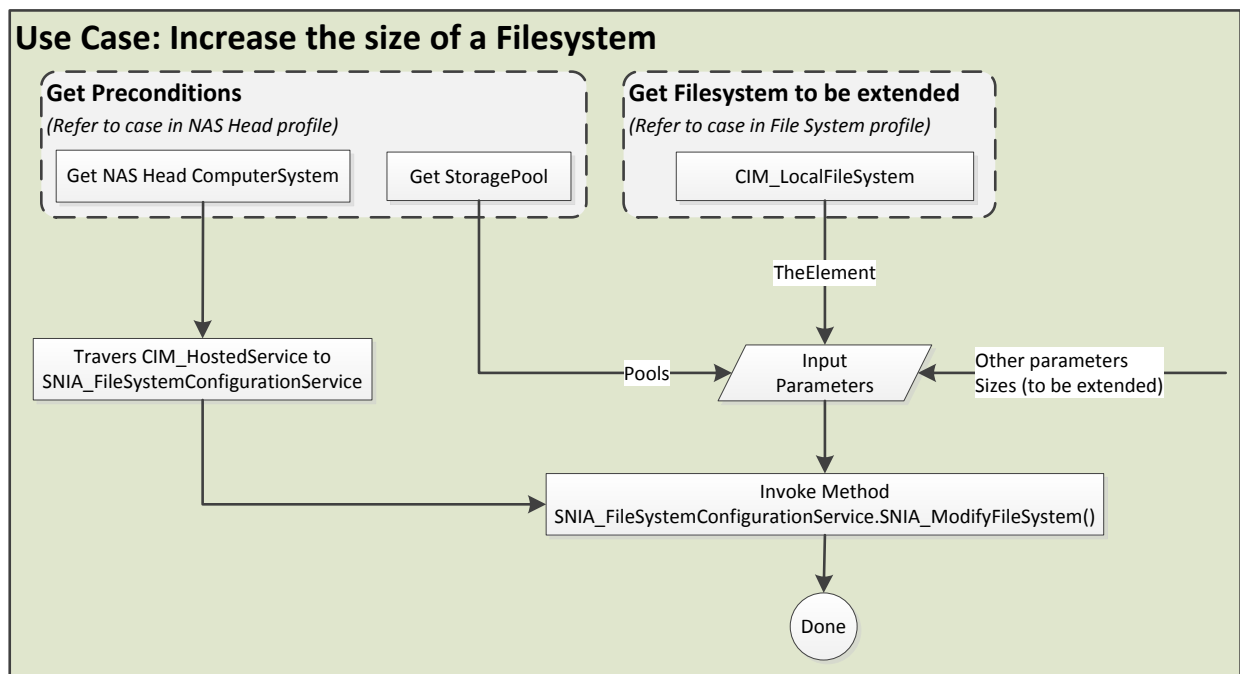


Figure 51 - Flowchart of Extending a File System

Use case: Delete a File System and return underlying StorageExtent

Users can invoke *SNIA_FileSystemConfigurationService.DeleteFileSystem* to delete a Filesystem.

The underlying *StorageExtent* used by the Filesystem will be recycled automatically. This is transparent to users.

CIM Elements

The implemented classes and associations related to File System Manipulation Subprofile in the VNXe storage system are described as follows:

Table 458 - CIM Elements for File System Manipulation Subprofile

CIM Class	Implemented Class	Description
CIM_ElementCapabilities	EMC_VNXe_FileSystemConfigurationService_FileSystemCapabilities_ElementCapabilitiesAssocLeaf	Represents the Filesystem Configuration Service to the FileSystemCapabilities elements that represent all the types of filesystems that are not the default type of file system and can be configured.
CIM_ElementCapabilities	EMC_VNXe_FileSystemConfigurationService_FileSystemConfigurationCapabilities_ElementCapabilitiesAssocLeaf	Represents the association between the Filesystem Configuration Service to the Capabilities element that represents the capabilities that it supports.
CIM_ElementCapabilities	EMC_VNXe_FileSystemConfigurationService_LocallyAccessibleFileSystemCapabilities_ElementCapabilitiesAssocLeaf	Represents the association between the Filesystem Configuration Service to the Capabilities instance that represents the capabilities for Local Access that it supports.
CIM_HostedDependency	EMC_VNXe_CIFSServer_LocallyAccessibleFileSystemCapabilities_HostedDependencyAssocLeaf	Represents the association between a predefined SNIA_LocallyAccessibleFileSystemSetting and the file server ComputerSystem that provides the operational scope for its functionality.
CIM_HostedDependency	EMC_VNXe_NFSServer_LocallyAccessibleFileSystemSetting_HostedDependencyAssocLeaf	Represents the association between a Local Access Capabilities to the File Server that provides the operational scope for its functionality.
CIM_HostedService	EMC_VNXe_StorageSystem_FileSystemConfigurationService_HostedServiceAssocLeaf	Represents the association between the Filesystem Configuration Service and its hosting ComputerSystem.
SNIA_SettingsDefineCapabilities	EMC_VNXe_FileSystemCapabilities_FileSystemSetting_SettingsDefineCapabilitiesAssocLeaf	Represents the Setting elements that provide detailed information about the FileSystemSettings supported by the associated FileSystemCapabilities element.
SNIA_FileSystemConfigurationService	EMC_VNXe_FileSystemConfigurationCapabilitiesLeaf	Represents the management Capabilities of the Filesystem Configuration Service.
SNIA_FileSystemConfigurationService	EMC_VNXe_FileSystemConfigurationServiceLeaf	Represents the Filesystem Configuration Service that provides the methods to

CIM Class	Implemented Class	Description
		manipulate file systems.
SNIA_LocallyAccessibleFileSystemCapabilities	EMC_VNXe_LocallyAccessibleFileSystemCapabilitiesLeaf	Represents the Local Access configuration Capabilities of the File System Configuration Service.
SNIA_SettingsDefineCapabilities	EMC_VNXe_FileSystemCapabilities_FileSystemSetting_SettingsDefineCapabilitiesAssocLeaf	Represents the Setting elements that provide detailed information about the FileSystemSettings supported by the associated FileSystemCapabilities element.
SNIA_SettingsDefineCapabilities	EMC_VNXe_LocallyAccessibleFileSystemCapabilities_LocallyAccessibleFileSystemSetting_SettingsDefineCapabilitiesAssocLeaf	Represents the Setting elements that are associated to this Capabilities element are scoped to the File Server ComputerSystem that provides the operational context for local access.

EMC_VNXe_FileSystemConfigurationServiceLeaf

Table 459 - Referenced properties/methods for EMC_VNXe_FileSystemConfigurationServiceLeaf

CIM property	Description/notes
SystemCreationClassName	Set to 'EMC_VNXe_StorageSystemLeaf'.
SystemName	Name of the storage system.
CreationClassName	Set to 'EMC_VNXe_FileSystemConfigurationServiceLeaf'
Name	Same as SystemName
ElementName	Set as 'File System Configuration Service'

EMC_VNXe_FileSystemConfigurationCapabilitiesLeaf

Table 460 - Referenced properties/methods for EMC_VNXe_FileSystemConfigurationCapabilitiesLeaf

CIM property	Description/notes
InstanceID	Unique ID for the instance.
ElementName	Set as 'File System Configuration Capabilities'
SupportedActualFileSystemTypes	Set to {21: EMC_UXFS}
SupportedSynchronousMethods	Set to [2: CreateFileSystem, 3: DeleteFileSystem, 4: ModifyFileSystem, 5: CreateGoalSettings]
SupportedAsynchronousMethods	Set as empty array.
BlockStorageCreationSupport	Set to {2: StoragePools Required}

CIM property	Description/notes
LocalAccessibilitySupport	Set to {2: Local Access Required}
DirectoryServerParameterSupported	Set to {2: Not Used}

EMC_VNXe_UxfsFileSystemCapabilitiesLeaf

Table 461 - Referenced properties/methods for EMC_VNXe_UxfsFileSystemCapabilitiesLeaf

CIM property	Description/notes
InstanceID	Unique ID for the instance.
ElementName	User friendly name of the instance.
ActualFileSystemType	Set to {21: EMC_UXFS}
SupportedProperties	Set to [4: FilenameCaseAttributes]

EMC_VNXe_LocallyAccessibleFileSystemCapabilitiesLeaf

Table 462 - Referenced properties/methods for EMC_VNXe_LocallyAccessibleFileSystemCapabilitiesLeaf

CIM property	Description/notes
InstanceID	Unique ID for the instance.
ElementName	User friendly name of the instance.
SupportedProperties	Set to [12: ReadWritePolicy]

EMC_VNXe_StorageSystem_FileSystemConfigurationService_HostedServiceAssocLeaf

Table 463 - Referenced properties/methods for EMC_VNXe_StorageSystem_FileSystemConfigurationService_HostedServiceAssocLeaf

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_StorageSystemLeaf
Dependent	Reference of EMC_VNXe_FileSystemConfigurationServiceLeaf

EMC_VNXe_CIFSServer_LocallyAccessibleFileSystemCapabilities_HostedDependencyAssocLeaf

Table 464 - Referenced properties/methods for EMC_VNXe_CIFSServer_LocallyAccessibleFileSystemCapabilities_HostedDependencyAssocLeaf

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_CIFSServerLeaf
Dependent	Reference of EMC_VNXe_LocallyAccessibleFileSystemCapabilitiesLeaf

EMC_VNXe_NFSServer_LocallyAccessibleFileSystemCapabilities_HostedDependencyAssocLeaf

Table 465 - Referenced properties/methods for EMC_VNXe_NFSServer_LocallyAccessibleFileSystemCapabilities_HostedDependencyAssocLeaf

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_NFSServerLeaf
Dependent	Reference of EMC_VNXe_LocallyAccessibleFileSystemCapabilitiesLeaf

EMC_VNXe_FileSystemConfigurationService_FileSystemCapabilities_ElementCapabilitiesAssocLeaf

Table 466 - Referenced properties/methods for EMC_VNXe_FileSystemConfigurationService_FileSystemCapabilities_ElementCapabilitiesAssocLeaf

CIM property	Description/notes
ManagedElement	Reference of EMC_VNXe_FileSystemConfigurationServiceLeaf
Capabilities	Reference of EMC_VNXe_UxfsFileSystemCapabilitiesLeaf

EMC_VNXe_FileSystemConfigurationService_FileSystemConfigurationCapabilities_ElementCapabilitiesAssocLeaf

Table 467 - Referenced properties/methods for EMC_VNXe_FileSystemConfigurationService_FileSystemConfigurationCapabilities_ElementCapabilitiesAssocLeaf

CIM property	Description/notes
ManagedElement	Reference of EMC_VNXe_FileSystemConfigurationServiceLeaf
Capabilities	Reference of EMC_VNXe_FileSystemConfigurationCapabilitiesLeaf

EMC_VNXe_FileSystemConfigurationService_LocallyAccessibleFileSystemCapabilities_ElementCapabilitiesAssocLeaf

Table 468 - Referenced properties/methods for EMC_VNXe_FileSystemConfigurationService_LocallyAccessibleFileSystemCapabilities_ElementCapabilitiesAssocLeaf

CIM property	Description/notes
ManagedElement	Reference of EMC_VNXe_FileSystemConfigurationServiceLeaf
Capabilities	Reference of EMC_VNXe_LocallyAccessibleFileSystemCapabilitiesLeaf

[EMC_VNXe_FileSystemCapabilities_FileSystemSetting_SettingsDefineCapabilitiesAssocLeaf](#)

Table 469 - Referenced properties/methods for EMC_VNXe_FileSystemCapabilities_FileSystemSetting_SettingsDefineCapabilitiesAssocLeaf

CIM property	Description/notes
GroupComponent	Reference of EMC_VNXe_UxfsFileSystemCapabilitiesLeaf
PartComponent	Reference of EMC_VNXe_FileSystemSettingLeaf

[EMC_VNXe_LocallyAccessibleFileSystemCapabilities_LocallyAccessibleFileSystemSetting_SettingsDefineCapabilitiesAssocLeaf](#)

Table 470 - Referenced properties/methods for EMC_VNXe_LocallyAccessibleFileSystemCapabilities_LocallyAccessibleFileSystemSetting_SettingsDefineCapabilitiesAssocLeaf

CIM property	Description/notes
GroupComponent	Reference of EMC_VNXe_UxfsFileSystemCapabilitiesLeaf
PartComponent	Reference of EMC_VNXe_LocallyAccessibleFileSystemSettingLeaf

File Export Profile

Overview

The File Export Profile is a subprofile for autonomous profiles that support the export of file systems. In this release of SMI-S, this includes the NAS Head Profile.

NOTE: For more details, refer to [Clause 4: File Export Profile in Storage Management Technical Specification, Part 4 Filesystems Version 1.6.0, Revision 4](#).

Class diagram

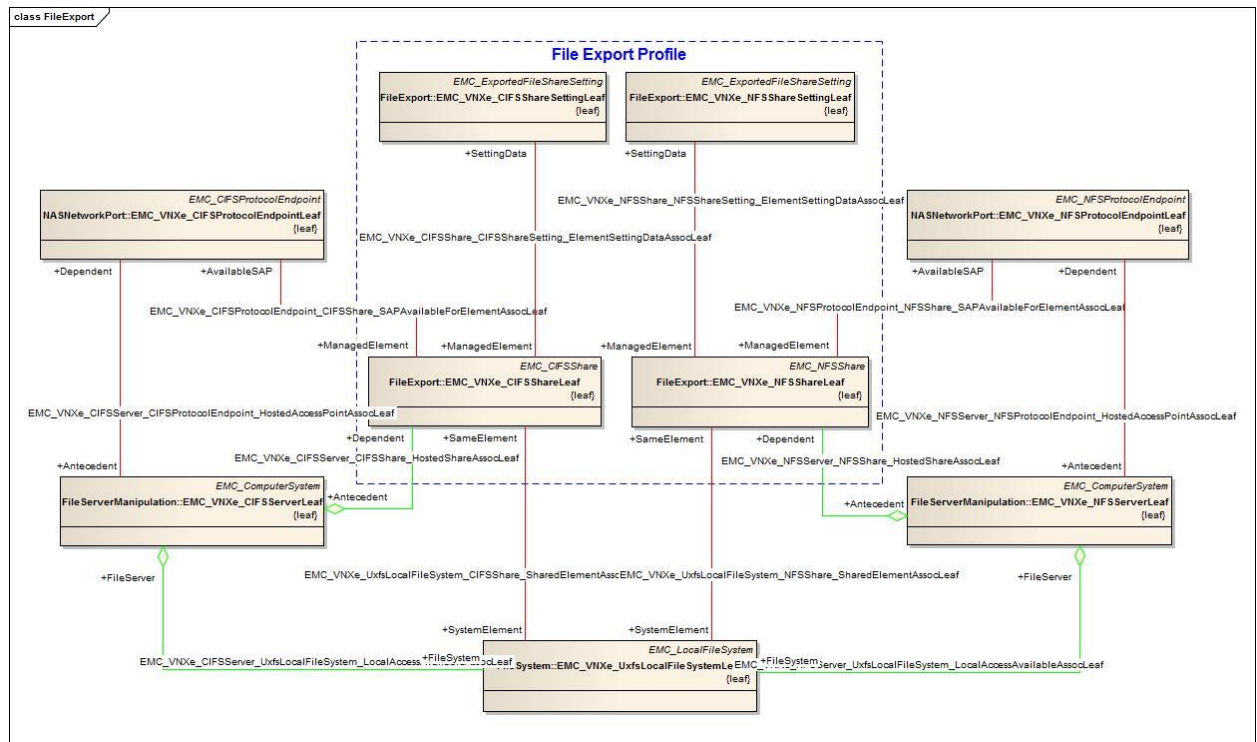


Figure 52 - File Export Profile Class Diagram

Methods of the Profile

This profile does not include any extrinsic methods.

Client considerations

Use case: Discover all File Shares Hosted on a File Server

This use case describes how to discover all the file shares hosted on a file server.

1. From *CIM_ComputerSystem* (the File Server), traverse *CIM_HostedShare* to *CIM_FileShare* to get all File Shares hosted on (exported from) the File Server.

Use case: Discover all File Shares of a File System

This use case describes how to discover all the file shares exported from a file system.

1. From *CIM_LocalFileSystem* (the Filesystem), traverse *CIM_SharedElement* to *CIM_FileShare* to get all File Shares of the Filesystem.

Use case: Get the Settings of a File Share

This use case describes how to get the settings of a file share.

1. From *CIM_FileShare* (the share), traverse *CIM_ElementSettingData* to *CIM_ExportedFileShareSetting* to get settings of the File Shares.

CIM Elements

The implemented classes and associations related to the File Export Profile in the VNXe storage system are described as follows:

Table 471 - CIM Elements for File Export Profile

CIM Class	Implemented Class	Description
CIM_ElementsSettingData	EMC_VNXe_CIFSShare_CIFSShareSetting_ElementSettingDataAssocLeaf	Represents the associates between a CIFS FileShare and ExportedFileShareSetting elements.
CIM_ElementsSettingData	EMC_VNXe_NFSShare_NFSShareSetting_ElementSettingDataAssocLeaf	Represents the associates between a NFS FileShare and ExportedFileShareSetting elements.
CIM_ExportedFileShareSetting	EMC_VNXe_CIFSShareSettingLeaf	Represents the configuration settings for an Exported CIFS FileShare that is a setting for a FileShare available for exporting.
CIM_ExportedFileShareSetting	EMC_VNXe_NFSShareSettingLeaf	Represents the configuration settings for an Exported NFS FileShare that is a setting for a FileShare available for exporting.
CIM_FileShare	EMC_VNXe_CIFSShareLeaf	Represents the sharing characteristics of a particular file element via CIFS protocol.
CIM_FileShare	EMC_VNXe_NFSShareLeaf	Represents the sharing characteristics of a particular file element via NFS protocol.
CIM_HostedShare	EMC_VNXe_CIFSServer_CIFSShare_HostedShareAssocLeaf	Represents that a shared element is hosted by a File Server Computer System via CIFS protocol.
CIM_HostedShare	EMC_VNXe_NFSServer_NFSShare_HostedShareAssocLeaf	Represents that a shared element is hosted by a File Server Computer System via NFS protocol.
CIM_SAPAvailableForElement	EMC_VNXe_CIFSProtocolEndpoint_CIFSShare_SAPAvailableForElementAssocLeaf	Represents the association between a ProtocolEndpoint to the shared element that is being accessed through that SAP.
CIM_SAPAvailableForElement	EMC_VNXe_NFSProtocolEndpoint	Represents the association between a

CIM Class	Implemented Class	Description
	nt_NFSShare_SAPAvailableForElementAssocLeaf	ProtocolEndpoint to the shared element that is being accessed through that SAP.
CIM_SharedElement	EMC_VNXe_UxfsLocalFileSystem_CIFSShare_SharedElementAssocLeaf	Represents the association between a CIFS Share and the LocalFileSystem on which it is based.
CIM_SharedElement	EMC_VNXe_UxfsLocalFileSystem_NFSShare_SharedElementAssocLeaf	Represents the association between a NFS Share and the LocalFileSystem on which it is based.

EMC_VNXe_CIFSShareLeaf

Table 472 - Referenced properties/methods for EMC_VNXe_CIFSShareLeaf

CIM property	Description/notes
InstanceID	Unique ID of this instance.
ElementName	Set as Name.
Name	Name of the share.
OperationalStatus	Current operational status of the share.
SharingDirectory	Indicates the share is a directory or file.
PathNameSeparatorString	Separator string of the share path. Set to '\'

EMC_VNXe_NFSShareLeaf

Table 473 - Referenced properties/methods for EMC_VNXe_NFSShareLeaf

CIM property	Description/notes
InstanceID	Unique ID of this instance.
ElementName	Set as Name.
Name	Name of the share.
OperationalStatus	Current operational status of the share.
SharingDirectory	Indicates the share is a directory or file.
PathNameSeparatorString	Separator string of the share path. Set to '/'

EMC_VNXe_CIFSShareSettingLeaf

Table 474 - Referenced properties/methods for EMC_VNXe_CIFSShareSettingLeaf

CIM property	Description/notes
InstanceID	Unique ID of this instance.
ElementName	User friendly name of this instance.
FileSharingProtocol	File sharing protocol share can support. Set to {3: CIFS}
ProtocolVersion	Supported versions of the sharing protocol.
DefaultReadWrite	Default privilege can be supported for read-write access.
CASupported	Indicates if Continuous Availability (CA) is supported.
RootAccess	Set to {3: Allow Root Access}
AccessPoints	Set to {5: Named Points}
DefaultUserIdSupported	Set to {2: No Default User Id}

EMC_VNXe_NFSShareSettingLeaf

Table 475 - Referenced properties/methods for EMC_VNXe_NFSShareSettingLeaf

CIM property	Description/notes
InstanceID	Unique ID of this instance.
ElementName	User friendly name of this instance.
FileSharingProtocol	File sharing protocol share can support. Set to {2: CIFS}
ProtocolVersion	Supported versions of the sharing protocol.
DefaultReadWrite	Default privilege can be supported for read-write access.
CASupported	Indicates if Continuous Availability (CA) is supported.
RootAccess	Set to {3: Allow Root Access}
AccessPoints	Set to {4: All}
DefaultUserIdSupported	Set to {2: No Default User Id}

EMC_VNXe_CIFSServer_CIFSShare_HostedShareAssocLeaf

Table 476 - Referenced properties/methods for EMC_VNXe_CIFSServer_CIFSShare_HostedShareAssocLeaf

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_CIFSServerLeaf

CIM property	Description/notes
Dependent	Reference of EMC_VNXe_CIFSShareLeaf

EMC_VNXe_NFSServer_NFSShare_HostedShareAssocLeaf

**Table 477 - Referenced properties/methods for
EMC_VNXe_NFSServer_NFSShare_HostedShareAssocLeaf**

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_NFSServerLeaf
Dependent	Reference of EMC_VNXe_NFSShareLeaf

EMC_VNXe_CIFSShare_CIFSShareSetting_ElementSettingDataAssocLeaf

**Table 478 - Referenced properties/methods for
EMC_VNXe_CIFSShare_CIFSShareSetting_ElementSettingDataAssocLeaf**

CIM property	Description/notes
ManagedElement	Reference of EMC_VNXe_CIFSShareLeaf
SettingData	Reference of EMC_VNXe_CIFSShareSettingLeaf

EMC_VNXe_NFSShare_NFSShareSetting_ElementSettingDataAssocLeaf

**Table 479 - Referenced properties/methods for
EMC_VNXe_NFSShare_NFSShareSetting_ElementSettingDataAssocLeaf**

CIM property	Description/notes
ManagedElement	Reference of EMC_VNXe_NFSShareLeaf
SettingData	Reference of EMC_VNXe_NFSShareSettingLeaf

EMC_VNXe_CIFSProtocolEndpoint_CIFSShare_SAPAvailableForElementAssocLeaf

**Table 480 - Referenced properties/methods for
EMC_VNXe_CIFSProtocolEndpoint_CIFSShare_SAPAvailableForElementAssocLeaf**

CIM property	Description/notes
ManagedElement	Reference of EMC_VNXe_CIFSShareLeaf
AvailableSAP	Reference of EMC_VNXe_CIFSProtocolEndpointLeaf

EMC_VNXe_NFSProtocolEndpoint_NFSShare_SAPAvailableForElementAssocLeaf

**Table 481 - Referenced properties/methods for
EMC_VNXe_NFSProtocolEndpoint_NFSShare_SAPAvailableForElementAssocLeaf**

CIM property	Description/notes
ManagedElement	Reference of EMC_VNXe_NFSShareLeaf
AvailableSAP	Reference of EMC_VNXe_NFSProtocolEndpointLeaf

EMC_VNXe_UxfsLocalFileSystem_CIFSShare_SharedElementAssocLeaf

**Table 482 - Referenced properties/methods for
EMC_VNXe_UxfsLocalFileSystem_CIFSShare_SharedElementAssocLeaf**

CIM property	Description/notes
SystemElement	Reference of EMC_VNXe_UxfsLocalFileSystemLeaf
SameElement	Reference of EMC_VNXe_CIFSShareLeaf

EMC_VNXe_UxfsLocalFileSystem_NFSShare_SharedElementAssocLeaf

**Table 483 - Referenced properties/methods for
EMC_VNXe_UxfsLocalFileSystem_NFSShare_SharedElementAssocLeaf**

CIM property	Description/notes
SystemElement	Reference of EMC_VNXe_UxfsLocalFileSystemLeaf
SameElement	Reference of EMC_VNXe_NFSShareLeaf

File Export Manipulation Subprofile

Overview

The File Export Manipulation Profile is a subprofile of autonomous profiles that support file systems. It makes use of elements of the file system subprofiles and supports creation, modification and deletion of FileShares that are exported by the File Export Subprofile. A number of other profiles and subprofiles also make use of elements in the File System subprofile and will be referred to in this specification as “file system related profiles” -- these include but are not limited to the File System subprofile, the File System Manipulation Subprofile, the File Export Profile, and the NAS Head Profile.

NOTE: For more details, refer to *Clause 5: File Export Manipulation Subprofile in Storage Management Technical Specification, Part 4 Filesystems Version 1.6.0, Revision 4.*

Class Diagram

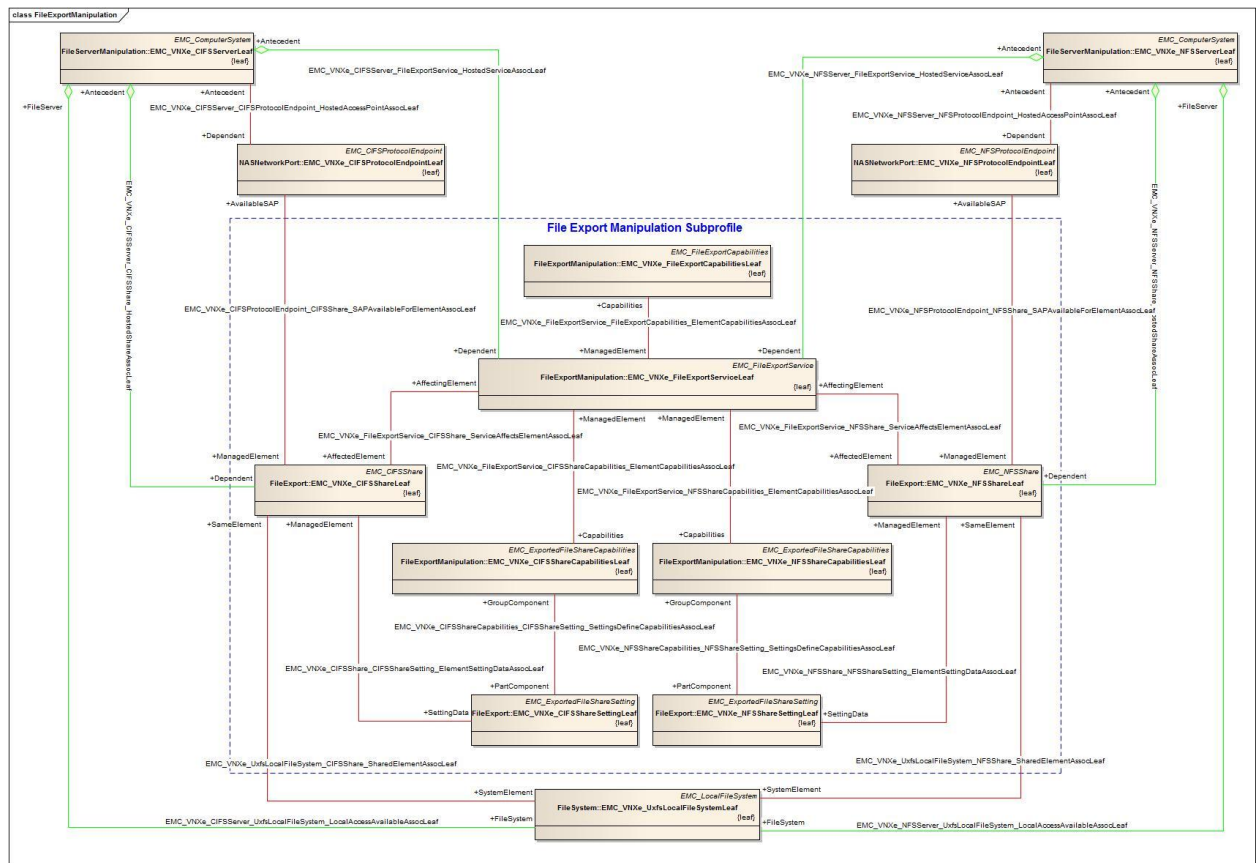


Figure 53 - File Export Manipulation Subprofile Class Diagram

Methods of the Profile

Extrinsic Methods on Capabilities

Method: EMC_VNXe_NFSShareCapabilitiesLeaf.CreateGoalSettings

- Description**

This extrinsic method of the *NFSShareCapabilities* class validates the support for a caller-proposed setting for share via the NFS protocol passed as the *TemplateGoalSettings* parameter.

NOTE:

- Any value for *TemplateGoalSettings* is ignored, that is whatever is passed in *TemplateGoalSettings* and *SupportedGoalSettings*. The client would always get the same settings in *SupportedGoalSettings*.

- Parameters**

Table 484 - Signature and Parameters of EMC_VNXe_NFSShareCapabilitiesLeaf.CreateGoalSettings

Parameter	Qualifiers	Type	Description
TemplateGoalSettings	IN, EmbeddedInstance, Ignored	CIM_SettingDataRef[]	Ignored because the <i>NFSShareSetting</i> is not supported to set by client. TemplateGoalSettings is a string array containing embedded instances of class <i>ExportedFileShareSetting</i> , or a derived class. This parameter specifies the client's requirements and is used to locate matching settings that the implementation can support.
SupportedGoalSettings	IN, Out, EmbeddedInstance	CIM_SettingDataRef[]	On input, this parameter is ignored. On output, it specifies a new setting that the implementation can support. This is always the same settings.

- Return Results**

Table 485 - Possible return code of EMC_VNXe_NFSShareCapabilitiesLeaf.CreateGoalSettings

Return code	Type	Description
0	uint32	Success
NA	CIM_Error	Exceptions will be thrown when any of the below cases is met: <ul style="list-style-type: none"> Any of the required parameters in SNIA Spec is NULL.

Return code	Type	Description
		<ul style="list-style-type: none"> Any of the input parameters is not of the right CIM/SNIA type defined in the MOF files.

Method: EMC_VNXe_CIFSShareCapabilitiesLeaf.CreateGoalSettings

• **Description**

This extrinsic method of the *CIFSShareCapabilities* class validates the support for a caller-proposed setting for share via the CIFS protocol passed as the *TemplateGoalSettings* parameter.

NOTE:

- 1) Clients are only allowed to specify *DefaultReadWrite* and *CASupported* via this method. Other properties will be always the same.

• **Parameters**

Table 486 - Signature and Parameters of EMC_VNXe_CIFSShareCapabilitiesLeaf.CreateGoalSettings

Parameter	Qualifiers	Type	Description
TemplateGoalSettings	IN, EmbeddedInstance, NULL Allowed	CIM_SettingData Ref[]	Only TemplateGoalSettings[0] is used. TemplateGoalSettings is a string array containing embedded instances of class ExportedFileShareSetting, or a derived class. This parameter specifies the client's requirements and is used to locate matching settings that the implementation can support.
SupportedGoalSettings	IN, Out, EmbeddedInstance	CIM_SettingData Ref[]	On input, this parameter is ignored. On output, it specifies a new setting that the implementation can support.

• **Return Results**

Table 487 - Possible return code of EMC_VNXe_CIFSShareCapabilitiesLeaf.CreateGoalSettings

Return code	Type	Description
0	uint32	Success
NA	CIM_Error	<p>Exceptions will be thrown when any of the below cases is met:</p> <ul style="list-style-type: none"> Any of the required parameters in SNIA Spec is NULL. Any of the input parameters is not of the right CIM/SNIA type defined in the MOF files.

Extrinsic Methods on FileExportService

Method: EMC_VNXe_FileExportServiceLeaf.SNIA_CreateExportedShare

- **Description**

This extrinsic method allows the client to create a File Share and export it with a set of caller-proposed settings.

NOTE:

- 1) Property *Goal.FileSharingProtocol* should be consistent with the protocol type supported by the Filesystem specified by *Root* parameter.
- 2) Property *Goal.DefaultReadWrite* is allowed to be specified only if *Goal.FileSharingProtocol* is set to {3: CIFS}.
- 3) Property *Goal.CASupported* is allowed to be specified only if *Goal.FileSharingProtocol* is set to {3: CIFS}.
- 4) Parameter *RootAccessHosts* is allowed to be specified only if *Goal.FileSharingProtocol* is set to {2: NFS}.
- 5) Share name specified in *ElementName* should follow the pattern:
 - a. Share names cannot include the following characters: /, \, %, ", NUL (Null character), STX (start of header), SOT (start of text), and LF (line feed).
 - b. Share names can contain spaces and other non-alpha-numeric characters but must be enclosed by quotes if spaces are used.
 - c. Share names cannot begin with a - (hyphen).
 - d. Share name length is limited to 32 characters.

- **Parameters**

Table 488 - Signature and Parameters of EMC_VNXe_FileExportServiceLeaf.SNIA_CreateExportedShare

Parameter name	Qualifier	Type	Description/note
ElementName	IN, Required	String	An end user relevant name for the FileShare being created.
Comment	IN, NULL allowed	String	Specifies Comment of the FileShare being created.
Job	OUT	CIM_ConcreteJobRef	Reference of the concrete job if this method is asynchronously supported; otherwise it will be NULL.
Root	IN, Required	CIM_LogicalElementRef	Filesystem whose sub-folder is being exported.
SharedElementPath	IN, OUT, NULL allowed	String	Represents a path to the shared element from the root directory of the FileSystem indicated by Root. If not provided, root directory of the Filesystem will be used as the share path.

Goal	IN, EmbeddedInstance, Required	SNIA_ExportedFileShareSettingRef	Setting indicating how the File Share element to be created and exported by the FileExportService. The Goal should be type of either <i>EMC_VNXe_CIFSShareSettingLeaf</i> or <i>EMC_VNXe_NFSShareSettingLeaf</i> , which is output by <i>CreateGoalSettings</i> .
TheShare	OUT	CIM_FileShareRef	Reference to the newly created File Share if this method is synchronously supported; otherwise it will be NULL and client needs to use Job to check the affected element.
DefaultUserId	IN, Ignored	CIM_IdentityRef	Ignored.
RootAccessHosts	IN, Conditional	String[]	Specify the hosts that have root access to this Share, if <i>ExportedFileShareSetting.RootAccess</i> is set to "Allow Root Access".
AccessPointPorts	IN, Ignored	String[]	Ignored.

- **Return Results**

**Table 489 - Possible return code of
EMC_VNXe_FileExportServiceLeaf.SNIA_CreateExportedShare**

Return code	Type	Description
0	uint32	Job Completed with No Error
4096	UInt32	Method Parameters Checked – Job Started
1	UInt32	Failed: <ul style="list-style-type: none"> • Invalid Parameters. • Any error occur when calling underlying component.
NA	CIM_Error	Exceptions will be thrown when any of the below cases is met: <ul style="list-style-type: none"> • Any of the required parameters in SNIA Spec is NULL. • Any of the input parameters is not of the right CIM/SNIA type defined in the MOF files. • Any of the input references cannot be found in the system.

Method: EMC_VNXe_FileExportServiceLeaf.SNIA_ModifyExportedShare

- **Description**

This extrinsic method allows the client to modify aspects of an exported File Share.

NOTE:

- 1) Share name (in parameter ElementName) can NOT be changed.
- 2) Share path (in parameter SharedElementPath) can NOT be changed.
- 3) Property *Goal.DefaultReadWrite* is allowed to be modified only if *TheShare* is a CIFS Share.
- 4) Property *Goal.CASupported* is allowed to be modified only if *TheShare* is a CIFS Share.
- 5) Parameter *RootAccessHosts* is allowed to be modified only if *TheShare* is a NFS Share.

- **Parameters**

**Table 490 - Signature and Parameters of
EMC_VNXe_FileExportServiceLeaf.SNIA_ModifyExportedShare**

Parameter name	Qualifier	Type	Description/note
ElementName	IN, Not Supported	String	Share name cannot be changed.
Comment	IN, NULL allowed	String	Specifies Comment of the FileShare being created.
Job	OUT	CIM_ConcreteJobRef	Reference of the concrete job if this method is asynchronously supported; otherwise it will be NULL.
Root	IN, Ignored	CIM_LogicalElementRef	Ignored.
SharedElementPath	IN, Not Supported	String	Share path cannot be changed.
Goal	IN, EmbeddedInstance, Required	SNIA_ExportedFileShareSettingRef	Setting indicating how the File Share element to be created and exported by the FileExportService.
TheShare	IN, Required	CIM_FileShareRef	Reference to the File Share to be modified.
DefaultUserId	IN, Ignored	CIM_IdentityRef	Ignored.
RootAccessHosts	IN, Conditional	String[]	Specify the hosts that have root access to this Share, if <i>ExportedFileShareSetting.RootAccess</i> is set to "Allow Root Access".
AccessPointPorts	IN, Ignored	String[]	Ignored.
InUseOptions	IN, Ignored	UInt16	Ignored.

WaitTime	IN, Ignored	UInt32	Ignored.
----------	----------------	--------	----------

- **Return Results**

**Table 491 - Possible return code of
EMC_VNXe_FileExportServiceLeaf.SNIA_ModifyExportedShare**

Return code	Type	Description
0	uint32	Job Completed with No Error
4096	UInt32	Method Parameters Checked – Job Started
1	UInt32	Failed: <ul style="list-style-type: none"> • Invalid Parameters. • Any error occurs when calling underlying component.
NA	CIM_Error	Exceptions will be thrown when any of the below cases is met: <ul style="list-style-type: none"> • Any of the required parameters in SNIA Spec is NULL. • Any of the input parameters is not of the right CIM/SNIA type defined in the MOF files. • Any of the input references cannot be found in the system.

Method: EMC_VNXe_FileExportServiceLeaf.ReleaseExportedShare

- **Description**

This extrinsic method allows the client to release an exported File Share.

- **Parameters**

**Table 492 - Signature and Parameters of
EMC_VNXe_FileExportServiceLeaf.ReleaseExportedShare**

Parameter name	Qualifier	Type	Description/note
Job	OUT	CIM_ConcreteJob Ref	Reference of the concrete job if this method is asynchronously supported; otherwise it will be NULL.
TheShare	IN, Required	CIM_FileShareRef	Reference to the File Share to be deleted.
InUseOptions	IN, Ignored	UInt16	Ignored.
WaitTime	IN,	UInt32	Ignored.

	Ignored		
--	---------	--	--

- **Return Results**

**Table 493 - Possible return code of
EMC_VNXe_FileExportServiceLeaf.ReleaseExportedShare**

Return code	Type	Description
0	uint32	Job Completed with No Error
4096	UInt32	Method Parameters Checked – Job Started
1	UInt32	Failed: <ul style="list-style-type: none"> • Invalid Parameters. • Any error occurs when calling underlying component.
NA	CIM_Error	Exceptions will be thrown when any of the below cases is met: <ul style="list-style-type: none"> • Any of the required parameters in SNIA Spec is NULL. • Any of the input parameters is not of the right CIM/SNIA type defined in the MOF files. • Any of the input references cannot be found in the system.

Method: EMC_VNXe_FileExportServiceLeaf.AssignPrivilegeToExportedShare

Refer to [Method: Assign Privilege](#) in [Simple Identity Management Profile](#).

Client considerations

Use case: Check the Supported Capabilities Pattern of a FileExportService

This use case describes how to check the supported capabilities of a specified FileExportService, for example, the supported file sharing protocol and the supported extrinsic methods.

1. From the *SNIA_FileExportService*, traverse *CIM_ElementCapabilities* to *SNIA_FileExportCapabilities* to get the instance defining supported capabilities of the specified *SNIA_FileExportService*.
2. The property – *FileSharingProtocol* specifies the sharing protocol (NFS/CIFS), the property – *SupportedSynchronousMethods* and *SupportedAsynchronousMethods* specify the extrinsic methods and their mode (Sync/Async).

Use case: Create Settings for File Share

ExportedFileShareSetting is required when creating file share via SMI-S to specify caller-proposed aspects of the share.

Refer to [Use Case: Create a File Share](#)

Use case: Create a File Share

This use case describes how to create a file share exported with a set of caller-proposed settings.

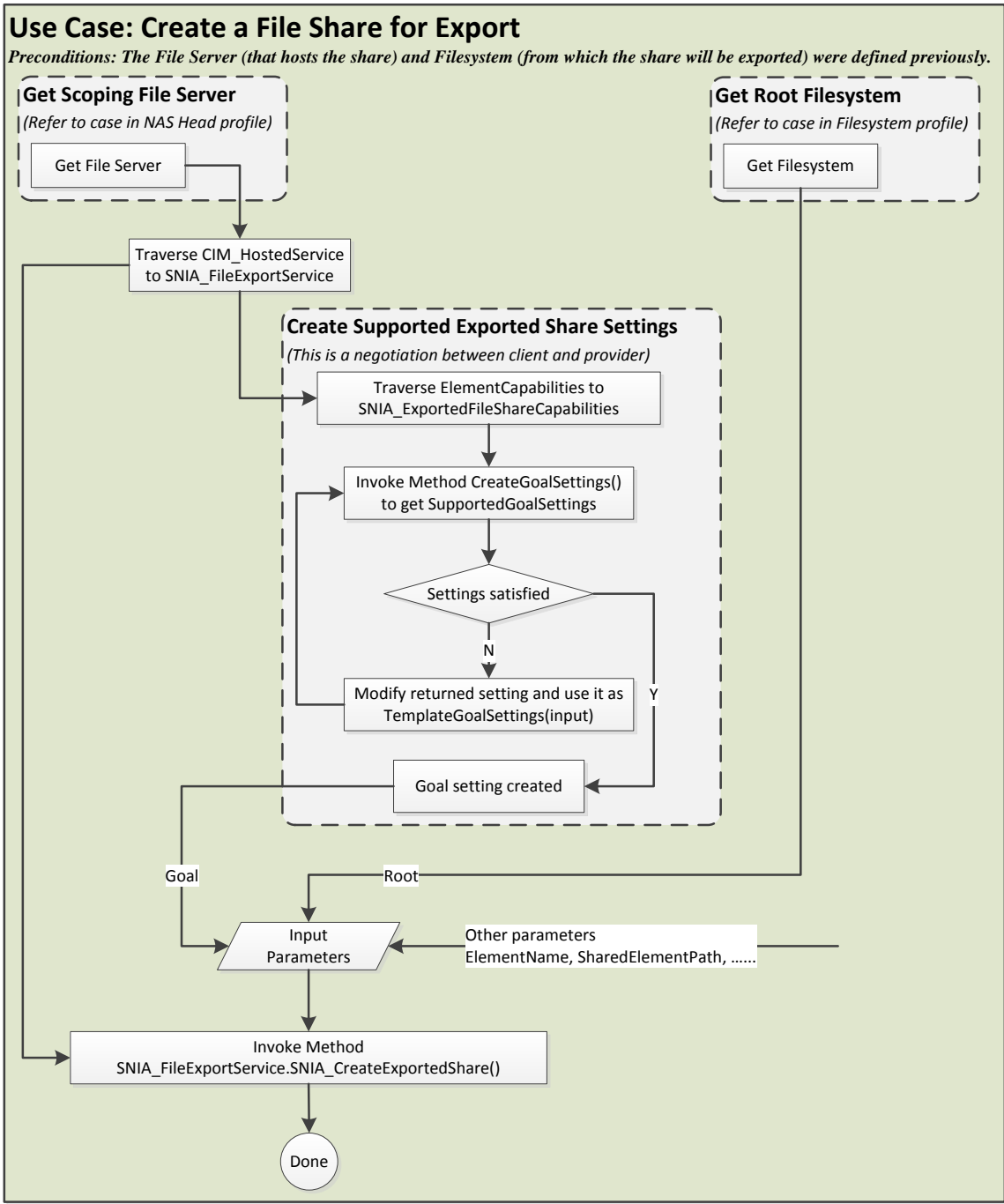


Figure 54 - Flowchart of Creating Exported File Share

Use case: Modify the Settings of a CIFS File Share

This use case shows how to modify aspects of an exported files share.

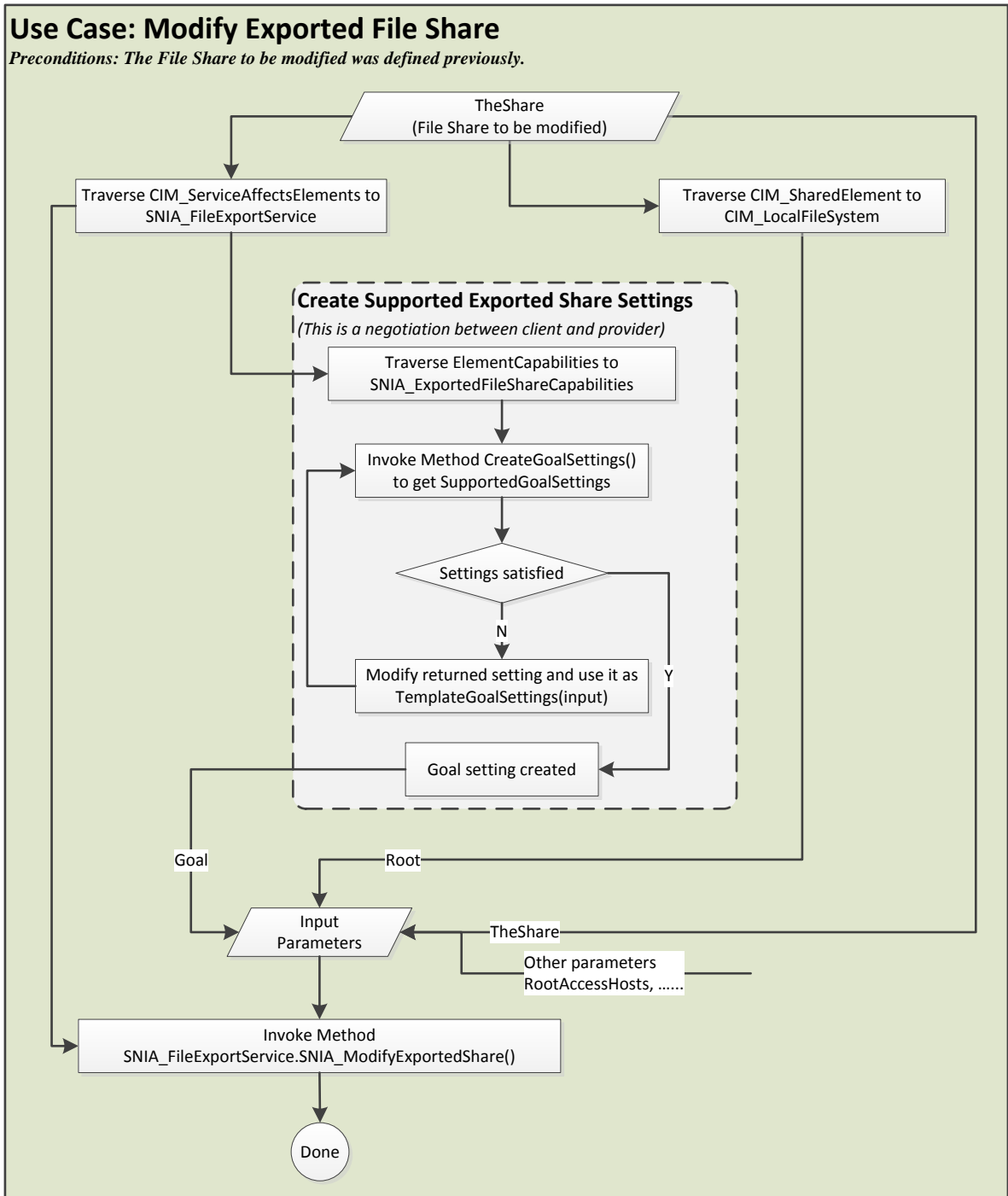


Figure 55 - Flowchart of Modifying the Settings of a CIFS File Share

Use case: Release an Exported File Share

This use case describes how to release an exported file share.

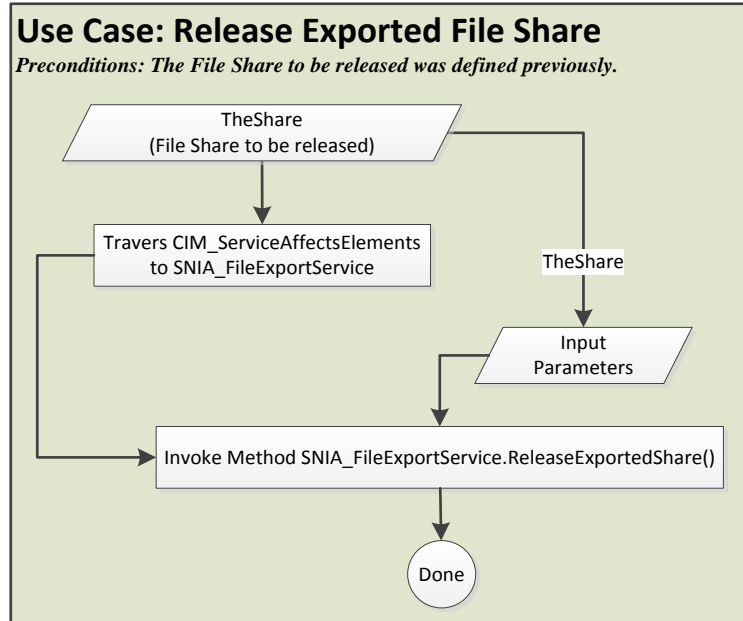


Figure 56 - Flowchart of Releasing an Exported File Share

CIM Elements

The implemented classes and associations related to the File Export Manipulation Subprofile in the VNXe storage system are described as follows:

Table 494 - CIM Elements for File Export Manipulation Subprofile

CIM Class	Implemented Class	Description
CIM_ElementCapabilities	EMC_VNXe_FileExportService_CIFSShareCapabilities_ElementCapabilitiesAssocLeaf	Represents the association between the File Export Service and the ExportedFileShareCapabilities element that indicates that support is available for managing an exported FileShare for CIFS protocol.
CIM_ElementCapabilities	EMC_VNXe_FileExportService_FileExportCapabilities_ElementCapabilitiesAssocLeaf	Represents the association between the File Export Service and the FileExportCapabilities element that describes the service capabilities.
CIM_ElementCapabilities	EMC_VNXe_FileExportService_NFSShareCapabilities_ElementCapabilitiesAssocLeaf	Represents the association between the File Export Service and the ExportedFileShareCapabilities element that indicates that support is available for managing an exported FileShare for NFS protocol.
CIM_HostedService	EMC_VNXe_CIFSServer_FileExportService_HostedServiceAssocLeaf	Represents the association between the File Export Service and its hosting CIFS File Server Computer System.
CIM_HostedService	EMC_VNXe_NFSServer_FileExportService_HostedServiceAssoc	Represents the association between the File Export Service and its hosting NFS File

CIM Class	Implemented Class	Description
	Leaf	Server Computer System.
CIM_ServiceAffectsElement	EMC_VNXe_FileExportService_CIFSShare_ServiceAffectsElementAssocLeaf	Represents the association between the File Export Service and the elements that the service manages via NFS protocol (such as a FileShare configured for exporting a LogicalFile).
CIM_ServiceAffectsElement	EMC_VNXe_FileExportService_NFSShare_ServiceAffectsElementAssocLeaf	Represents the association between the File Export Service and the elements that the service manages via NFS protocol (such as a FileShare configured for exporting a LogicalFile).
SNIA_ExportedFileShareCapabilities	EMC_VNXe_CIFSShareCapabilitiesLeaf	Represents the Capabilities of the File Export Service for managing FileShares of CIFS protocol.
SNIA_ExportedFileShareCapabilities	EMC_VNXe_NFSShareCapabilitiesLeaf	Represents the Capabilities of the File Export Service for managing FileShares of NFS protocol.
SNIA_ExportedFileShareSetting	EMC_VNXe_CIFSShareSettingLeaf	Represents the setting for an exported CIFS share.
SNIA_ExportedFileShareSetting	EMC_VNXe_NFSShareSettingLeaf	Represents the setting for an exported NFS share.
SNIA_FileExportCapabilities	EMC_VNXe_FileExportCapabilitiesLeaf	Represents the management capabilities of the File Export Service.
SNIA_FileExportService	EMC_VNXe_FileExportServiceLeaf	Represents the File Export Service provides the methods to create and export file elements as shares.
SNIA_SettingsDefineCapabilities	EMC_VNXe_CIFSShareCapabilities_CIFSShareSetting_SettingsDefineCapabilitiesAssocLeaf	Represents the association between a CIFS ExportedFileShareCapabilities and a predefined ExportedFileShareSetting element that specifies what the Capabilities can support.
SNIA_SettingsDefineCapabilities	EMC_VNXe_NFSShareCapabilities_NFSShareSetting_SettingsDefineCapabilitiesAssocLeaf	Represents the association between a NFS ExportedFileShareCapabilities and a predefined ExportedFileShareSetting element that specifies what the Capabilities can support.

EMC_VNXe_FileExportServiceLeaf

Table 495 - Referenced properties/methods for EMC_VNXe_FileExportServiceLeaf

CIM property	Description/notes
SystemCreationClassName	Set to 'EMC_VNXe_CIFSServerLeaf' or 'EMC_VNXe_NFSServerLeaf'
SystemName	Name of the FileServer that hosts the service.

CIM property	Description/notes
CreationClassName	Set to 'ENC_VNXe_FileExportServiceLeaf'
Name	Same as SystemName
ElementName	Set as SystemName + ':File Export Service'

EMC_VNXe_FileExportCapabilitiesLeaf

Table 496 - Referenced properties/methods for EMC_VNXe_FileExportCapabilitiesLeaf

CIM property	Description/notes
InstanceID	Unique ID of this instance.
ElementName	Set as InstanceID + ':File Export Capabilities'.
FileSharingProtocol	Sharing protocol supported by the FileServer.
CASupported	Indicates whether CA is supported.
SupportedAsynchronousMethod	Set to empty array.
SupportedSynchronousMethod	Set to [2: CreateExportedShare, 3: ModifyExportedShare, 4: ReleaseExportedShare, 5: CreateGoal]

EMC_VNXe_CIFSShareCapabilitiesLeaf

Table 497 - Referenced properties/methods for EMC_VNXe_CIFSShareCapabilitiesLeaf

CIM property	Description/notes
InstanceID	Unique ID of this instance.
FileSharingProtocol	Set to {3: CIFS}
CASupported	Indicates whether CA is supported.

EMC_VNXe_NFSShareCapabilitiesLeaf

Table 498 - Referenced properties/methods for EMC_VNXe_NFSShareCapabilitiesLeaf

CIM property	Description/notes
InstanceID	Unique ID of this instance.
FileSharingProtocol	Set to {2: CIFS}

EMC_VNXe_CIFSShareSettingLeaf

For further information, refer to [EMC_VNXe_CIFSShareSettingLeaf](#) in File Export Profile.

[EMC_VNXe_NFSShareSettingLeaf](#)

For further information, refer to [EMC_VNXe_NFSShareSettingLeaf](#) in File Export Profile.

[EMC_VNXe_CIFSServer_FileExportService_HostedServiceAssocLeaf](#)

**Table 499 - Referenced properties/methods for
EMC_VNXe_CIFSServer_FileExportService_HostedServiceAssocLeaf**

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_CIFSServerLeaf
Dependent	Reference of EMC_VNXe_FileExportServiceLeaf

[EMC_VNXe_NFSServer_FileExportService_HostedServiceAssocLeaf](#)

**Table 500 - Referenced properties/methods for
EMC_VNXe_NFSServer_FileExportService_HostedServiceAssocLeaf**

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_NFSServerLeaf
Dependent	Reference of EMC_VNXe_FileExportServiceLeaf

[EMC_VNXe_FileExportService_FileExportCapabilities_ElementCapabilitiesAssocLeaf](#)

**Table 501 - Referenced properties/methods for
EMC_VNXe_FileExportService_FileExportCapabilities_ElementCapabilitiesAssocLeaf**

CIM property	Description/notes
ManagedElement	Reference of EMC_VNXe_FileExportServiceLeaf
Capabilities	Reference of EMC_VNXe_FileExportCapabilitiesLeaf

[EMC_VNXe_FileExportService_CIFSShareCapabilities_ElementCapabilitiesAssocLeaf](#)

**Table 502 - Referenced properties/methods for
EMC_VNXe_FileExportService_CIFSShareCapabilities_ElementCapabilitiesAssocLeaf**

CIM property	Description/notes
ManagedElement	Reference of EMC_VNXe_FileExportServiceLeaf
Capabilities	Reference of EMC_VNXe_CIFSShareCapabilitiesLeaf

[EMC_VNXe_FileExportService_NFSShareCapabilities_ElementCapabilitiesAssocLeaf](#)

**Table 503 - Referenced properties/methods for
EMC_VNXe_FileExportService_NFSShareCapabilities_ElementCapabilitiesAssocLeaf**

CIM property	Description/notes
--------------	-------------------

CIM property	Description/notes
ManagedElement	Reference of EMC_VNXe_FileExportServiceLeaf
Capabilities	Reference of EMC_VNXe_NFSShareCapabilitiesLeaf

EMC_VNXe_FileExportService_CIFSShare_ServiceAffectsElementAssocLeaf

**Table 504 - Referenced properties/methods for
EMC_VNXe_FileExportService_CIFSShare_ServiceAffectsElementAssocLeaf**

CIM property	Description/notes
AffectingElement	Reference of EMC_VNXe_FileExportServiceLeaf
AffectedElement	Reference of EMC_VNXe_CIFSShareLeaf

EMC_VNXe_FileExportService_NFSShare_ServiceAffectsElementAssocLeaf

**Table 505 - Referenced properties/methods for
EMC_VNXe_FileExportService_NFSShare_ServiceAffectsElementAssocLeaf**

CIM property	Description/notes
AffectingElement	Reference of EMC_VNXe_FileExportServiceLeaf
AffectedElement	Reference of EMC_VNXe_NFSShareLeaf

EMC_VNXe_CIFSShareCapabilities_CIFSShareSetting_SettingsDefineCapabilitiesAssocLeaf

**Table 506 - Referenced properties/methods for
EMC_VNXe_CIFSShareCapabilities_CIFSShareSetting_SettingsDefineCapabilitiesAssocLeaf**

CIM property	Description/notes
GroupComponent	Reference of EMC_VNXe_CIFSShareCapabilitiesLeaf
PartComponent	Reference of EMC_VNXe_CIFSShareSettingLeaf

EMC_VNXe_NFSShareCapabilities_NFSShareSetting_SettingsDefineCapabilitiesAssocLeaf

**Table 507 - Referenced properties/methods for
EMC_VNXe_NFSShareCapabilities_NFSShareSetting_SettingsDefineCapabilitiesAssocLeaf**

CIM property	Description/notes
GroupComponent	Reference of EMC_VNXe_NFSShareCapabilitiesLeaf
PartComponent	Reference of EMC_VNXe_NFSShareSettingLeaf

Simple Identity Management Profile

Overview

The Simple Identity Management Profile provides the ability to represent a principal with a User ID that has been authenticated through third-party authentication, and to manage account privileges for the exported CIFS File Share.

Class diagram

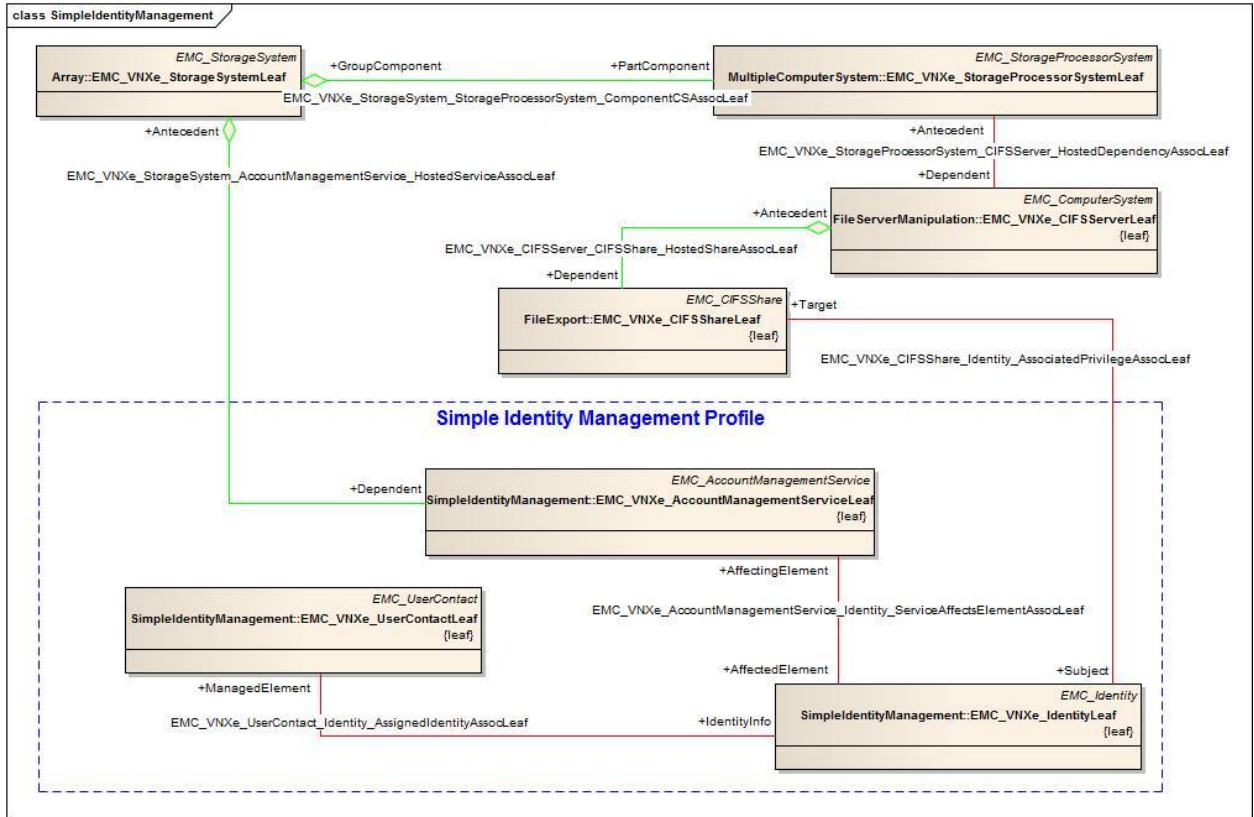


Figure 57 - Simple Identity Management Instance diagram

Method of the Profile

Intrinsic/Extrinsic Methods on UserContact Manipulation

Method: List all UserContacts

- **Description**

Implemented by intrinsic method *EnumerateInstances* of *EMC_VNXe_UserContactLeaf*.

NOTE: EMC recommends that users do not directly invoke *EnumerateInstances* to list all existing UserContacts managed in SMI-S. The result of this intrinsic method may be out-of-date.

Alternatively, a user should invoke extrinsic method *EMC_VNXe_AccountManagementServiceLeaf.getUserContact* to retrieve UserContact for a specific domain user.

Method: Create UserContact

- Description**

Implemented by *EMC_VNXe_AccountManagementServiceLeaf.CreateUserContact*.

This method allows a user to create a UserContact in SMI-S.

NOTE: A UserContact can be created in SMI-S if the following conditions are satisfied:

- 1) The UserContact has not been created yet. Client cannot create a same UserContact twice.
- 2) The corresponding account exists in Active Directory.
- 3) The communication between VNXe and Active Directory is active.

- Parameters**

Table 508 - Signature and parameters of EMC_VNXe_AccountManagementServiceLeaf.CreateUserContact

Parameter	Qualifiers	Type	Description
System	IN, Required	CIM_ComputerSystemRef	The scoping system in which the UserContact is managed. This is the top-level ComputerSystem.
UserContactTemplate	IN, Embedded Instance, Required	CIM_UserContactRef	An embedded instance of <i>EMC_VNXe_UserContactLeaf</i> containing domain and user information for creation.
UserContact	OUT	CIM_UserContactRef	Reference of the newly created UserContact.
Identities	OUT	CIM_IdentityRef[]	Array of references of Identity associated to the newly created UserContact.

- Return Results**

Table 509 - Possible return code of EMC_VNXe_AccountManagementServiceLeaf.CreateUserContact

Return code	Type	Description
0	uint32	Success
4	uint32	Failed: <ul style="list-style-type: none"> • Invalid input parameters • Required UserContact has already been created in SMI-S • Account does not exist in Active Directory

Return code	Type	Description
		<ul style="list-style-type: none"> Any error occurs when calling underlying component
NA	CIM_Error	Exceptions will be thrown when any of the below cases is met: <ul style="list-style-type: none"> Any of the required parameters is NULL Any of input parameters is not of the right CIM/SNIA type defined in the MOF files Any of the input references cannot be found in the system

Method: Retrieve UserContact

• **Description**

Implemented by *EMC_VNXe_AccountManagementServiceLeaf.GetUserContact*.

This method allows a user to retrieve a created UserContact in SMI-S.

NOTE: A UserContact can be retrieved in SMI-S if the following conditions are satisfied:

- 1) The UserContact has been created via SMI-S.
- 2) The corresponding account exists in Active Directory.
- 3) The communication between VNXe and Active Directory is active.

• **Parameters**

Table 510 - Signature and parameters of EMC_VNXe_AccountManagementServiceLeaf.GetUserContact

Parameter	Qualifiers	Type	Description
UserID	IN, Required	String	String including the domain and user name for retrieving UserContact. It should be in the format: "Domain\UserName".
UserContact	OUT	CIM_UserContactRef	Reference of the target UserContact..

• **Return Results**

Table 511 - Possible return code of EMC_VNXe_AccountManagementServiceLeaf.GetUserContact

Return code	Type	Description
0	uint32	Success
4	uint32	Failed: <ul style="list-style-type: none"> Invalid input parameters Required UserContact has not been created in SMI-S

Return code	Type	Description
		<ul style="list-style-type: none"> Account does not exist in Active Directory Any error occurs when calling underlying component.
NA	CIM_Error	Exceptions will be thrown when any of the below cases is met: <ul style="list-style-type: none"> Any of the required parameters is NULL Any of input parameters is not of the right CIM/SNIA type defined in the MOF files Any of the input references cannot be found in the system

Method: Remove UserContact

- Description**

Implemented by intrinsic method *DeleteInstance* of *EMC_VNXe_UserContactLeaf*.

Intrinsic/Extrinsic Methods on Privilege Manipulation

Method: List all AssociatedPrivileges

- Description**

Implemented by intrinsic method *EnumerateInstances* of *EMC_VNXe_CIFSShare_Identity_AssociatedPrivilegeAssocLeaf*.

NOTE: EMC recommends that users do not directly invoke the intrinsic method *EnumerateInstances* of *EMC_VNXe_CIFSShare_Identity_AssociatedPrivilegeAssocLeaf* to list all existing ACEs. Method: List all AssociatedPrivileges on a CIFS Share

- Description**

Implemented by intrinsic method *References* of *EMC_VNXe_CIFSShare_Identity_AssociatedPrivilegeAssocLeaf* for *EMC_VNXe_CIFSSharLeaf*.

NOTE: An ACE will be exposed to the SMI-S client if the following conditions are satisfied:

- 1) Permission of the ACE has been granted rather than denied. SMI-S only displays granted ACEs.
- 2) The UserContact associated with the Identity has been created in SMI-S. SMI-S only displays ACEs for users registered in SMI-S.
- 3) Account whose SID is represented by the Identity exists in Active Directory.
- 4) The communication between VNXe and Active Directory is active.

Method: Assign Privilege

- Description**

Implemented by *EMC_VNXe_FileExportServiceLeaf.AssignPrivilegeToExportedShare*.

This method allows a user to assign privilege for a domain user on a CIFS share.

NOTE: Privilege can be assigned successfully if the following conditions are satisfied:

- 1) Accounts whose SIDs are represented by the Identities exist in Active Directory.
- 2) Value of Activities is valid for VNXe SMI-S Provider (see [Activities Mapping](#)).
- 3) Privileges that have been assigned should NOT exist. Client cannot assign the privilege for an identity to a CIFS Share twice.
- 4) The communication between VNXe for File and Active Directory is active.

- **Parameters**

**Table 512 - Signature and parameters of
EMC_VNXe_FileExportServiceLeaf.AssignPrivilegeOnExportedShare**

Parameter	Qualifiers	Type	Description
Identities	IN, Required	CIM_IdentityRef[]	Array of Identities for which the privilege will be granted.
FileShare	IN, Required	CIM_FileShareRef	CIFS Share to which the privilege will be applied.
Activities	IN, Required	Uint16	Array of activities to be granted for the input Identities to the input FileShare.

- **Return Results**

**Table 513 - Possible return code of
EMC_VNXe_FileExportServiceLeaf.AssignPrivilegeOnExportedShare**

Return code	Type	Description
0	uint32	Success
1	uint32	Not Supported: <ul style="list-style-type: none"> • File server that hosts the FileExportService is not a CIFS server.
2	uint32	Failed: <ul style="list-style-type: none"> • File server that hosts the FileExportService is not the one hosting the FileShare. • FileShare is not a CIFS share. • Any identity is repeated in Identities. • Any identity has already been assigned privilege on the share • Any error occurs when calling underlying component
3	uint32	Activities Not Supported: <ul style="list-style-type: none"> • Activities is invalid (see Activities Mapping)

Return code	Type	Description
4	uint32	Identity Not Found: <ul style="list-style-type: none"> Identities array is empty. One or more identities cannot be located in system.
5	uint32	File Share Not Found: <ul style="list-style-type: none"> CIFS share cannot be located in the system.
NA	CIM_Error	Exceptions will be thrown when any of the below cases is met: <ul style="list-style-type: none"> Any of the required parameters is NULL. Any of input parameters is not of the right CIM/SNIA type defined in the MOF files. Any of the input references cannot be found in the system.

Method: Delete Privilege

- Description**

Implemented by intrinsic method *DeleteInstance* of *EMC_VNXe_CIFSShare_Identity_AssociatedPrivilegeAssocLeaf*.

Method: Modify Privilege

- Description**

Implemented by intrinsic method *ModifyInstance* of *EMC_VNXe_CIFSShare_Identity_AssociatedPrivilegeAssocLeaf*.

NOTE: Only property *Activities* is allowed to be modified, and the values should follow the convention defined in [Activities Mapping](#).

Client consideration

Model Specification

- EMC_VNXe_UserContactLeaf

Class *EMC_VNXe_UserContactLeaf* represents account information about an identity that has been authenticated through third-party authentication (Active Directory in VNXe).

NOTE: Creation/deletion of a UserContact via SMI-S Provider does **NOT literally add/remove** the corresponding account in the Active Directory. This operation only enables the share access privilege of the account to be manageable or unmanageable by the SMI-S Provider.

- EMC_VNXe_IdentityLeaf

Class *EMC_VNXe_IdentityLeaf* represents the SID of an account on Active Directory.

NOTE: Client **CANNOT** manipulate Identity directly via SMI-S Provider. Identity can only be affected by UserContact manipulation.

- EMC_VNXe_CIFSShare_Identity_AssociatedPrivilegeAssocLeaf
 Since VNXe applied ACL (Access Control List) for CIFS share access, class *EMC_VNXe_CIFSShare_Identity_AssociatedPrivilegeAssocLeaf* represents the ACE (Access Control Entry) on a pair of Identity and CIFS File Share.

Table 514 - Description of properties in the AssociatedPrivilege

Property	Description
Subject	Identity for which the privilege has been granted or denied.
Target	CIFS share to which the privilege has been applied.
Activities	Activities granted.
PrivilegeGranted	Whether the privilege is granted or denied.

NOTE: Property *PrivilegeGranted* is always set to TRUE as VNXe SMI-S Provider **ONLY** supports privilege granting and only exposes granted ACEs.

Mapping between Activities and Privileges in ACE can be described as follows:

Table 515 - Mapping between Activities and Privileges

Activities Set By Client	Activities Shown In SMI-S	Privileges To Share
Client input contains {5:Read, 6:Write, 14: Authorize to Grant/Deny Authorization}.	{2: Create, 3: Delete, 5, 6, 7:Execute, 14}	FULLCONTROL
Client input contains {5, 6} but does not contain {14}.	{2, 3, 5, 6, 7}	WRITE
Client input contains {5} but does not contain {6, 14}.	{5, 7}	READ
Client input contains none of the values in {5, 6, 14}	ERROR, Not Supported, Invalid	ERROR, Not Supported, Invalid

Use case: Create UserContact and Identities for a domain user

This use case describes how to create UserContact and Identities in SMI-S for a domain user on Active Directory.

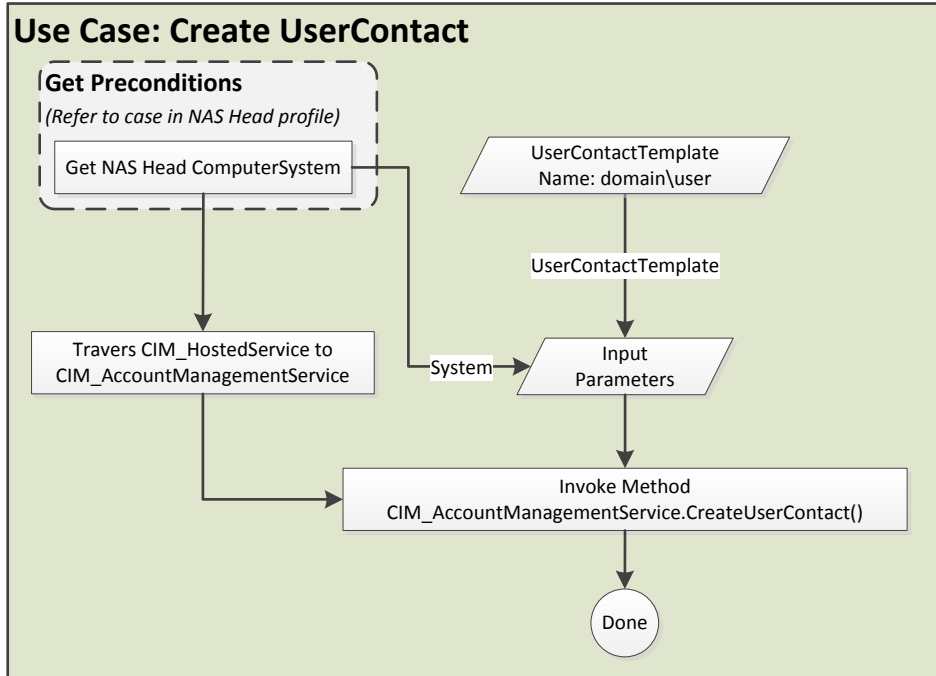


Figure 58 - Flowchart of creating UserContact

Use case: Get the UserContact for a specified domain user whose UserContact has been created

This use case describes how to get the UserContact for a specified domain user who's UserContact has been created in the system.

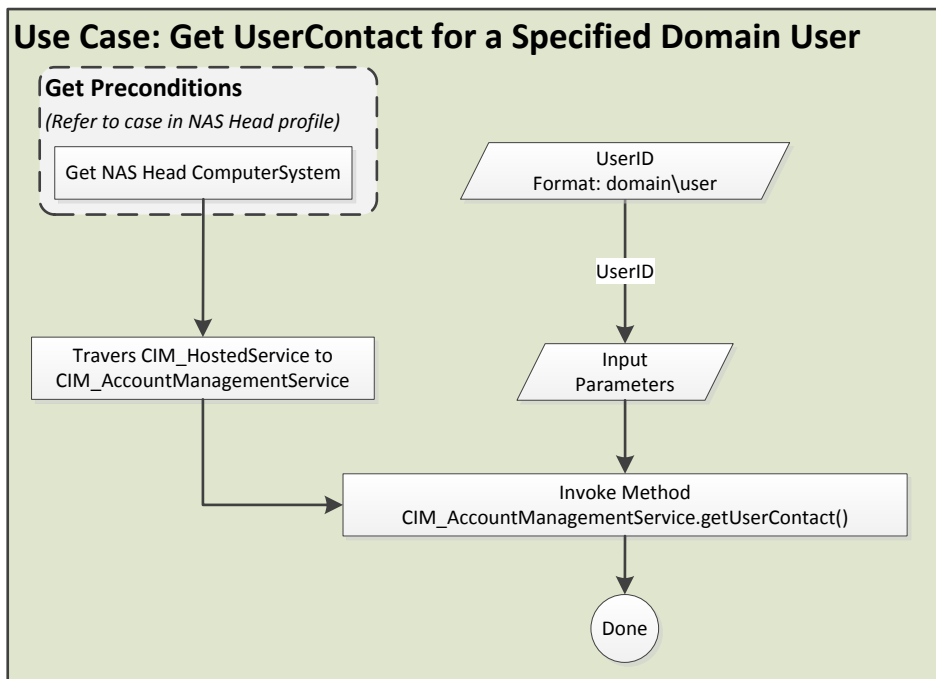


Figure 59 - Flowchart of getting UserContact

Use case: Removal of an existing UserContact

Client can invoke intrinsic method *DeleteInstance* on *CIM_UserContact* to remove the target instance of *UserContact*.

Use case: Get the Privilege and UserContact by an Exported File Share

This use case describes how to get the Privilege and UserContact by an Exported FileShare.

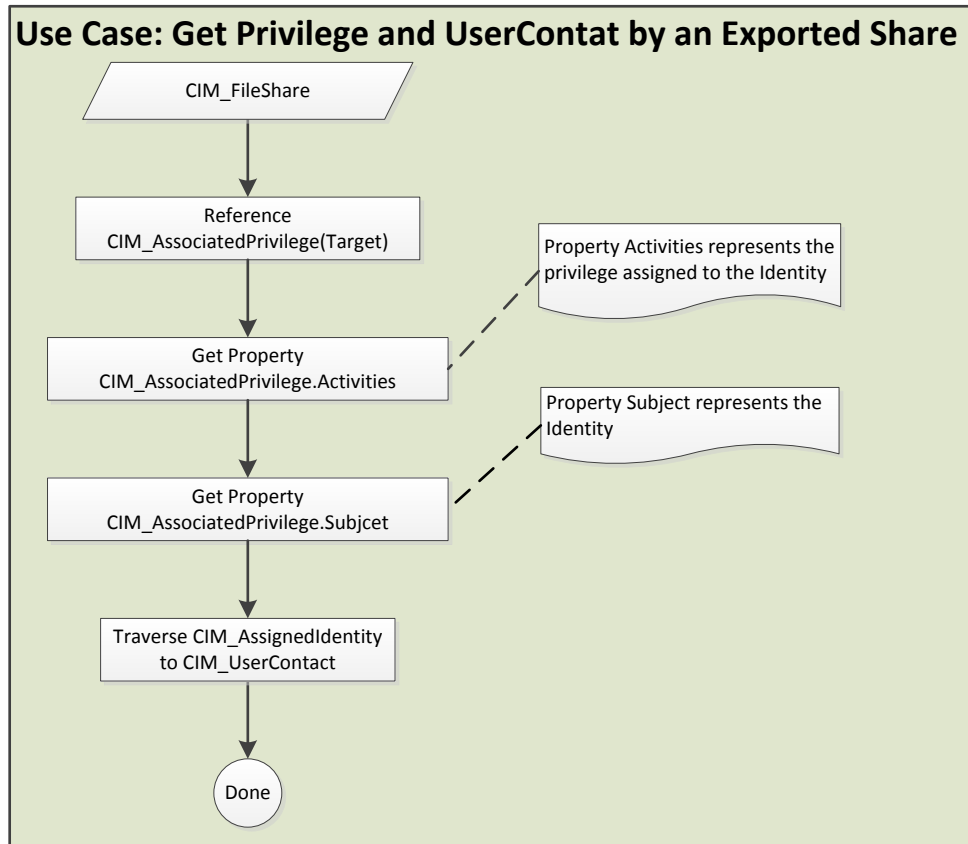


Figure 60 - Flowchart of getting Privilege and UserContact by an Exported FileShare

Use case: Assign a privilege to a pair of UserContact and Exported FileShare

This use case describes how to assign Privilege to a pair of UserContact and Exported FileShare.

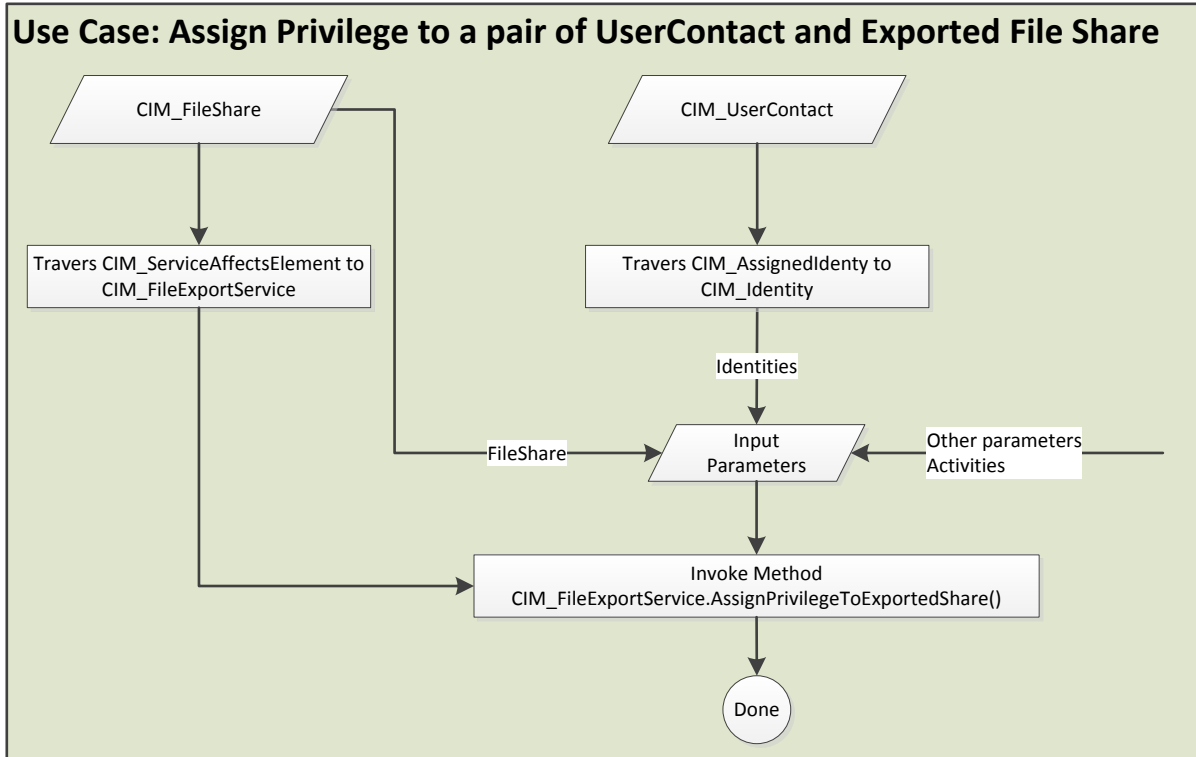


Figure 61 - Flowchart of assigning privilege to Exported FileShare

Use case: Modify privilege to a pair of UserContact and Exported FileShare

The client can invoke intrinsic method *ModifyInstance* on *CIM_AssociatedPrivilege* (associates the FileShare and the Identity of the UserContact) to change the granted activities.

Use case: Removal of the existing privilege

The client can invoke intrinsic method *DeleteInstance* on *CIM_AssociatedPrivilege* to remove the target instance of *AssociatedPrivilege*.

CIM Element

The implemented classes and associations related to the Simple Identity Management Profile in VNXe Storage System can be described as follows:

Table 516 - CIM Elements implemented in VNXe for Simple Identity Management Profile

CIM Class	Implemented Class	Description
CIM_AccountManagementService	EMC_VNXe_AccountManagementServiceLeaf	Provides methods for UserContact management.
CIM_AssignedIdentity	EMC_VNXe_UserContact_Identity_AssignedIdentityAssocLeaf	Associates the UserContact with the Identity.
CIM_AssociatedPrivilege	EMC_VNXe_CIFSShare_Identity_AssociatedPrivilegeAss	Associates the Identity with the CIFS FileShare and provides the privilege (activities) granted to the Identity on the

	ocLeaf	Share.
CIM_HostedService	EMC_VNXe_StorageSystem_AccountManagementService_HostedServiceAssocLeaf	Associates the Account Management Service with its scoping system (Top level Computer System).
CIM_Identity	EMC_VNXe_IdentityLeaf	Identity to identify a domain user in Active Directory.
CIM_ServiceAffectsElement	EMC_VNXe_AccountManagementService_Identity_ServiceAffectsElementAssocLeaf	Associates the Identity with the Account Management Service.
CIM_UserContact	EMC_VNXe_UserContactLeaf	UserContact for a domain user in Active Directory.

EMC_VNXe_AccountManagementServiceLeaf

Table 517 - Referenced properties/methods for EMC_VNXe_AccountManagementServiceLeaf

CIM property	Description/notes
SystemCreationClassName	CreationClassName of the scoping system.
SystemName	Name of the scoping system.
CreationClassName	Name of the class used to create this instance.
Name	Name of the instance.
CreateUserContact()	Method for creating UserContact and Identities.
GetUserContact()	Method for getting UserContact.

EMC_VNXe_UserContactLeaf

Table 518 - Referenced properties/methods for EMC_VNXe_UserContactLeaf

CIM property	Description/notes
CreationClassName	Name of the class used to create this instance.
Name	Label by which the object is known.
UserID	User ID of a domain user.

EMC_VNXe_IdentityLeaf

Table 519 - Referenced properties/methods for EMC_VNXe_IdentityLeaf

CIM property	Description/notes
InstanceID	Unique identifier for this instance.
ElementName	SID of a domain user.

EMC_VNXe_StorageSystem_AccountManagementService_HostedServiceAssocLeaf

**Table 520 - Referenced properties/methods for
EMC_VNXe_StorageSystem_AccountManagementService_HostedServiceAssocLeaf**

CIM property	Description/notes
Antecedent	Reference of EMC_VNXe_StorageSystem
Dependent	Reference of EMC_VNXe_AccountManagementServiceLeaf

EMC_VNXe_AccountManagementService_Identity_ServiceAffectsElementAssocLeaf

**Table 521 - Referenced properties/methods for
EMC_VNXe_AccountManagementService_Identity_ServiceAffectsElementAssocLeaf**

CIM property	Description/notes
AffectedElement	Reference of EMC_VNXe_IdentityLeaf
AffectingElement	Reference of EMC_VNXe_AccountManagementServiceLeaf

EMC_VNXe_UserContact_Identity_AssignedIdentityAssocLeaf

**Table 522 - Referenced properties/methods for
EMC_VNXe_UserContact_Identity_AssignedIdentityAssocLeaf**

CIM property	Description/notes
IdentityInfo	Reference of EMC_VNXe_IdentityLeaf
ManagedElement	Reference of EMC_VNXe_UserContactLeaf

EMC_VNXe_CIFSShare_Identity_AssociatedPrivilegeAssocLeaf

**Table 523 - Referenced properties/methods for
EMC_VNXe_CIFSShare_Identity_AssociatedPrivilegeAssocLeaf**

CIM property	Description/notes
Subject	Reference of EMC_VNXe_IdentityLeaf
Target	Reference of EMC_VNXe_CIFSShareLeaf
UseKey	Used to distinguish instances in case multiple instances of this association exist between the same Subject and Target. Set to Default
PrivilegeGranted	Indicates whether the Privilege is granted or denied. Set to TRUE.
Activities	Activities granted.

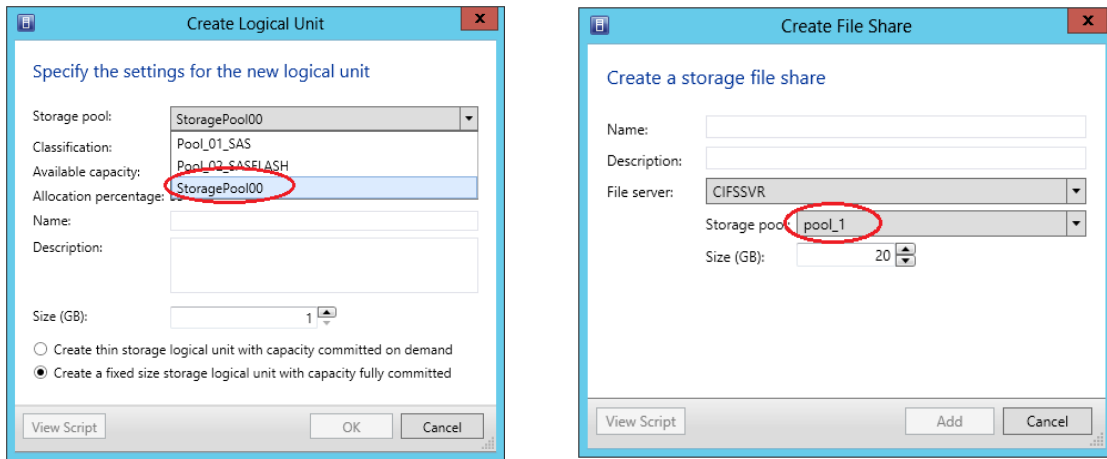
Appendix A: Known Issues in Third Party Integration

Microsoft System Center 2012 Virtual Machine Manager (SCVMM)

SCVMM is fully integrated with the VNXe array to configure VM storage through the SMI-S API.

Display Name of Pool Selection

There is an inconsistency issue in displaying VNXe storage pools in different wizards of SCVMM. As shown as Figure 56, the Create Logical Unit wizard (Figure 56.a) shows user-friendly names of available pools in the pool selection combo-box. However, the Create File Share wizard (Figure 56.b) shows a normalized format of *pool_<seq_num>*.



(a) Pool selection of LUN creation

(b) Pool selection of SHARE creation

Figure 62 - Storage Pool Selection in Different Wizards

These are two different ways of displaying storage pools. In the Create File Share wizard, SCVMM uses the property *CIM_StoragePool.InstanceID* as the pool display name. In the Create Logical Unit wizard, SCVMM uses the property *CIM_StoragePool.ElementName* as the pool display name. In Figure 56, *pool_1* and *StoragePool00* actually represent the same storage pool in the array.

To map the pool InstanceID to ElementName, execute the array command */stor/config/pool* to list detailed information on the pool:

Table 524 - Array Command for Pool Information Display

Example: <pre>uemcli -d 192.168.2.2 -u admin -p Password123! /stor/config/pool show</pre>
Output <pre>Storage system address: 192.168.2.2 Storage system port: 443 HTTPS connection 1: ID = pool_1 Name = StoragePool00 Total space = 4373823356928 (3.9T) Remaining space = 2632846934016 (2.3T)</pre>

Subscription percent = 39%
Number of drives = 10
RAID level = 5
Stripe length = 5
Rebalancing = no
Health state = OK (5)
FAST Cache enabled = no
Protection size used = 0

Certificate Import

When a user adds a VNXe array in SCVMM for the first time, an Import Certificate dialog displays indicating that the array's self-signed security certificate must be imported. After clicking Import, the array's self-signed certificate is saved in the Windows certificate store. After importing, SCVMM never checks again whether the certificate has changed or is missing, always using this original certificate in communication with the backend array.

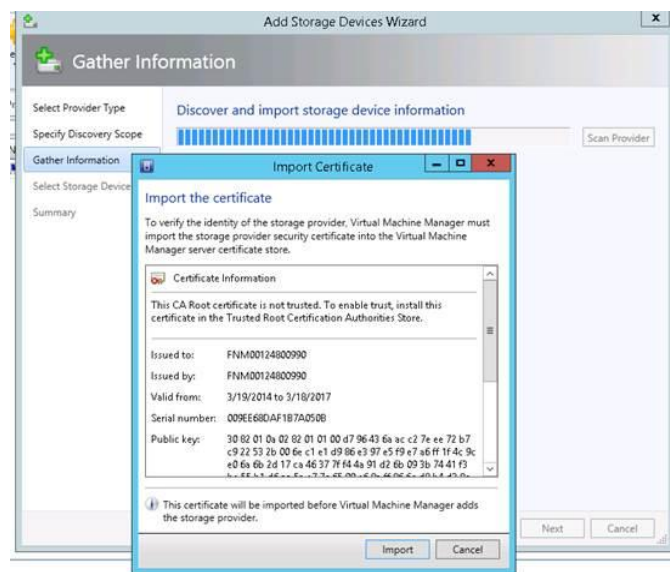


Figure 63 - Import Certificate Dialog

However, as designed, the self-signed certificate is re-generated after an upgrade or refresh of the VNXe array. Since SCVMM does not check for certificate change or loss, an upgrade or refresh will cause all SMI-S requests to fail and SCVMM will report that the VNXe array is not responding.

NOTE:

After a VNXe array upgrade or any other operation that could result in a certificate change (for example, removing the certificate from the Windows certificate store), the user must import the new certificate into the Windows certificate store to bring SCVMM back to normal operation.