Resource Management with CIM for Linux

--- draft ---

Base
File System & Volume
Process & Service

March 2002

author :
Linux Technology Center (LTC)
Systems Management
Heidi Neumann
heidineu@de.ibm.com

Version 1.4
November 2002

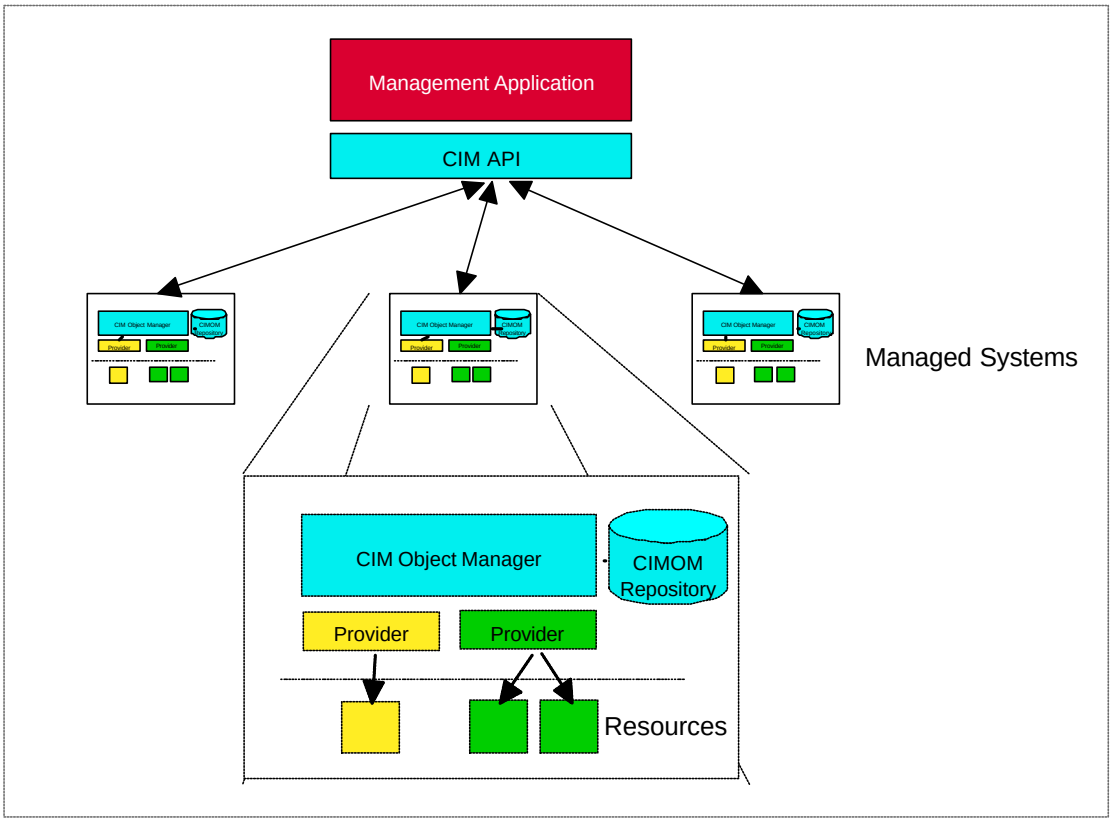
1 Introduction

CIM is an implementation-neutral schema to offer overall management information in a network / enterprise environment and accepted industry standard defined by the Distributed Management Task Force (DMTF).

The reader of this document should be familiar with CIM.

1.1 Architecture

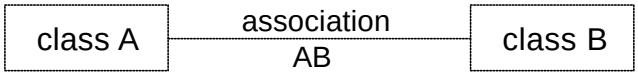
The architecture behind CIM shows picture 1.1.



1.1 Architecture

1.2 CIM Meta Schema

The schemata described in this paper are based on CIM schema version 2.6. The current selection, we identified as important base instrumentation, is only a small subset of possible classes. For the implementation of higher-level services, a well defined and fundamental base instrumentation is absolute necessary. The base instrumentation is available on each single CIM enabled system (Managed System).



1.2 CIM Meta Schema

Classes present entities of logical and physical objects in a Managed System. Associations reflect relations between the instances of classes. An association has a left and a right 'end'. These ends are 'pointers' to classes. When an association is called, one end is specified (that means one instance of the referred class is specified). The task of the provider is to figure out, which instance(s) of the opposite end stay in relation to this specific instance. Schema and instrumentation become valuable with a well defined set of associations between the classes.

For example, a Managed System has two disks available. Each disk is configured with 3 partitions. The disks itself are presented by two instances of a class called StorageVolume. The 6 partitions are presented by 6 instances of a class called DiskPartition. Only the association DiskPartitionBasedOnVolume shows which partition is based on which volume. If the association is called from the StorageVolume end, the provider returns these three instances of DiskPartition, which are based on the specified volume. If the association is called from the DiskPartition end, the provider returns the one instance of StorageVolume presenting the disk.

1.3 Provider Interface

1.3.1 Instance Provider Interface

enumInstances()
enumInstanceNames()
getInstance()
setInstance()
createInstance()
deleteInstance()
execQuery()

1.3.2 Associator Provider Interface

associators()
associatorNames()
references()
referenceNames()

1.3.3 Method Provider Interface

invokeMethod()

1.3.4 Event Provider Interface

tbd ...

1.4 How to read the following chapters

1.4.1 Headings

The style “Linux_ClassName : CIM_ClassName” refers to the mof notification used in CIM. The first name is the name of the new sub-class. The “:” can be translates as “is derived from”. The name after the “:” is the name of the CIM class. Each heading in this style identifies a Linux sub-class.

1.4.2 Property Table Description

The following table skeletons are used to lists the properties of classes, methods and associations. The heading of the next chapters describes the type and name of the provider. This style is used in the document to identify the type and name of the corresponding provider in the SBLIM instrumentation packages.

1.4.2.1 Class Provider : Linux_ClassName

The first column “Name” contains the Name of the CIM Property. The next column “Type” contains the type of the property and “(key)” if the Property acts as Key Property. The column “CIM class” contains the name of the CIM class (without the prefix “CIM”) where this property is derived from. The column “Value Description” contains the value in “ ” or a short description. An “x” in the “Impl.” column means, this Property

is implemented by the corresponding SBLIM provider. If the Property is currently not implemented but planned to be implemented the field contains an "'x (-)" combination. A "-" means will not be supported.

Name	Type	CIM class	Value / Description	Impl.
CreationClassName	string (key)	System	"Linux_UnitaryComputerSystem"	x
…				

1.4.2.2 Method Provider : Linux_ClassNameMethodProvider

See description of chapter 1.4.2.1 "Class Provider : Linux_ClassName".

Name	Type	CIM class	Value / Description	Impl.
Reboot()	uint32	OperatingSystem	reboot the system	-
…				

1.4.2.3 Association Provider : Linux_AssociationName

The first column "Reference A" describes the "left" end of the association and column "Reference B" the "right" end of the association. Both columns contain the type of the reference in the association, e.g. "Antecendet" / "Dependent" and the value of the reference. In general the value is a Linux sub-class, but there are cases, where a more generic association to CIM classes makes sense.

Reference A	Reference B
ReferenceType = "ClassName"	ReferenceType = "ClassName"

1.4.3 Instance Table Description

The next tables refer to example scenarios. They contain all instances, which exists in the specified scenario. The instances are managed by the corresponding provider. In general one provider is responsible for one class. If a CIM class has several Linux sub-classes with similar meanings, like in the case of UnixLocalFileSystem (chapter 3.1.1.1), it is possible to write one provider which is responsible for all subclasses.

1.4.3.1 Instance Table for Class Provider

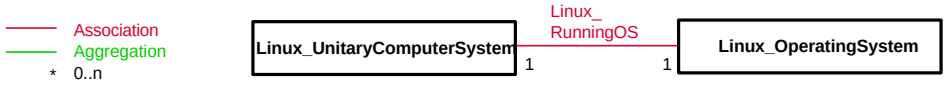
Linux Class (Property CreationClassName)	Key Property = "Value"
Linux_Ext2FileSystem	Name = "/dev/hda1"
…	

1.4.3.2 Instance Table for Association Provider

specified end	returned instance(s)
Linux_UnitaryComputerSystem "host.domain"	Linux_Ext2FileSystem "/dev/hda1" Linux_ReiserFileSystem "/dev/hda2"
Linux_Ext2FileSystem "/dev/hda1"	Linux_UnitaryComputerSystem "host.domain"
…	…
Linux_ClassName	- (NOT SUPPORTED)
Linux_ClassName "value"	-

2 Base Management

The necessary CIM schema parts for Base Management are Core and System. Appendix A “Base Management” shows the CIM hierarchy and the parents of the classes / associations.



2.1 Base Management

2.1 Class description

2.1.1 Linux_UnitaryComputerSystem : CIM_UnitaryComputerSystem

... represents a single node Computer System, the Managed System (MS). A single MS delivers one instance of the class Linux_UnitaryComputerSystem to represent itself. This instance acts as 'container' for all parts of the base instrumentation. Other classes reference within their keys to Linux_UnitaryComputerSystem.

Class Provider : Linux_ UnitaryComputerSystem

Name	Type	CIM class	Value / Description	Impl.
Caption	string	ManagedElement	short caption to describe the class	x
Description	string	ManagedElement	short description of this class	x
Name	string (key)	ManagedSystemElement	full qualified hostname “host.domain”	x
Status	string	ManagedSystemElement	current Status of the System : “OK”	x
InstallDate	datetime	ManagedSystemElement	Install date of the running OS	x (-)
CreationClassName	string (key)	System	“Linux_UnitaryComputerSystem”	x
NameFormat	string	System	indicates the heuristic used for the creation of ‘Name’ : contains the value “IP”	x
PrimaryOwnerName	string	System	name of the primary system owner	-
PrimaryOwnerContact	string	System	contact (e.g. e-mail) of the primary owner	-
Roles[]	string	System	specifies the role of this system in the IT-environment	-
Dedicated[]	uint16	ComputerSystem	dedicated to a special purpose : “0” (not dedicated)	x (-)
OtherIdentifyingInfo[]	string	ComputerSystem	additional data behind the system name	-
IdentifyingDescriptions[]	string	ComputerSystem	explanations and details behind ‘OtherIdentifyingInfo’	-
PowerManagementCapabilities[]	uint16	ComputerSystem	specific power related capabilities of the system	-
ResetCapability	uint16	ComputerSystem	hardware’s reset capability	-
SetPowerState()	uint32	ComputerSystem	method to define the desired power state	-
InitialLoadInfo[]	string	UnitaryComputerSystem	to find the initial load device or boot service	-
LastLoadInfo	string	UnitaryComputerSystem	contains the data of the initial load device or boot service	-
PowerManagementSupported	boolean	UnitaryComputerSystem	system and OS support power management	-
PowerState	uint16	UnitaryComputerSystem	current power state	-
WakeUpType	uint16	UnitaryComputerSystem	event that causes the system to power up	-

2.1.2 Linux_OperatingSystem : CIM_OperatingSystem

... is software/firmware that makes the ComputerSystem's hardware usable, and implements and / or manages the resources, file systems, processes, user interfaces, services, ... available on the ComputerSystem. The class Linux_OperatingSystem represents the OS Linux, independent of the Distribution specifics (SuSE, RedHat, Caldera, Mandrake ...). According to the CIM definition the provider should return as many instances as OperatingSystems are installed or loaded on the ComputerSystem, while an OperatingSystem is “installed” when placed on one of the ComputerSystem's StorageExtents. This definition expects from the provider programmer to search for possible OS installations on each disk and verify at least some of the capabilities.

The SBLIM provider return the instance of the current running Operating System.

Class Provider : Linux_ OperatingSystem

Name	Type	CIM class	Value / Description	Impl.
Caption	string	ManagedElement	short caption to describe the class	x
Description	string	ManagedElement	short description of this class	x
Name	string (key)	ManagedSystemElement	Linux and the release number, e.g. “Linux 2.4.10”	x
Status	string	ManagedSystemElement	current Status of the OS : “OK”	x
InstallDate	datetime	ManagedSystemElement	Install date of the running OS	x (-)
CreationClassName	string (key)	OperatingSystem	“Linux_OperatingSystem”	x
CSCreationClassName	string (key)	OperatingSystem	“Linux_UnitaryComputerSystem”	x
CSName	string (key)	OperatingSystem	“host.domain”	x
Version	string	OperatingSystem	version of the Linux OS instance	x
OSType	uint16	OperatingSystem	“36” indicating Linux	x
OtherTypeDescription	string	OperatingSystem	“LINUX”	x
LocalDateTime	datetime	OperatingSystem	OS notion of local date and time	x (-)
CurrentTimeZone	sint16	OperatingSystem	indicating the number of minutes the OS is away from Greenwich Mean Time (GMT)	x (-)
LastBootUpTime	datetime	OperatingSystem	time when the OS was last booted	x (-)
Distributed	boolean	OperatingSystem	“FALSE”	x
NumberOfUsers	uint32	OperatingSystem	number of current user sessions	-
NumberOfLicensedUsers	uint32	OperatingSystem	User license for the OS; if unlimited “0	-
NumberOfProcesses	uint32	OperatingSystem	number of processes currently running on the OS	x (-)
MaxNumberOfProcesses	uint32	OperatingSystem	max number of processes the OS can support	x (-)
MaxProcessesPerUser	uint32	OperatingSystem	max number of processes a user can be associated with	-
MaxProcessMemorySize	uint64	OperatingSystem	max kBytes of memory a process can allocate	x (-)
FreePhysicalMemory	uint64	OperatingSystem	kBytes of Physical Memory currently available	x (-)
FreeSpaceInPagingFiles	uint64	OperatingSystem	max kBytes in Paging Space without swapping	-
FreeVirtualMemory	uint64	OperatingSystem	kBytes of Virtual Memory currently available	x (-)
SizeStoredInPagingFiles	uint64	OperatingSystem	max kBytes that can be stored in OS Paging Space	-
TotalSwapSpaceSize	uint64	OperatingSystem	total kBytes of Swap Space : “0”	x
TotalVirtualMemorySize	uint64	OperatingSystem	total kBytes of Virtual Memory	x (-)
TotalVisibleMemorySize	uint64	OperatingSystem	total amount in kBytes of physical memory	x (-)

Method Provider : Linux_ OperatingSystemMethodProvider

Name	Type	CIM class	Value / Description	Impl.
Reboot()	uint32	OperatingSystem	reboot the system	-
ShutDown()	uint32	OperatingSystem	method to shut down the system	-

2.2 Association Description

2.2.1 Linux_RunningOS : CIM_RunningOS

... an association to indicate the currently executing instance of OperatingSystem on the ComputerSystem. This association is an one-to-one association. Each of the classes Linux_OperatingSystem and Linux_UnitaryComputerSystem return a single instance.

Association Provider : Linux_RunningOS

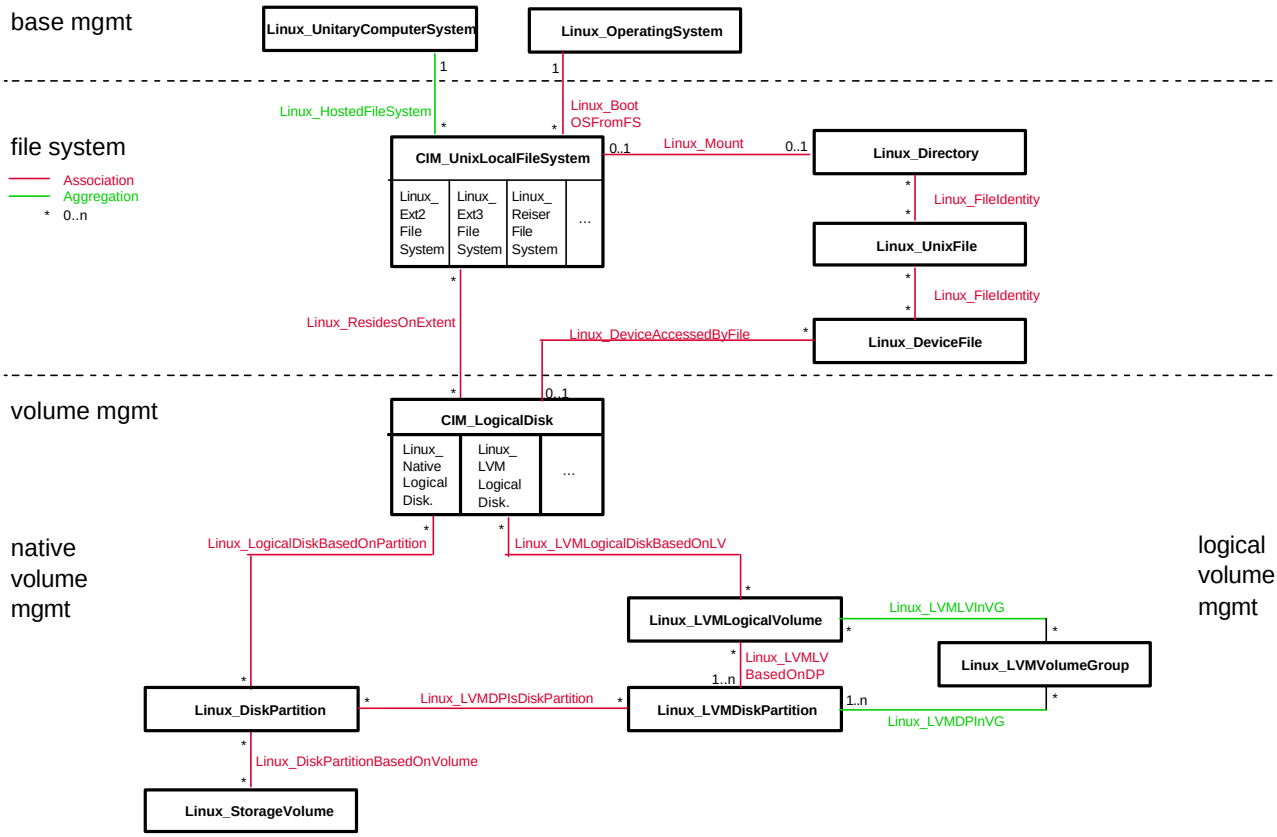
Reference A	Reference B
Antecedent = “Linux_OperatingSystem”	Dependent = “Linux_UnitaryComputerSystem”

Instance Table

specified end	returned instance(s)
Linux_UnitaryComputerSystem “host.domain”	Linux_OperatingSystem “Linux 2.4.10”
Linux_OperatingSystem “Linux 2.4.10”	Linux_UnitaryComputerSystem “host.domain”

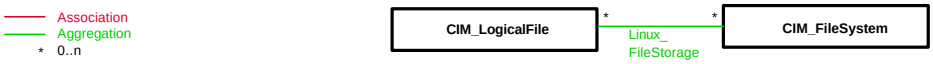
3 File System & Volume Management

The necessary CIM schema parts for File System & Storage Management are Core, Unix, System and Device. Appendix B “File System & Storage Management” shows the CIM hierarchy and the parents of the classes / associations.



3.1 File System & Volume Management

Picture 3.2 shows the relation between any type of LogicalFile (subclasses are Directory and DeviceFile) and any type of FileSystem (local and remote ones).



3.2 Relation between LogicalFile(s) and the FileSystem(s)

An example configuration of a System (Managed Node) will help to describe the behavior of the class and association provider. The System owns the following resources. These resources will be described by the schema above.

- one disk : had (ide0)
  - with 5 partitions : hda1 , hda2 , hda3 , hda4 , hda5

partition name	partition type	file system type	mount point
• hda1	native	ext2	/
• hda2	native	reiserfs	/mnt
• hda3	LVM		
• hda4	LVM		
• hda5	LVM		

•LVM Configuration

• vg1	contains	hda3 and hda4		
• vg2	contains	hda5		
• vg1	contains	lv1	ext2	not mounted
• lv1	is located on	hda3 and hda4		

• output of command "`ls -r /`"

- ⇨ bin
- ⇨ dev
  - ⇨ hda
  - ⇨ hda1
  - ⇨ hda2
  - ⇨ hda3
  - ⇨ hda4
  - ⇨ hda5
- ⇨ home
  - ⇨ user
- ⇨ mnt
- ⇨ usr

• output of command "`ls -r /mnt/`"

- ⇨ dir1
- ⇨ dir2

3.1 File System Management

3.1.1 Class description

3.1.1.1 CIM_UnixLocalFileSystem

... represents the Unix environment view of file store controlled by a ComputerSystem through local means (e.g. , direct device driver access). The file store is managed directly by the ComputerSystem without the need for another computer to act as a file server.

To differentiate the several types of FileSystem that are supported by Linux, each file system type is represented by a separate sub-class. An enumeration of CIM_UnixLocalFileSystem returns all local File Systems. The files /etc/fstab (contains all local file systems) and /proc/mounts or /etc/mtab (both contain all currently mounted file systems) are taken as sources. The entries are compared to avoid duplicates and get all entries.

Class Provider : Linux_LocalFileSystem

Name	Type	CIM class	Value / Description	Impl.
Caption	string	ManagedElement	short caption to describe the class	x
Description	string	ManagedElement	short description of this class	x
Name	string (key)	ManagedSystemElement	full qualified path of the LogicalDisk, the fs resides on (Property "DeviceID" of LogicalDisk), e.g. "/dev/hda2"	x
Status	string	ManagedSystemElement	current Status of the System : "OK"	x
InstallDate	datetime	ManagedSystemElement		-
CSCreationClassName	string (key)	FileSystem	"Linux_UnitaryComputerSystem"	x
CSName	string (key)	FileSystem	full qualified hostname "host.domain"	x
CreationClassName	string (key)	FileSystem	"Linux_xxxFileSystem"	x
Root	string	FileSystem	Pathname defining the root of the filesystem, e.g. "/"	x
BlockSize	uint64	FileSystem	Filesystem's block size for data storage and retrieval	x
AvailableSpace	uint64	FileSystem	total amount of free space in bytes	x
FileSystemSize	uint64	FileSystem	Total size in bytes	x
FileSystemType	string	FileSystem	type	x
MaxFileNameLength	uint32	FileSystem	max. length of file names	x
ReadOnly	boolean	FileSystem	Filesystem is read only	x
CaseSensitive	boolean	FileSystem	case sensitive file names are supported	-
CasePreserved	boolean	FileSystem	case of the file names are preserved	-
EncryptionMethod	string	FileSystem	Algorithm / tool used to encrypt the filesystem	-

CompressionMethod	string	FileSystem	Algorithm / tool used to compress the filesystem	-
CodeSet[]	uint16	FileSystem	character sets / encoding supported by the filesystem	-
ClusterSize	uint32	FileSystem	minimum file allocation size	x (-)
FreeInodes	uint64	UnixLocalFileSystem	Number of free Inodes	x (-)
TotalInodes	uint64	UnixLocalFileSystem	total number of Inodes	x (-)
FSReservedCapacity	uint64	UnixLocalFileSystem	Reserve data capacity of the filesystem in bytes	x (-)

Chapter 3.5 “File System Table” describes a solution how additional file system entries, which are not listed in /etc/fstab or /etc/mtab can be made known to the system via CIM.

The provider Linux_LocalFileSystem is responsible for all sub-classes of the class CIM_UnixLocalFileSystem. The following table contains all instances returned by the Linux_LocalFileSystem provider for the example environment described on page 7 and 8.

Linux Class (Property CreationClassName)	Key Property = “Value”
Linux_Ext2FileSystem	Name = “/dev/hda1”
Linux_ReiserFileSystem	Name = “/dev/hda2”
Linux_Ext2FileSystem	Name = “/dev/vg1/lv1”

3.1.1.1.1 Linux_Ext2FileSystem : CIM_UnixLocalFileSystem

... represents the file store of type 'ext2' locally controlled by a ComputerSystem.

Class Provider : Linux_LocalFileSystem

3.1.1.1.2 Linux_Ext3FileSystem : CIM_UnixLocalFileSystem

... represents the file store of type 'ext3' locally controlled by a ComputerSystem.

Class Provider : Linux_LocalFileSystem

3.1.1.1.3 Linux_ReiserFileSystem : CIM_UnixLocalFileSystem

... represents the file store of type 'reiserfs' locally controlled by a ComputerSystem.

Class Provider : Linux_LocalFileSystem

3.1.1.1.4 Linux_ . . . FileSystem : CIM_UnixLocalFileSystem

... there are more types of FileSystem that need to be supported in future.

Class Provider : Linux_LocalFileSystem

3.1.1.2 CIM_LogicalFile

... represents a LogicalDevice (Linux_DeviceFile) or a Directory (Linux_Directory). It is located within the context of a FileSystem. See Appendix B “File System & Storage Management” page 2 for the class hierarchy.

To get exact information about the LogicalFiles, it is absolutely necessary to exactly specify the path. In correspondence to the scenario (page 7 & 8), the '/' FileSystem is of type “ext2” and a “reiserfs” is mounted at “/mnt”. It is a very small but crucial difference between the mount point “/mnt” and the directory path ”/mnt/”. The first one - “/mnt” - defines the directory, which acts as mount point. This directory is logically located within the Linux_Ext2FileSystem. The value of the key FSCreationClassName is “Linux_Ext2FileSystem”. The second one “/mnt/” defines the root of the mounted FileSystem, and is logically located within the Linux_ReiserFileSystem. The value of the key FSCreationClassName is “Linux_ReiserFileSystem”. Keep this information in mind !

3.1.1.2.1 Linux_Directory : CIM_UnixDirectory

... is a type of File that logically groups UnixFiles 'contained' in it. Because of the high amount of returned instances (each directory entry of each FileSystem !) and no known requirement of this general information, an enumeration of this class is not supported. The only supported provider API call is getInstance(), where the full qualified path of the directory acts as key.

Class Provider : Linux_Directory

Name	Type	CIM class	Value / Description	Impl.
Caption	string	ManagedElement	short caption to describe the class	x
Description	string	ManagedElement	short description of this class	x
Name	string (key)	ManagedSystemElement	full qualified path of the directory, e.g. “/usr”	x
Status	string	ManagedSystemElement	current Status of the System : “OK”	x
InstallDate	datetime	ManagedSystemElement		-
CSCreationClassName	string (key)	LogicalFile	“Linux_UnitaryComputerSystem”	x
CSName	string (key)	LogicalFile	full qualified hostname “host.domain”	x
FSCreationClassName	string (key)	LogicalFile	“Linux_xxxFileSystem” or “Linux_NFS”	x
FSName	string (key)	LogicalFile	Value of Property Name in class Linux_xxxFileSystem or Linux_NFS, e.g. “/dev/hda2”	x
CreationClassName	string (key)	LogicalFile	“Linux_Directory”	x
CreationDate	datetime	LogicalFile	the date of file’s creation	x (-)
LastAccessed	datetime	LogicalFile	last access time of the logical file	x (-)
LastModified	datetime	LogicalFile	last time the logical file was modified	x (-)
Readable	boolean	LogicalFile	Logical file is redable	x (-)
Writable	boolean	LogicalFile	Logical file is writable	x (-)
Executable	boolean	LogicalFile	Logical file is executable	x (-)
FileSize	uint64	LogicalFile	Size of the logical file in bytes	x (-)
InUseCount	uint64	LogicalFile	Indicating the ‘file open’ that are currently active against the logical file	x (-)
EncryptionMethod	string	LogicalFile	Algorithm / tool used to encrypt the logical file	-
CompressionMethod	string	LogicalFile	Algorithm / tool used to compress the logical file	-
FileSizeBits	uint64	UnixDirectory	minimum number of bits needed to present the maximum size of a unix file allowed in the directory	-

3.1.1.2.2 Linux_DeviceFile : CIM_UnixDeviceFile

... is a special type of LogicalFile that represents a Device. This convention is useful for operating systems that manage devices using a byte stream I/O model. The LogicalDevice that is associated with this file is specified using the DeviceAccessedByFile relationship.

The provider lists all entries of the /dev directory as instances.

Class Provider : Linux_DeviceFile

Name	Type	CIM class	Value / Description	Impl.
Caption	string	ManagedElement	short caption to describe the class	x
Description	string	ManagedElement	short description of this class	x
Name	string (key)	ManagedSystemElement	full qualified path of the device file in “/dev/…”	x
Status	string	ManagedSystemElement	current Status of the System : “OK”	x
InstallDate	datetime	ManagedSystemElement		-
CSCreationClassName	string (key)	LogicalFile	“Linux_UnitaryComputerSystem”	x
CSName	string (key)	LogicalFile	full qualified hostname “host.domain”	x
FSCreationClassName	string (key)	LogicalFile	“Linux_xxxFileSystem” or “Linux_NFS”	x
FSName	string (key)	LogicalFile	Value of Property Name in class Linux_xxxFileSystem or Linux_NFS, e.g. “/dev/hda2”	x
CreationClassName	string (key)	LogicalFile	“Linux_Directory”	x
CreationDate	datetime	LogicalFile	the date of file’s creation	x (-)
LastAccessed	datetime	LogicalFile	last access time of the logical file	x (-)
LastModified	datetime	LogicalFile	last time the logical file was modified	x (-)
Readable	boolean	LogicalFile	Logical file is redable	x (-)
Writable	boolean	LogicalFile	Logical file is writable	x (-)
Executable	boolean	LogicalFile	Logical file is executable	x (-)
FileSize	uint64	LogicalFile	Size of the logical file in bytes	x (-)
InUseCount	uint64	LogicalFile	Indicating the ‘file open’ that are currently active against the logical file	x (-)
EncryptionMethod	string	LogicalFile	Algorithm / tool used to encrypt the logical file	-
CompressionMethod	string	LogicalFile	Algorithm / tool used to compress the logical file	-

DeviceDescription	string	UnixDeviceFile	additional information provided by the driver	-
DeviceID	string	UnixDeviceFile	Device ID	x (-)
DeviceMajor	string	UnixDeviceFile	Device Major ID	x (-)
DeviceMinor	string	UnixDeviceFile	Device Minor ID	x (-)
DeviceFileType	string	UnixDeviceFile	Type of the device file	x (-)
OtherTypeDescription	string	UnixDeviceFile	Additional description if type is “Other	x (-)

The following table contains all instances returned by the Linux_DeviceFile provider for the example environment described at beginning of this chapter.

Linux Class (Property CreationClassName)	Key Property = “Value”
Linux_DeviceFile	Name = “/dev/hda”
Linux_DeviceFile	Name = “/dev/hda1”
Linux_DeviceFile	Name = “/dev/hda2”
Linux_DeviceFile	Name = “/dev/hda3”
Linux_DeviceFile	Name = “/dev/hda4”
Linux_DeviceFile	Name = “/dev/hda5”

3.1.1.3 Linux_UnixFile : CIM_UnixFile

... hosts the properties of the different types of LogicalFiles in a Unix environment. It is only associated with the various sub-classes of CIM_LogicalFile.

An enumeration of this class is not supported because of the high amount of returned instances on a single system. The only supported provider API call is getInstance(), where the value of the Property 'Name' acts as key.

Class Provider : Linux_UnixFile

Name	Type	CIM class	Value / Description	Impl.
Caption	string	ManagedElement	short caption to describe the class	x
Description	string	ManagedElement	short description of this class	x
Name	string	ManagedSystemElement	free-form string to describe the instance	x
Status	string	ManagedSystemElement	current Status of the System : “OK”	x
InstallDate	datetime	ManagedSystemElement		-
CSCreationClassName	string (key)	UnixFile	“Linux_UnitaryComputerSystem”	x
CSName	string (key)	UnixFile	full qualified hostname “host.domain”	x
FSCreationClassName	string (key)	UnixFile	“Linux_xxxFileSystem” or “Linux_NFS”	x
FSName	string (key)	UnixFile	Value of Property Name in class Linux_xxxFileSystem or Linux_NFS, e.g. “/dev/hda2”	x
LFCreationClassName	string (key)	UnixFile	“Linux_UnixFile”	x
LFName	string (key)	UnixFile	full qualified path to the instance of LogicalFile	x
UserID	string	UnixFile	UID of the file owner	x (-)
GroupID	string	UnixFile	GID of the file owner	x (-)
FileInodeNumber	string	UnixFile	Inode number of the file	x (-)
LastModifiedInode	datetime	UnixFile	Time the Inode was last modified	x (-)
LinkCount	uint64	UnixFile	Number of names for this file	x (-)
LinkMax	uint64	UnixFile	Max. number of links to a single file	x (-)
NameMax	uint64	UnixFile	Max. number of bytes in a filename	x (-)
PathMax	uint64	UnixFile	Max. number of bytes in a pathname	x (-)
SaveText	boolean	UnixFile	Restricted deletion (sticky bit)	x (-)
SetGid	boolean	UnixFile	file has setgid permission	x (-)
SetUid	boolean	UnixFile	file has setuid permission	x (-)
PosixAsyncIo	uint64	UnixFile	async. input / output operations	x (-)
PosixChownRestricted	uint64	UnixFile	chown() is restricted to processes with appropriate privileges	x (-)
PosixNoTrunc	uint64	UnixFile	pathname components longer than NameMax generate an error	x (-)
PosixPrioIo	uint64	UnixFile	prioritized input / output operations	
PosixSyncIo	uint64	UnixFile	synchronized input / output operations	x (-)

Example :

The Client has an instance of the class Linux_Directory and an instance of the class Linux_DeviceFile.

Linux Class (Property CreationClassName)	Key Property = “Value”
Linux_Directory	Name = “/home/user”
Linux_DeviceFile	Name = “/dev/hda2”

The provider of the class Linux_UnixFile offers information about these instances with the following key values. The association Linux_FileIdentity is responsible to access these instances.

Linux Class (Property CreationClassName)	Key Property = "Value"
Linux_UnixFile	LFName = "/home/user"
Linux_UnixFile	LFName = "/dev/hda2"
...	...

3.1.1.4 CIM_LogicalDisk

... is a presentation of a contiguous range of logical blocks that is identifiable by a FileSystem via the Disk's DeviceID field. In a Linux environment, the key property DeviceID contains the full qualified access path to the disk, e.g. '/dev/hda7'. LogicalDisks are typically built on a DiskPartition or a StorageVolume. However, it can be based on other StorageExtents, like CIM_Memory, in the case of a RAM disk. That depends on the modeled environment.

A DiskPartition or StorageVolume becomes a LogicalDisk, when a file system was created on it (command "mkfs"). Indirectly the class LogicalDisk distinguishes between native and LVM type partitions. It is not possible to create a file system on a DiskPartition of type 'Linux LVM (8e)'. In this case the file system is created on the logical volume build on top of the LVM DiskPartition(s). See chapter 3.3 "Native Volume Management" and 3.4 "Logical Volume Management" for more information.

The class LogicalDisk can be interpreted as an interface between the File System Management and the Volume Management. The Volume Management directly depends on the system's hardware, while the File System Management depends on the OperatingSystem. File System and Volume Management are independent of each other. Additional Volume Management Concepts can be modeled without changing the File System part.

3.1.2 Association description

3.1.2.1 Linux_HostedFileSystem : CIM_HostedFileSystem

... is an aggregation between the ComputerSystem and the total amount of FileSystem(s) (local and remote) hosted on this node. This association can be compared with the output of the "df" / "mount" command in Linux.

Association Provider : Linux_HostedFileSystem

Reference A	Reference B
GroupComponent = "Linux_UnitaryComputerSystem"	PartComponent = "CIM_FileSystem"

Instance Table

specified end	returned instance(s)
Linux_UnitaryComputerSystem "host.domain"	Linux_Ext2FileSystem "/dev/hda1" Linux_ReiserFileSystem "/dev/hda2" Linux_Ext2FileSystem "/dev/vg1/lv1"
Linux_Ext2FileSystem "/dev/hda1"	Linux_UnitaryComputerSystem "host.domain"
Linux_ReiserFileSystem "/dev/hda2"	Linux_UnitaryComputerSystem "host.domain"
Linux_Ext2FileSystem "/dev/vg1/lv1"	Linux_UnitaryComputerSystem "host.domain"

3.1.2.2 Linux_BootOSFromFS : CIM_BootOSFromFS

... is the association between the OperatingSystem and the FileSystem(s) from which the OperatingSystem is loaded. The association is one-to-many since Linux can depend on several FileSystems ("/" and "/boot/") in order to correctly and completely load.

Association Provider : Linux_BootOSFromFS

Reference A	Reference B
Antecedent = “CIM_UnixLocalFileSystem”	Dependent = “Linux_OperatingSystem”

Instance Table

specified end	returned instance(s)
Linux_OperatingSystem “Linux 2.4.10”	Linux_Ext2FileSystem “/dev/hda1”
Linux_Ext2FileSystem “/dev/hda1”	Linux_OperatingSystem “Linux 2.4.10”
Linux_ReiserFileSystem “/dev/hda2”	-
Linux_Ext2FileSystem “/dev/vg1/lv1”	-

3.1.2.3 Linux_Mount : CIM_Mount

... is an association between a FileSystem (local and remote ones) and a Directory, which indicates that the Directory is being attached to the FileSystem. The semantics of this relationship require that the mounted Directory be contained by a FileSystem (via the FileStorage association) that is different from the FileSystem referenced as the Dependent.

Linux_Mount represents the currently mounted file systems (local and remote ones). The support is one directional – from FileSystem to Directory. It is possible to figure out which instance of Directory acts as mount point of the specified instance of FileSystem. The provider returns no instance, if the FileSystem instance is currently not mounted.

Remember ! To get exact information about the Directory, it is absolutely necessary to exactly specify the path. It is a very small but crucial difference between the mount point “/mnt” and the directory path ”/mnt/”. The first one - “/mnt” - defines the directory, which acts as mount point. This directory is logically located within the Linux_Ext2FileSystem. The value of the key FSCreationClassName is “Linux_Ext2FileSystem”. The second one “/mnt/” defines the root of the mounted FileSystem, and is logically located within the Linux_ReiserFileSystem. The value of the key FSCreationClassName is “Linux_ReiserFileSystem”.

Association Provider : Linux_Mount

Reference A	Reference B
Antecedent = “Linux_Directory”	Dependent = “CIM_FileSystem”

Instance Table

specified end	returned instance(s)
Linux_Directory “?”	- (NOT SUPPORTED)
Linux_Ext2FileSystem “/dev/hda1”	Linux_Directory “/”
Linux_ReiserFileSystem “/dev/hda2”	Linux_Directory “/mnt”
Linux_Ext2FileSystem “/dev/vg1/lv1”	-

3.1.2.4 Linux_DeviceAccessedByFile : CIM_ DeviceAccessedByFile

... specifies the LogicalDevice that is associated with and accessed by using the referenced DeviceFile.

The association provider support is one directional - from LogicalDevice to DeviceFile. It is possible to figure out which instance of DeviceFile is responsible for a specific instance of LogicalDevice.

Association Provider : Linux_DeviceAccessedByFile

Reference A	Reference B
Antecedent = “Linux_DeviceFile”	Dependent = “CIM_LogicalDisk”

Instance Table
(see chapter 3.3 “Native Volume Management” and 3.4 “Logical Volume Management” for further information to LogicalDisk and its Linux sub-classes)

specified end	returned instance(s)
Linux_DeviceFile “?”	- (NOT SUPPORTED)
Linux_NativeLogicalDisk “/dev/hda1”	Linux_DeviceFile “/dev/hda1”
Linux_NativeLogicalDisk “/dev/hda2”	Linux_DeviceFile “/dev/hda2”
Linux_LVMLogicalDisk “/dev/vg1/lv1”	Linux_DeviceFile “/dev/vg1/lv1”

3.1.2.5 Linux_ResidesOnExtent : CIM_ResidesOnExtent

... is an association between a FileSystem and the local StorageExtent where it is located. Typically, a FileSystem resides on a LogicalDisk.

This association reflects which instance of LogicalDisk represents the storage range where a FileSystem was created on. The FileSystem is represented by one instance of FileSystem.

Association Provider : Linux_ResidesOnExtent

Reference A	Reference B
Antecedent = "CIM_LogicalDisk"	Dependent = "CIM_LocalFileSystem"

Instance Table
(see chapter 3.3 "Native Volume Management" and 3.4 "Logical Volume Management" for further information to LogicalDisk and its Linux sub-classes)

specified end	returned instance(s)
Linux_NativeLogicalDisk "/dev/hda1"	Linux_Ext2FileSystem "/dev/hda1"
Linux_NativeLogicalDisk "/dev/hda2"	Linux_ReiserFileSystem "/dev/hda2"
Linux_LVMLogicalDisk "/dev/vg1/lv1"	Linux_Ext2FileSystem "/dev/vg1/lv1"
Linux_Ext2FileSystem "/dev/hda1"	Linux_NativeLogicalDisk "/dev/hda1"
Linux_ReiserFileSystem "/dev/hda2"	Linux_NativeLogicalDisk "/dev/hda2"
Linux_Ext2FileSystem "/dev/vg1/lv1"	Linux_LVMLogicalDisk "/dev/vg1/lv1"

3.1.2.6 Linux_FileIdentity : CIM_FileIdentity

... is an association indicating that a UnixFile instance is describing additional aspects of a LogicalFile.

Each LogicalFile has one association to an instance of Linux_UnixFile. This association will be the only support to get the corresponding instance of Linux_UnixFile, hosting more information to the LogicalFile itself. The association provider supports only API calls which return the corresponding instance of UnixFile (one directional – from LogicalFile to UnixFile).

Association Provider : Linux_FileIdentity

Reference A	Reference B
SameElement = "Linux_UnixFile"	SystemElement = "CIM_LogicalFile"

Instance Table

specified end	returned instance(s)
Linux_UnixFile "?"	- (NOT SUPPORTED)
Linux_Directory "/home/user"	Linux_UnixFile "/home/user"; FS Linux_Ext2FileSystem "/dev/hda1"
Linux_Directory "/mnt/dir1"	Linux_UnixFile "/mnt/dir1"; FS Linux_ReiserFileSystem "/dev/hda2"
…	…

3.1.2.7 Linux_FileStorage : CIM_FileStorage

... is a link between the FileSystem and the LogicalFile(s) addressed through this FileSystem.

The association can be used to indicate which LogicalFile(s) are contained by a FileSystem and backward in which FileSystem is a LogicalFile located. This association indirectly offers the possibility to enumerate all LogicalFile(s) contained by a FileSystem and should be used with care. The amount of returned instances will be very high.

Association Provider : Linux_FileStorage

Reference A	Reference B
GroupComponent = "CIM_FileSystem"	PartComponent = "CIM_LogicalFile"

Instance Table

specified end	returned instance(s)
Linux_Directory “/home/user”	Linux_Ext2FileSystem “/dev/hda1”
Linux_ReiserFileSystem “/dev/hda2”	Linux_Directory “/mnt/dir1” Linux_Directory “/mnt/dir2”
Linux_Ext2FileSystem “/dev/vg1/lv1”	-
…	…

3.2.1 Class description

3.2.1.1 Linux_NativeLogicalDisk : CIM_LogicalDisk

... see chapter 3.1.1.4 CIM_LogicalDisk for the definition and intention of LogicalDisk and this sub-class.

A DiskPartition becomes a NativeLogicalDisk, when a file system was created on it (command “mkfs”). The class NativeLogicalDisk returns DiskPartition(s) of type “Linux (83)”, on which a FileSystem was created.

Class Provider : Linux_NativeLogicalDisk

Name	Type	CIM class	Value / Description	Impl.
Caption	string	ManagedElement	short caption to describe the class	x
Description	string	ManagedElement	short description of this class	x
Name	string	ManagedSystemElement	full qualified path of the device file in “/dev/…”	x
Status	string	ManagedSystemElement	current Status of the System : “OK”	x
InstallDate	datetime	ManagedSystemElement		-
SystemCreationClassName	string (key)	LogicalDevice	“Linux_UnitaryComputerSystem”	x
SystemName	string (key)	LogicalDevice	full qualified hostname “host.domain”	x
CreationClassName	string (key)	LogicalDevice	“Linux_NativeLogicalDisk”	x
DeviceID	string (key)	LogicalDevice	full qualified path of the LogicalDisk, e.g. “/dev/hda2”	x
StatusInfo	uint16	LogicalDevice	Status of the device can be enabled(3), disabled(4), other(1), unknown(2) or not applicable(5)	x (-)
OtherIdentifyingInfo[]	string	LogicalDevice	additional information to the device ID	x (-)
IdentifyingDescriptions[]	string	LogicalDevice	description to the value of property OtherIdentifyingInfo	x (-)
Availability	uint16	LogicalDevice	Status of availability, see CIM mapping	x (-)
AdditionalAvailability[]	uint16	LogicalDevice	Additional description to the property Availability	x (-)
MaxQuiesceTime	uint64	LogicalDevice	Max. time in ms, where a device can run in “quiesced” status	-
PowerOnHours	uint64	LogicalDevice	total number of hours since the last power cycle	x (-)
TotalPowerOnHours	uint64	LogicalDevice	total number of hours the device was under power	x (-)
PowerManagementCapabilities[]	uint16	LogicalDevice	Power mgmt. related capabilities of the device	-
PowerManagementSupported	boolean	LogicalDevice	Device can be power managed	-
LastErrorCode	uint32	LogicalDevice	Last error code reported by the device	-
ErrorCleared	boolean	LogicalDevice	Error was cleared	-
ErrorDescription	string	LogicalDevice	Additional error description	-
BlockSize	uint64	StorageExtent	Block size of the extent in bytes	x
NumberOfBlocks	uint64	StorageExtent	Total number of blocks this extent consists of	x
ConsumableBlocks	uint64	StorageExtent	Max. number of blocks available for consumption	x
Access	uint16	StorageExtent	Media is readable(1), writable(2), both(3), unknown(0), write once(4)	x (-)
SequentialAccess	boolean	StorageExtent	TRUE if sequentially accessed by MediaAccessDevice	-
Purpose	string	StorageExtent	Free from string to describe the media	x (-)
DataOrganization	uint16	StorageExtent	Type of data organization	-
ErrorMethodology	string	StorageExtent	free-form string to describe error detection and correction	-
IsBasedOnUnderlyingRedund.	boolean	StorageExtent	Underlying StorageExtents participate in a StorageRedundancyGroup	-
MajorNumber	uint16	Linux_LogicalDisk	Device major ID	x
MinorNumber	uint16	Linux_LogicalDisk	Device minor ID	x

The following table contains all instances returned by the Linux_NativeLogicalDisk provider for the example environment described at beginning of this chapter.

Linux Class (Property CreationClassName)	Key Property = “Value”
Linux_NativeLogicalDisk	Name = “/dev/hda1”
Linux_NativeLogicalDisk	Name = “/dev/hda2”

3.2.1.2 Linux_DiskPartition : CIM_DiskPartition

... is a presentation of a contiguous range of logical blocks that is identifiable by the OperatingSystem via the Partition's type and subtype fields. DiskPartitions should be directly realized by PhysicalMedia (indicated by the RealizesDiskPartition association) or built on StorageVolumes (indicated by the PartitionBasedOnVolume association).

Typically, one or more DiskPartitions are placed on a single disk via 'fdisk'. This class does not distinguish between 'native' and 'LVM' partitions. It enumerates all physical partitions, which have been defined via 'fdisk' on the physical disk.

Class Provider : Linux_DiskPartition

Name	Type	CIM class	Value / Description	Impl.
Caption	string	ManagedElement	short caption to describe the class	x
Description	string	ManagedElement	short description of this class	x
Name	string	ManagedSystemElement	full qualified path of the device file in "/dev/…"	x
Status	string	ManagedSystemElement	current Status of the System : "OK"	x
InstallDate	datetime	ManagedSystemElement		-
SystemCreationClassName	string (key)	LogicalDevice	"Linux_UnitaryComputerSystem"	x
SystemName	string (key)	LogicalDevice	full qualified hostname "host.domain"	x
CreationClassName	string (key)	LogicalDevice	"Linux_DiskPartition"	x
DeviceID	string (key)	LogicalDevice	full qualified name of the DiskPartition, e.g. "hda2"	x
StatusInfo	uint16	LogicalDevice	Status of the device can be enabled(3), disabled(4), other(1), unknown(2) or not applicable(5)	x (-)
OtherIdentifyingInfo[]	string	LogicalDevice	additional information to the device ID	x (-)
IdentifyingDescriptions[]	string	LogicalDevice	description to the value of property OtherIdentifyingInfo	x (-)
Availability	uint16	LogicalDevice	Status of availability, see CIM mapping	x (-)
AdditionalAvailability[]	uint16	LogicalDevice	Additional description to the property Availability	x (-)
MaxQuiesceTime	uint64	LogicalDevice	Max. time in ms, where a device can run in "quiesced" status	-
PowerOnHours	uint64	LogicalDevice	total number of hours since the last power cycle	x (-)
TotalPowerOnHours	uint64	LogicalDevice	total number of hours the device was under power	x (-)
PowerManagementCapabilities[]	uint16	LogicalDevice	Power mgmt. related capabilities of the device	-
PowerManagementSupported	boolean	LogicalDevice	Device can be power managed	-
LastErrorCode	uint32	LogicalDevice	Last error code reported by the device	-
ErrorCleared	boolean	LogicalDevice	Error was cleared	-
ErrorDescription	string	LogicalDevice	Additional error description	-
BlockSize	uint64	StorageExtent	Block size of the extent in bytes	x
NumberOfBlocks	uint64	StorageExtent	Total number of blocks this extent consists of	x
ConsumableBlocks	uint64	StorageExtent	Max. number of blocks available for consumption	x
Access	uint16	StorageExtent	Media is readable(1), writable(2), both(3), unknown(0), write once(4)	x (-)
SequentialAccess	boolean	StorageExtent	TRUE if sequentially accessed by MediaAccessDevice	-
Purpose	string	StorageExtent	Free from string to describe the media	x (-)
DataOrganization	uint16	StorageExtent	Type of data organization	-
ErrorMethodology	string	StorageExtent	free-form string to describe error detection and correction	-
IsBasedOnUnderlyingRedund.	boolean	StorageExtent	Underlying StorageExtents participate in a StorageRedundancyGroup	-
Allocatable	boolean	MediaPartition	Available and can be allocated for use	x (-)
Bootable	boolean	MediaPartition	Labeled as bootable	x (-)
Extendable	boolean	MediaPartition	Can grow / extend without reformatting	x (-)
Signature	string	MediaPartition	Identifying string	-
SignatureAlgorithm	string	MediaPartition	Free-form string to describe the algorithm of signature	-
SignatureState	string	MediaPartition	State of signature	-
PartitionType	uint16	DiskParttion	Type of partition	x (-)
PrimaryPartition	boolean	DiskParttion	Is primary partition	x (-)
MajorNumber	uint16	Linux_DiskPartition	Device major ID	x
MinorNumber	uint16	Linux_DiskPartition	Device minor ID	x

The following table contains all instances returned by the Linux_DiskPartition provider for the example environment described at beginning of this chapter.

Linux Class (Property CreationClassName)	Key Property = "Value"
Linux_DiskPartition	DeviceID = "/hda1"
Linux_DiskPartition	DeviceID = "hda2"
Linux_DiskPartition	DeviceID = "hda3"
Linux_DiskPartition	DeviceID = "hda4"
Linux_DiskPartition	DeviceID = "hda5"

3.2.1.3 Linux_StorageVolume : CIM_StorageVolume

... is an Extent that is presented to the OperatingSystem or to another entity. StorageVolumes are directly realized in hardware or are the end result of assembling lower level Extents.

In this case the class represents the logical view of single disk, which are directly realized in hardware.

Class Provider : Linux_StorageVolume

Name	Type	CIM class	Value / Description	Impl.
Caption	string	ManagedElement	short caption to describe the class	x
Description	string	ManagedElement	short description of this class	x
Name	string	ManagedSystemElement	full qualified path of the device file in “/dev/...”	x
Status	string	ManagedSystemElement	current Status of the System : “OK”	x
InstallDate	datetime	ManagedSystemElement		-
SystemCreationClassName	string (key)	LogicalDevice	“Linux_UnitaryComputerSystem”	x
SystemName	string (key)	LogicalDevice	full qualified hostname “host.domain”	x
CreationClassName	string (key)	LogicalDevice	“Linux_StorageVolume”	x
DeviceID	string (key)	LogicalDevice	full qualified path of theStrageVolume, e.g. “ide0”	x
StatusInfo	uint16	LogicalDevice	Status of the device can be enabled(3), disabled(4), other(1), unknown(2) or not applicable(5)	x (-)
OtherIdentifyingInfo[]	string	LogicalDevice	additional information to the device ID	x (-)
IdentifyingDescriptions[]	string	LogicalDevice	description to the value of property OtherIdentifyingInfo	x (-)
Availability	uint16	LogicalDevice	Status of availability, see CIM mapping	x (-)
AdditionalAvailability[]	uint16	LogicalDevice	Additional description to the property Availability	x (-)
MaxQuiesceTime	uint64	LogicalDevice	Max. time in ms, where a device can run in “quiesced” status	-
PowerOnHours	uint64	LogicalDevice	total number of hours since the last power cycle	x (-)
TotalPowerOnHours	uint64	LogicalDevice	total number of hours the device was under power	x (-)
PowerManagementCapabilities[]	uint16	LogicalDevice	Power mgmt. related capabilities of the device	-
PowerManagementSupported	boolean	LogicalDevice	Device can be power managed	-
LastErrorCode	uint32	LogicalDevice	Last error code reported by the device	-
ErrorCleared	boolean	LogicalDevice	Error was cleared	-
ErrorDescription	string	LogicalDevice	Additional error description	-
BlockSize	uint64	StorageExtent	Block size of the extent in bytes	x
NumberOfBlocks	uint64	StorageExtent	Total number of blocks this extent consists of	x
ConsumableBlocks	uint64	StorageExtent	Max. number of blocks available for consumption	x
Access	uint16	StorageExtent	Media is readable(1), writable(2), both(3), unknown(0), write once(4)	x (-)
SequentialAccess	boolean	StorageExtent	TRUE if sequentially accessed by MediaAccessDevice	-
Purpose	string	StorageExtent	Free from string to describe the media	x (-)
DataOrganization	uint16	StorageExtent	Type of data organization	-
ErrorMethodology	string	StorageExtent	free-form string to describe error detection and correction	-
IsBasedOnUnderlyingRedund.	boolean	StorageExtent	Underlying StorageExtents participate in a StorageRedundancyGroup	-
MajorNumber	uint16	Linux_StorageVolume	Device major ID	x

The following table contains all instances returned by the Linux_StorageVolume provider for the example environment described at beginning of this chapter.

Linux Class (Property CreationClassName)	Key Property = “Value”
Linux_StorageVolume	DeviceID = “ide0”

3.2.2 Association description

3.2.2.1 Linux_LogicalDiskBasedOnPartition : CIM_LogicalDiskBasedOnPartition

... aggregates the NativeLogicalDisk with the DiskPartition (type 'native'), it is based on .

An instance of NativeLogicalDisk is based directly on an instance of DiskPartition. The Antecedent DiskPartition is Max(1), what dictates that a LogicalDisk can not span more than one Partition. However, there are cases where this is true, e.g. when DiskPartitions are managed by a software volume manager. When it occurs, the LogicalDisk is more correctly based on a StorageVolume.

Via this association a client can figure out which instances of LogicalDisk are based on 'native' DiskPartitions and indirectly which instances of LogicalDisk are managed by the software volume manager 'LVM for Linux' .

Association Provider : Linux_LogicalDiskBasedOnPartition

Reference A	Reference B
Antecedent = “Linux_DiskPartition”	Dependent = “Linux_NativeLogicalDisk”

Instance Table

specified end	returned instance(s)
Linux_NativeLogicalDisk “/dev/hda1”	Linux_DiskPartition “hda1”
Linux_NativeLogicalDisk “/dev/hda2”	Linux_DiskPartition “hda2”
Linux_LVMLogicalDisk “/dev/vg1/lv1”	-
Linux_DiskPartition “hda1”	Linux_NativeLogicalDisk “/dev/hda1”
Linux_DiskPartition “hda2”	Linux_NativeLogicalDisk “/dev/hda2”
Linux_DiskPartition “hda3”	-
Linux_DiskPartition “hda4”	-
Linux_DiskPartition “hda5”	-

3.2.2.2 Linux_DiskPartitionBasedOnVolume : CIM_DiskPartitionBasedOnVolume

... represents which DiskPartitions are based on a single StorageVolume.

An instance of DiskPartition is based on only one StorageVolume Max(1), where the StorageVolume presents a logical view onto the PhysicalMedia. An alternative is the Linux_RealizesDiskPartition association.

Association Provider : Linux_DiskPartitionBasedOnVolume

Reference A	Reference B
Antecedent = “Linux_StorageVolume”	Dependent = “Linux_DiskPartition”

Instance Table

specified end	returned instance(s)
Linux_StorageVolume “ide0”	Linux_DiskPartition “hda1” Linux_DiskPartition “hda2” Linux_DiskPartition “hda3” Linux_DiskPartition “hda4” Linux_DiskPartition “hda5”
Linux_DiskPartition “hda1”	Linux_StorageVolume “ide0”
Linux_DiskPartition “hda2”	Linux_StorageVolume “ide0”
Linux_DiskPartition “hda3”	Linux_StorageVolume “ide0”
Linux_DiskPartition “hda4”	Linux_StorageVolume “ide0”
Linux_DiskPartition “hda5”	Linux_StorageVolume “ide0”

3.3 Logical Volume Management

See “Appendix B : Logical Volume Management” for the CIM class hierarchy and “Appendix B : Association Hierarchy” for the Associations. The left part of picture 3.1 “File System and Volume Management” shows a more common view than the CIM Hierarchy does.

3.3.1 Class Description

3.3.1.1 Linux_LVMVolumeGroup : CIM_DiskGroup

... consolidates one to many DiskPartition and consists of 0 to many LVMLogicalVolumes. The summarized sizes of the DiskPartitions define the maximum capacity of the LVMVolumeGroup.

Class Provider : Linux_LVMVolumeGroup

Name	Type	CIM class	Value / Description	Impl.
Caption	string	ManagedElement	short caption to describe the class	x
Description	string	ManagedElement	short description of this class	x
CollectionID	string (key)	CollectionOfMSEs	contains name of volume group, e.g. “vg1”	x
Name	string	Linux_LVMVolumeGroup	contains the UID of the volume group	x
VolumeStatus	uint16	Linux_LVMVolumeGroup	Status of colume group (active / deactive)	x
Capacity	uint64	Linux_LVMVolumeGroup	total kBytes of VG	x
AllocatedSpace	uint64	Linux_LVMVolumeGroup	allocated kBytes of VG	x
FreeSpace	uint64	Linux_LVMVolumeGroup	free kBytes of VG	x
NumberOfPEs	uint64	Linux_LVMVolumeGroup	total number of Physical Extents	x
NumberOfFreePEs	uint64	Linux_LVMVolumeGroup	free number of Physical Extents	x
NumberOfAllocatedPEs	uint64	Linux_LVMVolumeGroup	allocated number of Physical Extents	x
PhysicalExtentSize	uint64	Linux_LVMVolumeGroup	Physical Extent Size in kBytes	x
MaxNumberOfPVs	uint64	Linux_LVMVolumeGroup	max. number of physical volumes	x
NumberOfDefinedPVs	uint64	Linux_LVMVolumeGroup	defined number of physical volumes	x
NumberOfActivePVs	uint64	Linux_LVMVolumeGroup	active number of physical volumes	x
MaxNumberOfLVs	uint64	Linux_LVMVolumeGroup	max. number of logical volumes	x
NumberOfDefiniedLVs	uint64	Linux_LVMVolumeGroup	defined number of logical volumes	x
NumberOfActiveLVs	uint64	Linux_LVMVolumeGroup	active number of logical volumes	x
MaxSizeOfLV	uint64	Linux_LVMVolumeGroup	max. size of logical volume	x
NumberOfVGDAs	uint64	Linux_LVMVolumeGroup		-
activate()	string	Linux_LVMVolumeGroup	method : activate volume group	x
deactivate()	string	Linux_LVMVolumeGroup	method : deactivate volume group	x
createLogicalVolume()	string	Linux_LVMVolumeGroup	method : create new LV in volume group	x

The following table contains all instances returned by the Linux_LVMVolumeGroup provider for the example environment described on page 7 and 8.

Linux Class (Property CreationClassName)	Key Property = “Value”
Linux_LVMVolumeGroup	CollectionID = “vg1”
Linux_LVMVolumeGroup	CollectionID = “vg2”

3.3.1.2 Linux_LVMLogicalDisk : CIM_LogicalDisk

... see chapter 3.1.1.4 CIM_LogicalDisk for the definition and intention of LogicalDisk and this sub-class.

The class LVMLogicalDisk returns these LVMLogicalVolumes, on which a FileSystem was created (command “mkfs”).

Class Provider : Linux_LVMLogicalDisk

Name	Type	CIM class	Value / Description	Impl.
Caption	string	ManagedElement	short caption to describe the class	x
Description	string	ManagedElement	short description of this class	x
Name	string	ManagedSystemElement	full qualified path of the device file in “/dev/…”	x
Status	string	ManagedSystemElement	current Status of the System : “OK”	x
InstallDate	datetime	ManagedSystemElement		-
SystemCreationClassName	string (key)	LogicalDevice	“Linux_UnitaryComputerSystem”	x

SystemName	string (key)	LogicalDevice	full qualified hostname “host.domain”	x
CreationClassName	string (key)	LogicalDevice	“Linux_LVMLogicalDisk”	x
DeviceID	string (key)	LogicalDevice	full qualified path of the LogicalDisk, e.g. “/dev/vg1/lv1”	x
StatusInfo	uint16	LogicalDevice	Status of the device can be enabled(3), disabled(4), other(1), unknown(2) or not applicable(5)	x (-)
OtherIdentifyingInfo[]	string	LogicalDevice	additional information to the device ID	x (-)
IdentifyingDescriptions[]	string	LogicalDevice	description to the value of property OtherIdentifyingInfo	x (-)
Availability	uint16	LogicalDevice	Status of availability, see CIM mapping	x (-)
AdditionalAvailability[]	uint16	LogicalDevice	Additional description to the property Availability	x (-)
MaxQuiesceTime	uint64	LogicalDevice	Max. time in ms, where a device can run in “quiesced” status	-
PowerOnHours	uint64	LogicalDevice	total number of hours since the last power cycle	x (-)
TotalPowerOnHours	uint64	LogicalDevice	total number of hours the device was under power	x (-)
PowerManagementCapabilities[]	uint16	LogicalDevice	Power mgmt. related capabilities of the device	-
PowerManagementSupported	boolean	LogicalDevice	Device can be power managed	-
LastErrorCode	uint32	LogicalDevice	Last error code reported by the device	-
ErrorCleared	boolean	LogicalDevice	Error was cleared	-
ErrorDescription	string	LogicalDevice	Additional error description	-
BlockSize	uint64	StorageExtent	Block size of the extent in bytes	x
NumberOfBlocks	uint64	StorageExtent	Total number of blocks this extent consists of	x
ConsumableBlocks	uint64	StorageExtent	Max. number of blocks available for consumption	x
Access	uint16	StorageExtent	Media is readable(1), writable(2), both(3), unknown(0), write once(4)	x (-)
SequentialAccess	boolean	StorageExtent	TRUE if sequentially accessed by MediaAccessDevice	-
Purpose	string	StorageExtent	Free from string to describe the media	x (-)
DataOrganization	uint16	StorageExtent	Type of data organization	-
ErrorMethodology	string	StorageExtent	free-form string to describe error detection and correction	-
IsBasedOnUnderlyingRedund.	boolean	StorageExtent	Underlying StorageExtents participate in a StorageRedundancyGroup	-
MajorNumber	uint16	Linux_LogicalDisk	Device major ID	x
MinorNumber	uint16	Linux_LogicalDisk	Device minor ID	x

The following table contains all instances returned by the Linux_LVMLogicalDisk provider for the example environment described at beginning of this chapter.

Linux Class (Property CreationClassName)	Key Property = “Value”
Linux_LVMLogicalDisk	Name = “/dev/vg1/lv1”

3.3.1.3 Linux_LVMLogicalVolume : CIM_StorageVolume

Typically, one or more LogicalVolumes are defined within a VolumeGroup. This class enumerates all LogicalVolumes of the system, independent of their connection to the VolumeGroup.

Class Provider : Linux_LVMLogicalVolume

Name	Type	CIM class	Value / Description	Impl.
Caption	string	ManagedElement	short caption to describe the class	x
Description	string	ManagedElement	short description of this class	x
Name	string	ManagedSystemElement	UID of the logical volume	x
Status	string	ManagedSystemElement	current Status of the LV : “OK”	x
InstallDate	datetime	ManagedSystemElement		-
SystemCreationClassName	string (key)	LogicalDevice	“Linux_UnitaryComputerSystem”	x
SystemName	string (key)	LogicalDevice	full qualified hostname “host.domain”	x
CreationClassName	string (key)	LogicalDevice	“Linux_LVMLogicalVolume”	x
DeviceID	string (key)	LogicalDevice	full qualified path of the LVMLogicalVolume, e.g. “/dev/vg1/lv1”	x
StatusInfo	uint16	LogicalDevice	Status of the device can be enabled(3), disabled(4), other(1), unknown(2) or not applicable(5)	x (-)
OtherIdentifyingInfo[]	string	LogicalDevice	additional information to the device ID	x (-)
IdentifyingDescriptions[]	string	LogicalDevice	description to the value of property OtherIdentifyingInfo	x (-)
Availability	uint16	LogicalDevice	Status of availability, see CIM mapping	x (-)
AdditionalAvailability[]	uint16	LogicalDevice	Additional description to the property Availability	x (-)
MaxQuiesceTime	uint64	LogicalDevice	Max. time in ms, where a device can run in “quiesced” status	-
PowerOnHours	uint64	LogicalDevice	total number of hours since the last power cycle	x (-)
TotalPowerOnHours	uint64	LogicalDevice	total number of hours the device was under power	x (-)
PowerManagementCapabilities[]	uint16	LogicalDevice	Power mgmt. related capabilities of the device	-
PowerManagementSupported	boolean	LogicalDevice	Device can be power managed	-
LastErrorCode	uint32	LogicalDevice	Last error code reported by the device	-
ErrorCleared	boolean	LogicalDevice	Error was cleared	-
ErrorDescription	string	LogicalDevice	Additional error description	-

BlockSize	uint64	StorageExtent	Block size of the logical extent (LE) in bytes	x
NumberOfBlocks	uint64	StorageExtent	Total number of blocks (LE) this LV consists of	x
ConsumableBlocks	uint64	StorageExtent	Max. number of blocks (LE) available for consumption	x
Access	uint16	StorageExtent	Media is readable(1), writable(2), both(3), unknown(0), write once(4)	x (-)
SequentialAccess	boolean	StorageExtent	TRUE if sequentially accessed by MediaAccessDevice	-
Purpose	string	StorageExtent	Free from string to describe the media	x (-)
DataOrganization	uint16	StorageExtent	Type of data organization	-
ErrorMethodology	string	StorageExtent	free-form string to describe error detection and correction	-
IsBasedOnUnderlyingRedund.	boolean	StorageExtent	Underlying StorageExtents participate in a StorageRedundancyGroup	-
Capacity	uint64	Linux_LVMLogicalVolume	total kBytes of LV	x
AllocatedSpace	uint64	Linux_LVMLogicalVolume	allocated kBytes of LV	x
FreeSpace	uint64	Linux_LVMLogicalVolume	free kBytes of LV	x
AllocatedBlocks	uint64	Linux_LVMLogicalVolume	number of allocated blocks (LE)	x
Striped	boolean	Linux_LVMLogicalVolume	LV was created as “striped”	x
StripeSize	uint64	Linux_LVMLogicalVolume	size of Stripe in kByte	x
NumberOfStripes	uint64	Linux_LVMLogicalVolume	total number of stripes	x
MajorNumber	uint16	Linux_LVMLogicalVolume	Device major ID	x
MinorNumber	uint16	Linux_LVMLogicalVolume	Device minor ID	x

The following table contains all instances returned by the Linux_LVMLogicalVolume provider for the example environment described at beginning of this chapter.

Linux Class (Property CreationClassName)	Key Property = “Value”
Linux_LVMLogicalVolume	DeviceID = “/dev/vg1/lv1”

3.3.1.4 Linux_LVMDiskPartition : CIM_DiskPartition

Typically, one or more LVMDiskPartitions are consolidated within a VolumeGroup. This class enumerates all LVMDiskPartitions of the system, independent of their connection to a VolumeGroup. The enumeration of a DiskPartition as an LVMDiskPartition depends on the partition type - must be 'Linux LVM (8e)'.

Class Provider : Linux_LVMDiskPartition

Name	Type	CIM class	Value / Description	Impl.
Caption	string	ManagedElement	short caption to describe the class	x
Description	string	ManagedElement	short description of this class	x
Name	string	ManagedSystemElement	UID of the physical volume	x
Status	string	ManagedSystemElement	current Status of the PV : “OK”	x
InstallDate	datetime	ManagedSystemElement		-
SystemCreationClassName	string (key)	LogicalDevice	“Linux_UnitaryComputerSystem”	x
SystemName	string (key)	LogicalDevice	full qualified hostname “host.domain”	x
CreationClassName	string (key)	LogicalDevice	“Linux_LVMDiskPartition”	x
DeviceID	string (key)	LogicalDevice	full qualified path of the LVMDiskPartition, e.g. “/dev/hda2”	x
StatusInfo	uint16	LogicalDevice	Status of the device can be enabled(3), disabled(4), other(1), unknown(2) or not applicable(5)	x (-)
OtherIdentifyingInfo[]	string	LogicalDevice	additional information to the device ID	x (-)
IdentifyingDescriptions[]	string	LogicalDevice	description to the value of property OtherIdentifyingInfo	x (-)
Availability	uint16	LogicalDevice	Status of availability, see CIM mapping	x (-)
AdditionalAvailability[]	uint16	LogicalDevice	Additional description to the property Availability	x (-)
MaxQuiesceTime	uint64	LogicalDevice	Max. time in ms, where a device can run in “quiesced” status	-
PowerOnHours	uint64	LogicalDevice	total number of hours since the last power cycle	x (-)
TotalPowerOnHours	uint64	LogicalDevice	total number of hours the device was under power	x (-)
PowerManagementCapabilities[]	uint16	LogicalDevice	Power mgmt. related capabilities of the device	-
PowerManagementSupported	boolean	LogicalDevice	Device can be power managed	-
LastErrorCode	uint32	LogicalDevice	Last error code reported by the device	-
ErrorCleared	boolean	LogicalDevice	Error was cleared	-
ErrorDescription	string	LogicalDevice	Additional error description	-
BlockSize	uint64	StorageExtent	Block size of the physical extent (PE) in bytes	x
NumberOfBlocks	uint64	StorageExtent	Total number of blocks (PE) this PV consists of	x
ConsumableBlocks	uint64	StorageExtent	Max. number of blocks (PE) available for consumption	x
Access	uint16	StorageExtent	Media is readable(1), writable(2), both(3), unknown(0), write once(4)	x (-)
SequentialAccess	boolean	StorageExtent	TRUE if sequentially accessed by MediaAccessDevice	-
Purpose	string	StorageExtent	Free from string to describe the media	x (-)
DataOrganization	uint16	StorageExtent	Type of data organization	-
ErrorMethodology	string	StorageExtent	free-form string to describe error detection and correction	-

IsBasedOnUnderlyingRedund.	boolean	StorageExtent	Underlying StorageExtents participate in a StorageRedundancyGroup	-
Allocatable	boolean	MediaPartition	Available and can be allocated for use	x (-)
Bootable	boolean	MediaPartition	Labeled as bootable	x (-)
Extendable	boolean	MediaPartition	Can grow / extend without reformatting	x (-)
Signature	string	MediaPartition	Identifying string	-
SignatureAlgorithm	string	MediaPartition	Free-form string to describe the algorithm of signature	-
SignatureState	string	MediaPartition	State of signature	-
PartitionType	uint16	DiskParttion	Type of partition	x (-)
PrimaryPartition	boolean	DiskParttion	Is primary partition	x (-)
NumberOfLVsUsingPV	uint64	Linux_LVMDiskPartition	Number of LVs located on this PV	x
NumberOfVGDAs	uint64	Linux_LVMDiskPartition		x (-)
VolumeGroup	string	Linux_LVMDiskPartition	Name of the volume group	x
MajorNumber	uint16	Linux_LVMDiskPartition	Device major ID	x
MinorNumber	uint16	Linux_LVMDiskPartition	Device minor ID	x

The following table contains all instances returned by the Linux_LVMDiskPartition provider for the example environment described at beginning of this chapter.

Linux Class (Property CreationClassName)	Key Property = "Value"
Linux_LVMDiskPartition	DeviceID = "/dev/hda3"
Linux_LVMDiskPartition	DeviceID = "/dev/hda4"
Linux_LVMDiskPartition	DeviceID = "/dev/hda5"

3.3.2 Association Description

3.3.2.1 Linux_LVMLogicalDiskBasedOnLV : CIM_LogicalDiskBasedOnVolume

... aggregates the LogicalDisk with the LVMLogicalVolume, it is based on.

An LVMLogicalDisk is based on a single StorageVolume; in this case the LVMLogicalVolume exposed by the software volume manager 'LVM for Linux'. This relationship is made explicit in this association. An LVMLogicalDisk can be based on Max(1) LVMLogicalVolumes.

Association Provider : Linux_LVMLogicalDiskBasedOnLV

Reference A	Reference B
Antecedent = "Linux_LVMLogicalVolume"	Dependent = "Linux_LVMLogicalDisk"

Instance Table

specified end	returned instance(s)
Linux_NativeLogicalDisk "/dev/hda1"	-
Linux_NativeLogicalDisk "/dev/hda2"	-
Linux_LVMLogicalDisk "/dev/vg1/lv1"	Linux_LVMLogicalVolume "/dev/vg1/lv1"
Linux_LVMLogicalVolume "/dev/vg1/lv1"	Linux_LVMLogicalDisk "/dev/vg1/lv1"

3.3.2.2 Linux_LVMLVInVG : CIM_ExtentInDiskGroup

... shows which LVMLogicalVolumes are hosted by which LVMVolumeGroup.
Association Provider : Linux_LVMLVInVG

Reference A	Reference B
Member = "Linux_LVMLogicalVolume"	Collection = "Linux_LVMVolumeGroup"

Instance Table

specified end	returned instance(s)
Linux_LVMLogicalVolume "/dev/vg1/lv1"	Linux_LVMVolumeGroup "vg1"
Linux_LVMVolumeGroup "vg1"	Linux_LVMLogicalVolume "/dev/vg1/lv1"
Linux_LVMVolumeGroup "vg2"	-

3.3.2.3 Linux_LVMDPInVG : CIM_ExtentInDiskGroup

... shows which LVMPhysicalVolumes are hosted by which LVMVolumeGroup.

Association Provider : Linux_LVMDPInVG

Reference A	Reference B
Member = “Linux_LVMDiskPartition”	Collection = “Linux_LVMVolumeGroup”

Instance Table

specified end	returned instance(s)
Linux_LVMDiskPartition “/dev/hda3”	Linux_LVMVolumeGroup “vg1”
Linux_LVMDiskPartition “/dev/hda4”	Linux_LVMVolumeGroup “vg1”
Linux_LVMDiskPartition “/dev/hda5”	Linux_LVMVolumeGroup “vg2”
Linux_LVMVolumeGroup “vg1”	Linux_LVMDiskPartition “/dev/hda3” Linux_LVMDiskPartition “/dev/hda4”
Linux_LVMVolumeGroup “vg2”	Linux_LVMDiskPartition “/dev/hda5”

3.3.2.4 Linux_LVMLVBasedOnDP : CIM_BasedOn

... shows the mapping between LVMLogicalVolume and LVMDiskPartition.

The 'LVM for Linux' supports two different kinds of mapping methods : linear and striped. One instance of LVMLogicalVolume can be based on 1-to many LVMDiskPartitions and one LVMDiskPartition can 'host' 0-to many LVMLogicalVolumes. This depends on the relation of the free space on the LVMDiskPartition to the needed space by the LVMLogicalVolume and the mapping method.

Association Provider : Linux_LVMLVBasedOnDP

Reference A	Reference B
Antecedent = “Linux_LVMDiskPartition”	Dependent = “Linux_LVMLogicalVolume”

Instance Table

specified end	returned instance(s)
Linux_LVMDiskPartition “/dev/hda3”	Linux_LVMLogicalVolume “/dev/ vg1/lv1”
Linux_LVMDiskPartition “/dev/hda4”	Linux_LVMLogicalVolume “/dev/vg1/lv1”
Linux_LVMDiskPartition “/dev/hda5”	-
Linux_LVMLogicalVolume “/dev/vg1/lv1”	Linux_LVMDiskPartition “/dev/hda3” Linux_LVMDiskPartition “/dev/hda4”

3.3.2.5 Linux_LVMDPIsDiskPartition : CIM_DeviceIdentity

... presents which DiskPartition is a LVMDiskPartition.

Association Provider : Linux_LVMDPIsDiskPartition

Reference A	Reference B
SameElement = “Linux_LVMDiskPartition”	SystemElement = “Linux_DiskPartition”

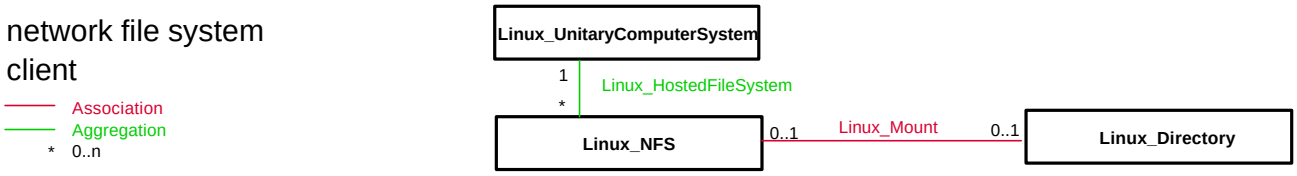
Instance Table

specified end	returned instance(s)
Linux_LVMDiskPartition “/dev/hda3”	Linux_DiskPartition “hda3”
Linux_LVMDiskPartition “/dev/hda4”	Linux_DiskPartition “hda4”
Linux_LVMDiskPartition “/dev/hda5”	Linux_DiskPartition “hda5”
Linux_DiskPartition “hda1”	-
Linux_DiskPartition “hda2”	-
Linux_DiskPartition “hda3”	Linux_LVMDiskPartition “/dev/hda3”
Linux_DiskPartition “hda4”	Linux_LVMDiskPartition “/dev/hda4”
Linux_DiskPartition “hda5”	Linux_LVMDiskPartition “/dev/hda5”

3.4 Network File System Management

3.4.1 Client

The necessary CIM schema parts for Network File System Management on Client side are Core, Unix and System. Appendix B “NFS Mgmt.” shows the CIM hierarchy and the parents of the classes / associations.



3.3 NFS Client Schema

3.4.1.1 Linux_NFS : CIM_NFS

... derived from RemoteFileSystem representing that the FileSystem is mounted, using the NFS protocol. The properties of the NFS object deal with the operational aspects of the mount and represent the client-side configuration for NFS access. The FileSystemType (inherited from FileSystem) should be set to indicate the type of this FileSystem as it appears to the client.

Class Provider : Linux_NFS

Name	Type	CIM class	Value / Description	Impl.
Caption	string	ManagedElement	short caption to describe the class	x
Description	string	ManagedElement	short description of this class	x
Name	string (key)	ManagedSystemElement	full qualified path on the NFS Server, where the directory was exported for mounts, e.g. “nfshost:/nfs/pub”	x
Status	string	ManagedSystemElement	current Status of the System : “OK”	x
InstallDate	datetime	ManagedSystemElement		-
CSCreationClassName	string (key)	FileSystem	“Linux_UnitaryComputerSystem”	x
CSName	string (key)	FileSystem	full qualified hostname “host.domain”	x
CreationClassName	string (key)	FileSystem	“Linux_NFS”	x
Root	string	FileSystem	Pathname defining the root of the filesystem, e.g. “/”	x
BlockSize	uint64	FileSystem	Filesystem’s block size for data storage and retrieval	x
AvailableSpace	uint64	FileSystem	total amount of free space in bytes	x
FileSystemSize	uint64	FileSystem	Total size in bytes	x
FileSystemType	string	FileSystem	type	x
MaxFileNameLength	uint32	FileSystem	max. length of file names	x
ReadOnly	boolean	FileSystem	Filesystem is read only	x
CaseSensitive	boolean	FileSystem	case sensitive file names are supported	-
CasePreserved	boolean	FileSystem	case of the file names are preserved	-
EncryptionMethod	string	FileSystem	Algorithm / tool used to encrypt the filesystem	-
CompressionMethod	string	FileSystem	Algorithm / tool used to compress the filesystem	-
CodeSet[]	uint16	FileSystem	character sets / encoding supported by the filesystem	-
ClusterSize	uint32	FileSystem	minimum file allocation size	x (-)
AttributeCaching	boolean	NFS	Control attribute caching enabled	-
AttributeCachingForDirMax	uint16	NFS	max. number of seconds cached attributes are held after directory update	-
AttributeCachingForDirMin	uint16	NFS	min. number of seconds cached attributes are held after directory update	-
AttributeCachingForRegFileMax	uint16	NFS	max. number of seconds cached attributes are held after file modification	-
AttributeCachingForRegFileMin	uint16	NFS	Min. number of seconds cached attributes are held after file modification	-
ForegroundMount	boolean	NFS	Retries are performed in forground	x (-)
HardMounted	boolean	NFS	Retries are performed until hosting system responses	x (-)
Interrupt	boolean	NFS	Interrupts permitted for hardmount	x (-)
MountFailureRetries	uint16	NFS	max. number of retries by mount failure	x (-)
ReadBufferSize	uint64	NFS	Read buffer size in bytes	x (-)
RetransmissionAttempts	uint16	NFS	max. number of NFS retransmission allowed	x (-)
RetransmissionTimeout	uint32	NFS	NFS timeout in tenth of seconds	x (-)
ServerCommunicationPort	uint32	NFS	Remote CS’s UDP port number	x (-)
WriteBufferSize	uint64	NFS	Write buffer size in bytes	x (-)

The Client has the two local and one remote File Systems mounted :

```
> mount
/dev/hda1          on /              type ext2         (rw)
/dev/hda2          on /mnt           type reiserfs     (rw)
nfshost:/home/pub  on /home/user     type nfs          (rw,addr=172.18.179.112)
```

The following table contains all instances returned by the Linux_NFS provider for the example environment above.

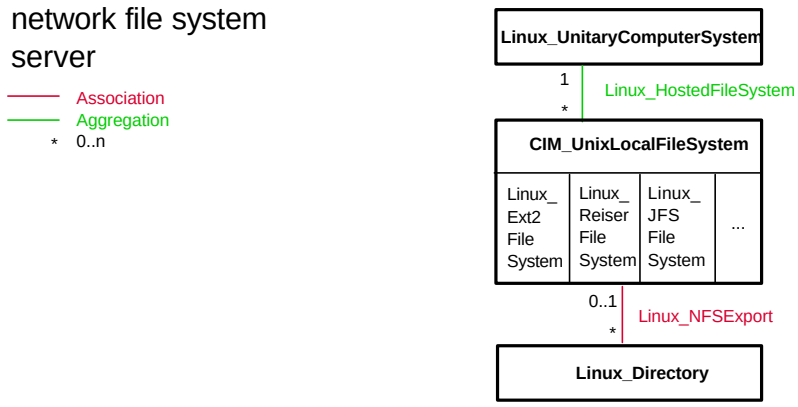
Linux Class (Property CreationClassName)	Key Property = "Value"
Linux_NFS	Name = "nfshost:/home/pub"

Instance Table of association Linux_Mount

specified end	returned instance(s)
Linux_Directory "?"	- (NOT SUPPORTED)
Linux_Ext2FileSystem "/dev/hda1"	Linux_Directory "/"
Linux_ReiserFileSystem "/dev/hda2"	Linux_Directory "/mnt"
Linux_Ext2FileSystem "/dev/vg1/lv1"	-
Linux_NFS "nfshost:/home/pub"	Linux_Directory "/home/user"

3.4.2 Server

The necessary CIM schema parts for Network File System Management on Server side are Core, Unix and System. Appendix B "NFS Mgmt." shows the CIM hierarchy and the parents of the classes / associations.



3.4 NFS Server Schema

3.4.2.1 Linux_NFSExport : CIM_Export

... is an association between a LocalFileSystem and its Directories indicating that the specified Directories are exported and now available to be mounted by other ComputerSystems. When exporting an entire FileSystem, the Directory should reference the topmost directory of the FileSystem.

The association provider support is one directional - from CIM_UnixLocalFileSystem to Directory. It is possible to figure out which instance of CIM_UnixLocalFileSystem exports instance(s) of Directories for mount. The provider supports the enumeration to offer a fast access to all exported Directories. This is similar to the command "cat /etc/exports".

The Server has three local File Systems mounted and exports one Directory for mount :

```
> mount
/dev/hda13         on /              type reiserfs     (rw)
/dev/hda5          on /boot          type ext2         (rw)
/dev/hda7          on /home          type ext2         (rw)
```

```
> cat /etc/exports
/home/pub         host.domain(rw,insecure,all_squash)
```

Association Provider : Linux_NFSExport

Reference A	Reference B
Directory = “Linux_Directory”	LocalFS = “CIM_UnixLocalFileSytem”

Instance Table

specified end	returned instance(s)
Linux_Directory “?”	- (NOT SUPPORTED)
Linux_Ext2FileSystem “/dev/hda5”	-
Linux_Ext2FileSystem “/dev/hda7”	Linux_Directory “/home/pub”
Linux_ReiserFileSystem “/dev/hda13”	-

3.5 File System Table Manipulation

The current schema definition allows to view the operational state regarding the mounted FileSystems. Only these file systems are known to the system and accessible via CIM / CIMOM, which exist in “/etc/fstab” and all actual mounted FileSystems (“/etc/mtab”). To offer write access to the “/etc/fstab”, the schema needs to be enhanced.

The “/etc/fstab” file contains the total amount of possible mount entries.

Linux_FSTABConf : CIM_UnixSysConf
... represents the '/etc/fstab' file and is used to collect the Linux_FSTABSetting instances .

Linux_FSTABSetting : CIM_UnixSetting
... represents the entries in the '/etc/fstab' file . Each line is mapped to one instance of this class .

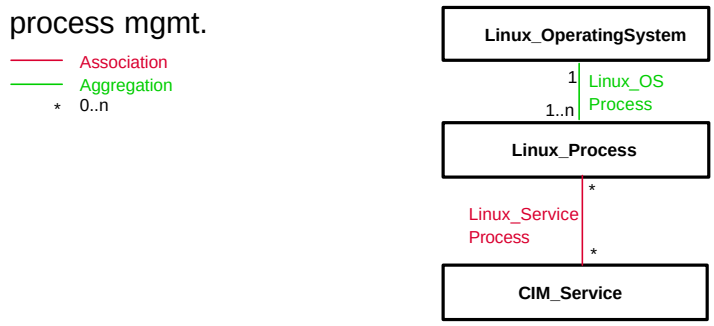
Linux_FSTABSettingContext : CIM_UnixSettingContext
... associates the Linux_FSTABSetting(s) with the Linux_FSTABConf instance .

Linux_FSTABConfForOS : CIM_ElementConfiguration
... associates the instance of Linux_FSTABConf with the running instance of Linux_OperatingSystem .

Linux_FSTABSettingForOS : CIM_ElementSetting
... associates the instance of Linux_FSTABSetting(s) with the running instance of Linux_OperatingSystem .

4 Process & Service Management

The necessary CIM schema parts for Process Management are Core, Unix and System. Appendix C "Process / Service Mgmt." shows the CIM hierarchy and the parents of the classes / associations.



4.1 Process Schema

4.1 Class description

4.1.1 Linux_Process : CIM_UnixProcess

... represents instances of running programs. A user of the Operating System will typically see a Process as an application or task. Within an OperatingSystem, a Process is defined by a workspace of memory resources and environmental settings that are allocated to it. On a multitasking System, this workspace prevents intrusion of resources by other Processes. Additionally, a Process can execute as multiple Threads, all which run within the same workspace.

An enumeration is similar to the "ps" command. Linux represents Threads as Processes ! The Linux_Process provider returns a mix of processes and threads.

Class Provider : Linux_Process

Name	Type	CIM class	Value / Description	Impl.
Caption	string	ManagedElement	short caption to describe the class	x
Description	string	ManagedElement	short description of this class	x
Name	string	ManagedSystemElement	Process name	x
Status	string	ManagedSystemElement	current Status of the System : "OK"	x
InstallDate	datetime	ManagedSystemElement		-
CSCreationClassName	string (key)	Process	"Linux_UnitaryComputerSystem"	x
CSName	string (key)	Process	full qualified hostname "host.domain"	x
OSCreationClassName	string (key)	Process	"Linux_OperatingSystem"	x
OSName	string (key)	Process	Linux and the release number, e.g. "Linux 2.4.10"	x
CreationClassName	string (key)	Process	"Linux_Process"	x
Handle	string (key)	Process	Process PID	x
CreationDate	datetime	Process	Process started execution	x (-)
TerminationDate	datetime	Process	Process was stopped / terminated; only applicable when ExecutionState = 4 (Blocked)	-
ExecutionState	uint16	Process	Supported values : Running (3), Blocked (4), Suspended Ready (6), Stopped (7), Terminated (8)	x
OtherExecutionDescription	string	Process	Additional information if ExecutionState = Other(1)	-
Priority	uint32	Process	0 (highest) … 40 (lowest)	x
KernelModeTime	uint64	Process	Time of process in kernel mode in ms	x (-)
UserModeTime	uint64	Process	Time of process in user mode in ms	x (-)
WorkingSetSize	uint64	Process	Memory in bytes, the process needs to efficient execute	-
RealUserID	uint64	UnixProcess	Real user ID of the process	x
ParentProcessID	string	UnixProcess	the parent's process Process ID	x
ProcessGroupID	uint64	UnixProcess	Group ID of this process	x
ProcessSessionID	uint64	UnixProcess	If part of a group of processes are under the control of a session leader, this property holds the session ID for the group.	x

ProcessNiceValue	uint32	UnixProcess	'nice' value of this process; used to compute the priority	x
ProcessTTY	string	UnixProcess	the TTY currently associated with this process	x
ModulePath	string	UnixProcess	Executing process's command path	x
Parameters[]	string	UnixProcess	The operating system parameters provided to the executing process. These are the argv[] values.	x (-)
ProcessWaitingForEvent	string	UnixProcess	A description of the event this process is currently sleeping for. Only used when status is 'Blocked' or 'SuspendedBlocked'	-

4.1.2 Linux_Service : CIM_Service

... represents instances of a service. A Service is a general-purpose object to configure and manage the implementation of functionality. It is not the functionality itself.

An enumeration lists /etc/init.d scripts (services).

Class Provider : Linux_Service

Name	Type	CIM class	Value / Description	Impl.
Caption	string	ManagedElement	short caption to describe the class	x
Description	string	ManagedElement	short description of this class	x
Name	string (key)	ManagedSystemElement	Unique name of the service	x
Status	string	ManagedSystemElement	current Status of the System : "OK"	x
InstallDate	datetime	ManagedSystemElement		-
SystemCreationClassName	string (key)	Service	"Linux_UnitaryComputerSystem"	x
SystemName	string (key)	Service	full qualified hostname "host.domain"	x
CreationClassName	string (key)	Service	"Linux_Service"	x
Started	boolean	Service	Service is started (TRUE) or stopped (FALSE)	x
StartMode	string	Service	Automatically started or on request	x

4.2 Association description

4.2.1 Linux_OSProcess : CIM_OSProcess

... is a link between the OperatingSystem and Process(es) running in the context of this OperatingSystem.

Association Provider : Linux_OSProcess

Reference A	Reference B
GroupComponent = "Linux_OperatingSystem"	PartComponent = "Linux_Process"

Instance Table

specified end	returned instance(s)
Linux_OperatingSystem "Linux 2.4.10"	… Linux_Process "25" … Linux_Process "317" … Linux_Process "945" …
Linux_Process "25"	Linux_OperatingSystem "Linux 2.4.10"
Linux_Process "945"	Linux_OperatingSystem "Linux 2.4.10"
…	…

4.2.2 Linux_ServiceProcess : CIM_ServiceProcess

... is used to establish relationships between Services and Processes . It is used to indicate if a Service is running in a particular Process . It is also used to indicate , via the ExecutionType property , if the Service started and is wholly responsible for the Process , or if the Service is running in an existing Process , perhaps with other unrelated Services , which is owned or started by a different entity . This association relates a Service with a externally visible system signature .

A deeper specification how the provider works depends on the supported Service . This will be specified when modelling services .

Association Provider : Linux_ServiceProcess

Reference A	Reference B
Process = “Linux_Process”	Service = “Linux_Service”

Instance Table

specified end	returned instance(s)
Linux_Process “25”	Linux_Service “httpd”
Linux_Process “945”	Linux_Service “sendmail”
Linux_Service “sendmail”	… Linux_Process “945” Linux_Process “946” …
…	…

5 ToDo's

Definition of schemata for the following Management tasks :

- ⇨ Cluster Management
- ⇨ syslog / Event Management
- ⇨ User & Security Management (e.g. Account , Group , ...)
- ⇨ Network Connectivity
- ⇨ Hardware Management (e.g. Processor , ...)
- ⇨ Software Management
- ⇨ Kernel Parameters
- ⇨ Printers
- ⇨ Drivers

Any comments / suggestions / additions are welcome !

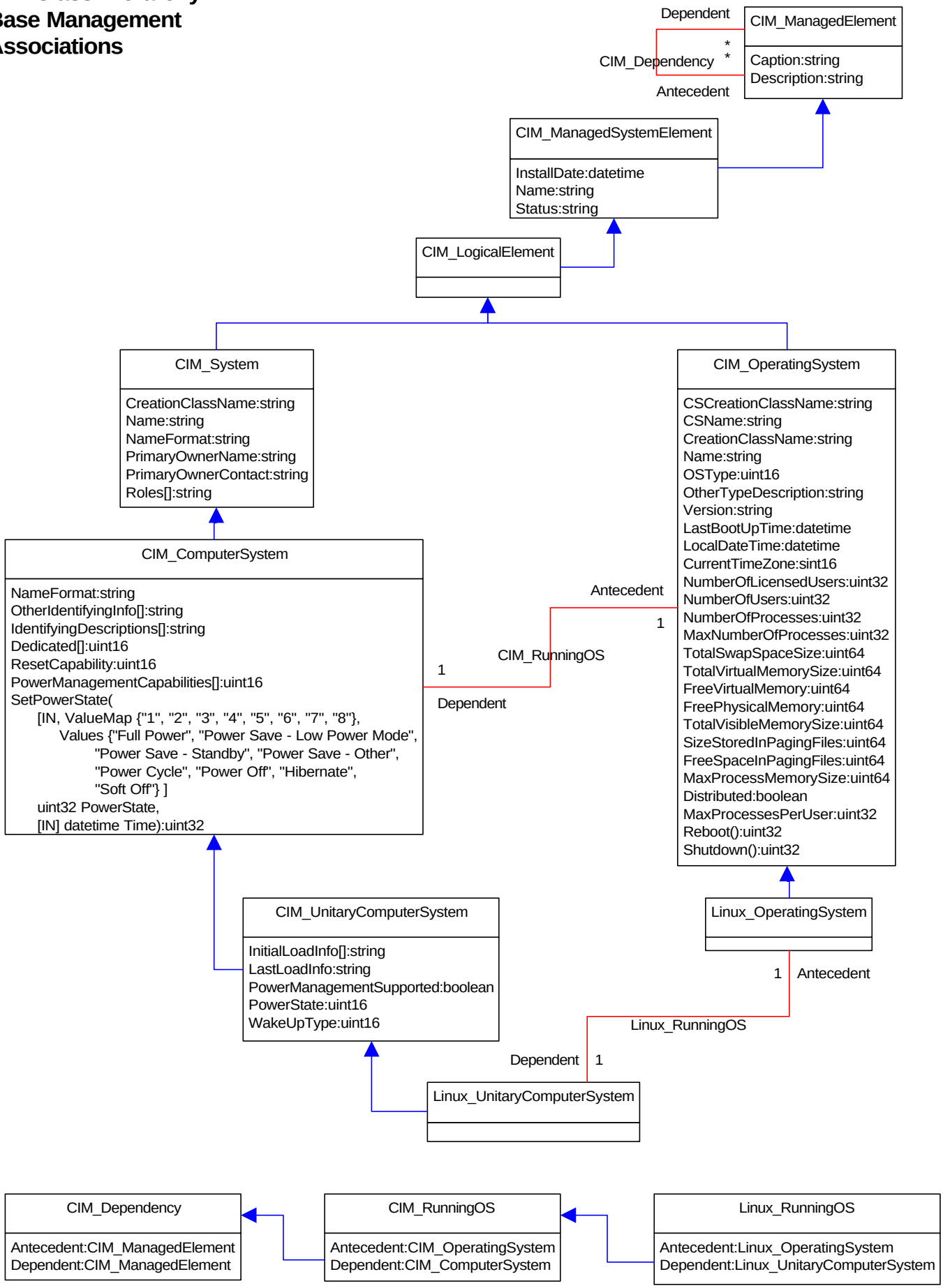
6 Instrumentation Overview

The following table list all classes and associations described in this paper.

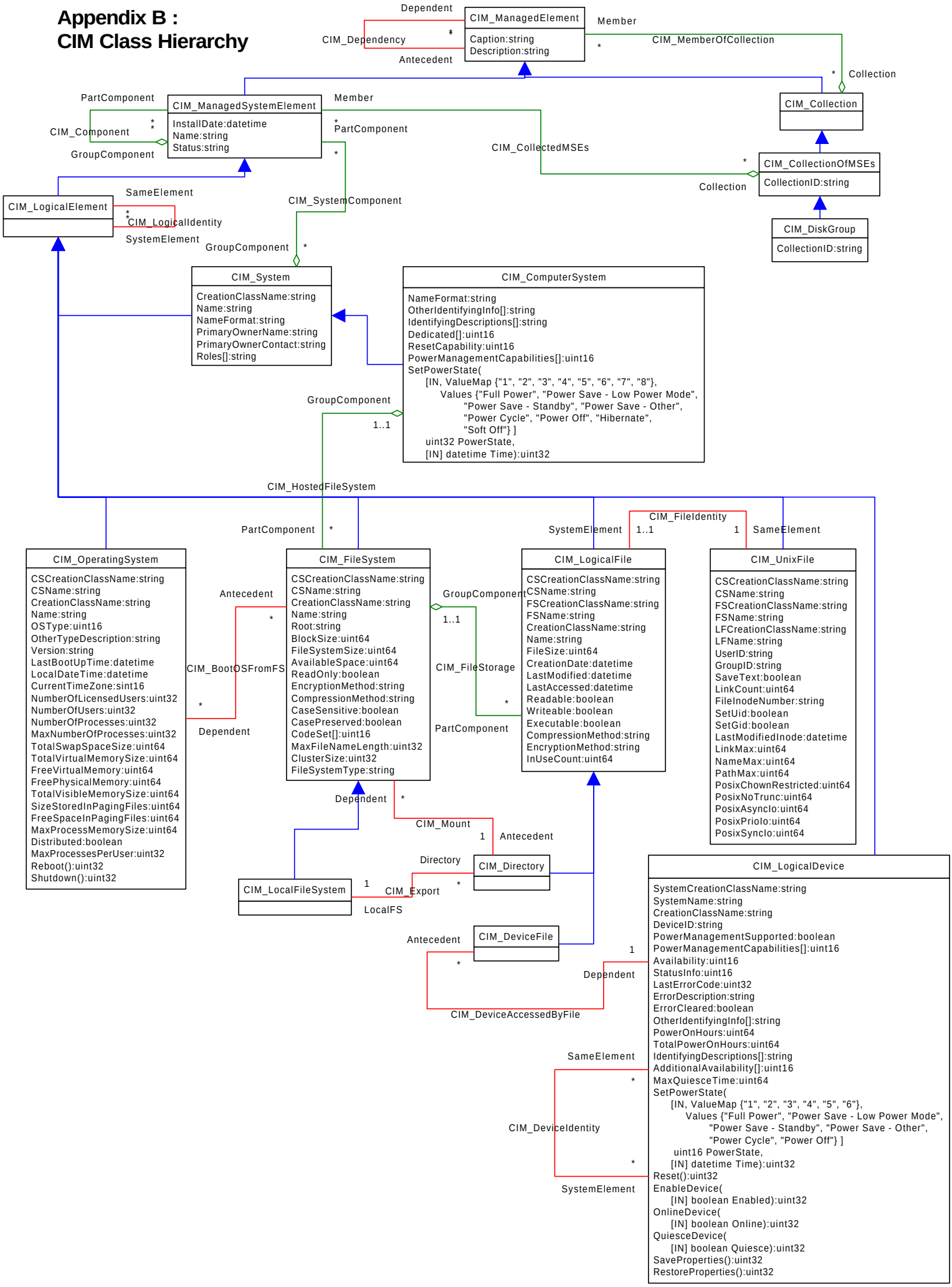
Name	*Instance*	*Provider*	*Association*	*Provider*	*Method*	*Provider*	*Indication*	*Provider*
	supported	implemented	supported	implemented	supported	implemented	supported	implemented
2 Base Management								
Linux_UnitaryComputerSystem	x	x	-	-				
Linux_OperatingSystem	x	x	-	-	x	x		
Linux_RunningOS	-	-	x	x				
3 File System & Volume Mgmt								
Linux_LocalFileSystem	x	x	-	-				
Linux_Directory	x	x	-	-				
Linux_DeviceFile	x		-	-				
Linux_UnixFile	x		-	-				
Linux_DataFile	x		-	-				
Linux_NativeLogicalDisk	x	x	-	-	x	x		
Linux_HostedFileSystem	x	x	x	x				
Linux_BootOSFromFS	-	-	x	x				
Linux_Mount	x	x	x	x				
Linux_ResidesOnExtent	x	x	x	x				
Linux_DeviceAccessedByFile	-	-	x					
Linux_FileIdentity	-	-	x					
Linux_FileStorage	x		x					
Linux_DiskPartition	x	x	-	-				
Linux_StorageVolume	x	x	-	-				
Linux_LogicalDiskBasedOnPartition	-	-	x	x				
Linux_DiskPartitionBasedOnVolume	-	-	x	x				
Linux_LVMLogicalDisk	x	x	-	-				
Linux_LVMLogicalVolume	x	x	-	-	x	x		
Linux_LVMDiskPartition	x	x	-	-				
Linux_LVMVolumeGroup	x	x	-	-	x	x		
Linux_LVMLVInVG	x	x	x	x				
Linux_LVMDPInVG	x	x	x	x				
Linux_LVMLVBasedOnDP	x	x	x	x				
Linux_LVMLogicalDiskBasedOnLV	-	-	x	x				
Linux_LVMDPIsDiskPartition	-	-	x	x				
Linux_NFS	x	x	-	-				
Linux_NFSExport	x	x	x	x				

Name	*Instance*	*Provider*	*Association*	*Provider*	*Method*	*Provider*	*Indication*	*Provider*
	supported	implemented	supported	implemented	supported	implemented	supported	implemented
Linux_FSTABConf	x		-	-				
Linux_FSTABSetting	x		-	-				
Linux_FSTABSettingContext	x		x					
Linux_FSTABConfForOS	-		x					
Linux_FSTABSettingForOS	x		x					
4 Process & Service Management								
Linux_Process	x	x	-	-				
Linux_ServiceProcess	x	x	x	x				
Linux_OSProcess	x	x	x	x				
Linux_Service	x	x			x	x		

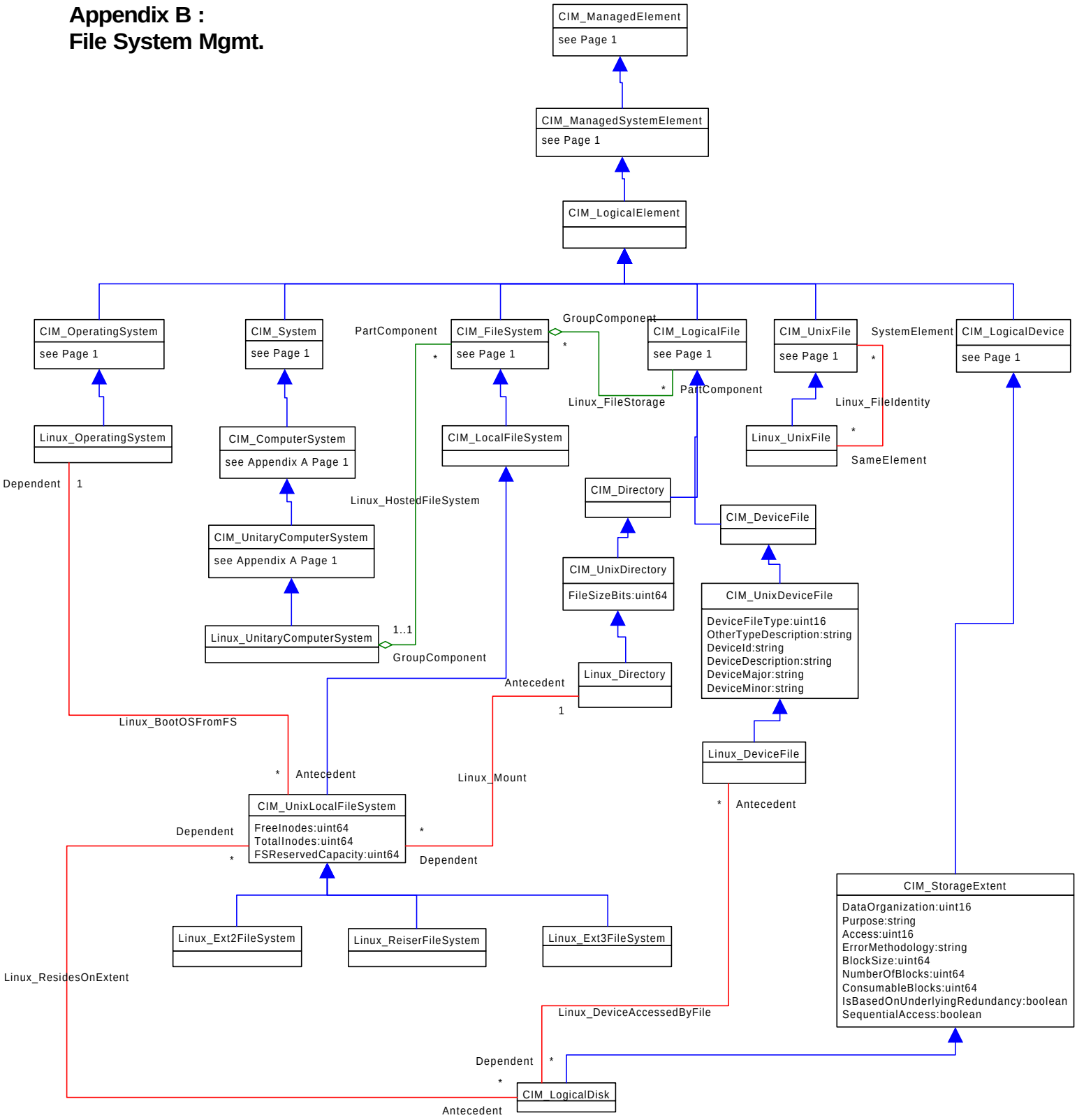
Appendix A :
CIM Class Hierarchy
Base Management
Associations



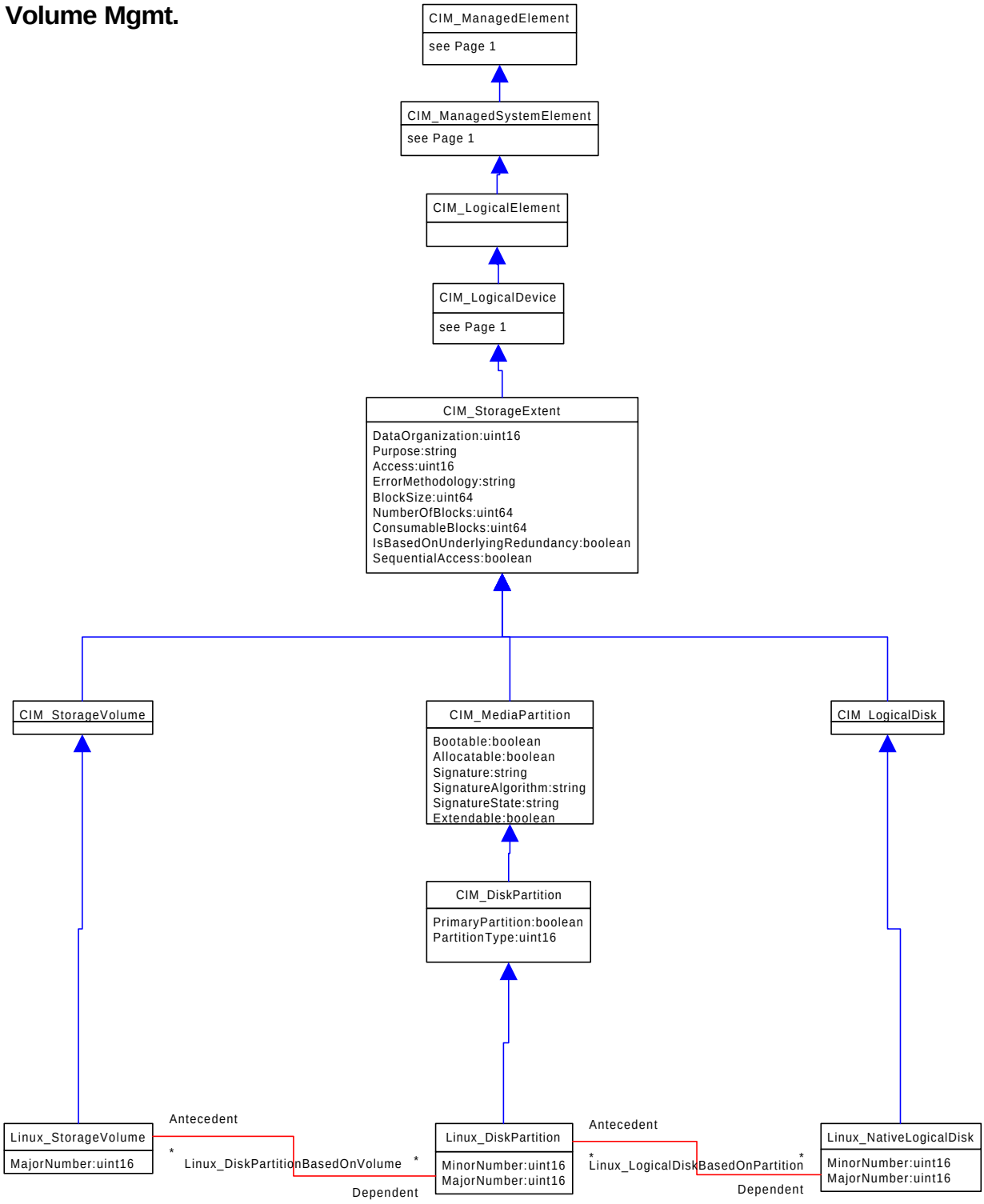
Appendix B :
CIM Class Hierarchy



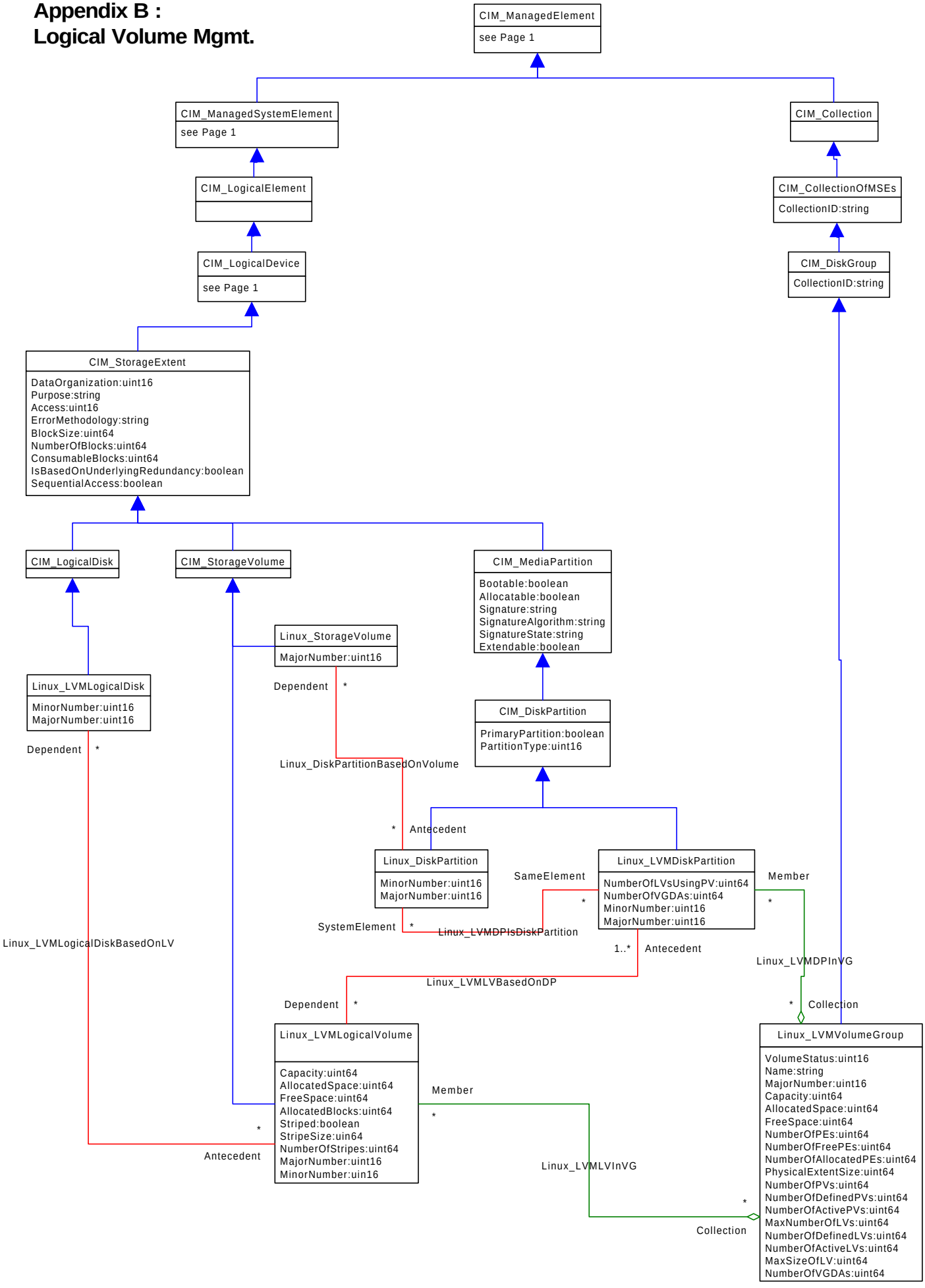
Appendix B :
File System Mgmt.



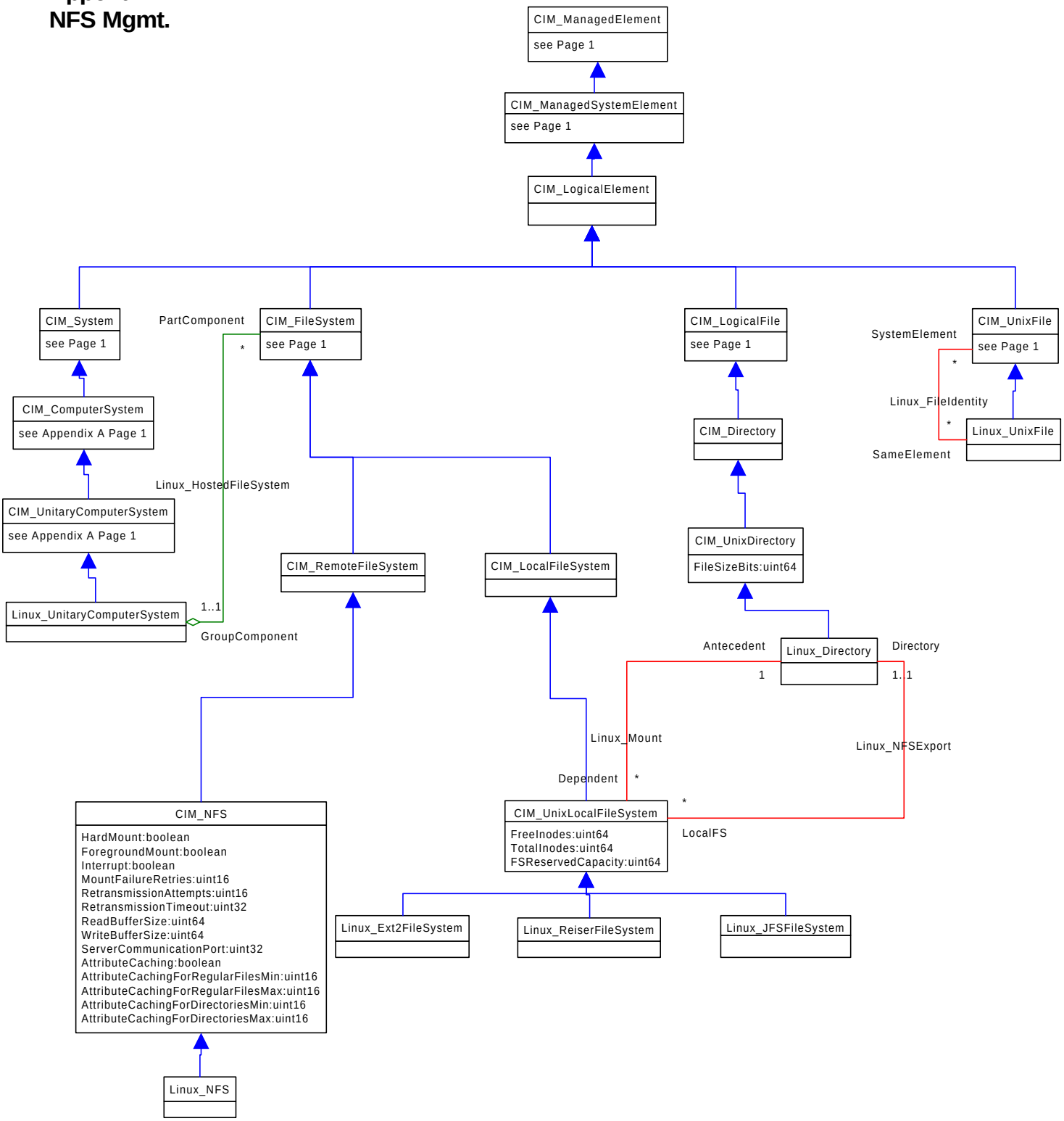
Appendix B :
Native Volume Mgmt.



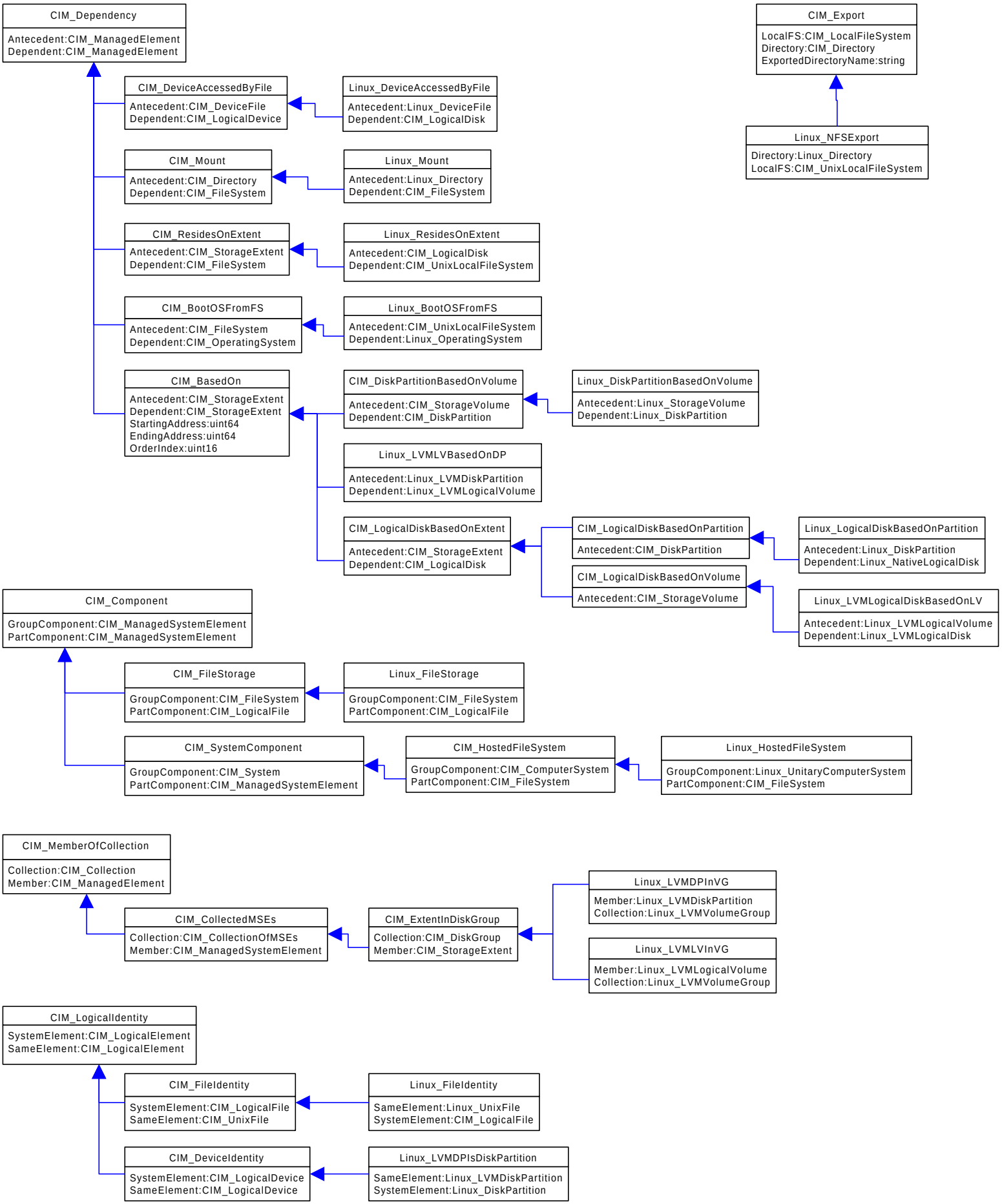
Appendix B :
Logical Volume Mgmt.



Appendix B :
NFS Mgmt.



Appendix B :
Association Hierarchy



Appendix C :
Process / Service Mgmt.