

Simple NIC Profile

Document Number: DCIM1032
Document Type: Specification
Document Status: Published
Document Language: E
Date: 2010-08-10

Version: 1.0.0



THIS PROFILE IS FOR INFORMATIONAL PURPOSES ONLY, AND MAY CONTAIN TYPOGRAPHICAL ERRORS AND TECHNICAL INACCURACIES. THE CONTENT IS PROVIDED AS IS, WITHOUT EXPRESS OR IMPLIED WARRANTIES OF ANY KIND. ABSENT A SEPARATE AGREEMENT BETWEEN YOU AND DELL™ WITH REGARD TO FEEDBACK TO DELL ON THIS PROFILE SPECIFICATION, YOU AGREE ANY FEEDBACK YOU PROVIDE TO DELL REGARDING THIS PROFILE SPECIFICATION WILL BE OWNED AND CAN BE FREELY USED BY DELL.

© 2010 Dell Inc. All rights reserved. Reproduction in any manner whatsoever without the express written permission of Dell, Inc. is strictly forbidden. For more information, contact Dell.

Dell and the *DELL* logo are trademarks of Dell Inc. *Microsoft* and *WinRM* are either trademarks or registered trademarks of Microsoft Corporation in the United States and/or other countries. Other trademarks and trade names may be used in this document to refer to either the entities claiming the marks and names or their products. Dell disclaims proprietary interest in the marks and names of others.

CONTENTS

1	Scope	5
2	Normative References.....	5
	2.1 Approved References	5
	2.2 Other References.....	5
3	Terms and Definitions	5
4	Symbols and Abbreviated Terms.....	7
5	Synopsis	7
6	Description.....	8
7	Implementation Requirements.....	9
	7.1 NIC View.....	9
	7.2 DCIM_NICEnumeration.....	11
	7.3 DCIM_NICString	13
	7.4 DCIM_NICInteger	15
	7.5 DCIM_NICService.....	18
	7.6 Simple NIC Profile Registration.....	19
8	Methods.....	20
	8.1 CIM_SimpleNICService.SetAttribute().....	20
	8.2 DCIM_NICService.SetAttributes()	21
	8.3 DCIM_NICService.CreateTargetedConfigJob()	22
	8.4 DCIM_NICService.DeletePendingConfiguration()	23
9	Use Cases	24
	9.1 Discovery of NIC profile support	24
	9.2 Inventory of NICs in system.....	25
	9.3 Get the first NIC's information.....	25
	9.4 List all NIC attributes	25
	9.5 Setting attributes.....	25
	9.6 Apply SetAttribute(s) pending values for a particular NIC	26
	9.7 Delete SetAttribute(s) pending values for a particular NIC	26
	ANNEX A (informative) Related MOF Files	27

Figures

Figure 1 – Simple NIC Profile: Class Diagram.....	8
---	---

Tables

Table 1 – Related Profiles.....	8
Table 2 – CIM Elements: Simple NIC Profile.....	9
Table 3 – DCIM_NICView - Operations.....	10
Table 4 – DCIM_NICView - Properties.....	10
Table 5 – DCIM_NICEnumeration - Operations.....	11
Table 6 – Class: DCIM_NICEnumeration.....	12
Table 7 – DCIM_NICEnumeration Attributes.....	13
Table 8 – DCIM_NICString - Operations.....	14
Table 9 – Class: DCIM_NICString.....	14
Table 10 – DCIM_NICString Attributes.....	15
Table 11 – DCIM_NICInteger - Operations.....	16
Table 12 – Class: DCIM_NICInteger.....	17
Table 13 – DCIM_NICInteger Attributes.....	18
Table 14 – DCIM_NICService – Operations.....	18
Table 15 – Class: DCIM_NICService.....	19
Table 16 – DCIM_LCRegisteredProfile - Operations.....	19
Table 17 – Class: CIM_RegisteredProfile.....	20
Table 18 – SetAttribute() Method: Return Code Values.....	20
Table 19 – SetAttribute() Method: Parameters.....	20
Table 20 – SetAttributes() Method: Return Code Values.....	21
Table 21 – SetAttributes() Method: Parameters.....	21
Table 22 – CreateTargetedConfigJob() Method: Return Code Values.....	22
Table 23 – CreateTargetedConfigJob() Method: Parameters.....	23
Table 24 – DeletePendingConfiguration() Method: Return Code Values.....	23
Table 25 – DeletePendingConfiguration() Method: Parameters.....	23

Simple NIC Profile

1 Scope

The Simple NIC Profile extends the management capabilities of referencing profiles by adding the capability to represent the configuration of NIC network controllers. The NIC controllers are modeled as views and attributes where there is a view for each individual controller and multiple attributes that allow configuration of the NIC.

2 Normative References

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

2.1 Approved References

DMTF DSP1033, *Profile Registration Profile 1.0.0*

DMTF DSP1061, *Management Profile 1.0.0*

DMTF DSP0200, *CIM Operations over HTTP 1.2.0*

DMTF DSP0004, *CIM Infrastructure Specification 2.3.0*

DMTF DSP1000, *Management Profile Specification Template*

DMTF DSP1001, *Management Profile Specification Usage Guide*

DMTF DSP0226, *Web Services for Management (WS-Management) Specification 1.1.0*

DMTF DSP0227, *WS-Management CIM Binding Specification 1.0.0*

2.2 Other References

ISO/IEC Directives, Part 2, *Rules for the structure and drafting of International Standards*, <http://isotc.iso.org/livelink/livelink.exe?func=ll&objId=4230456&objAction=browse&sort=subtype>

Unified Modeling Language (UML) from the Open Management Group (OMG), <http://www.uml.org>

3 Terms and Definitions

For the purposes of this document, the following terms and definitions apply.

3.1

can

used for statements of possibility and capability, whether material, physical, or causal

3.2

cannot

used for statements of possibility and capability, whether material, physical, or causal

3.3

conditional

indicates requirements to be followed strictly in order to conform to the document when the specified conditions are met

3.4

mandatory

indicates requirements to be followed strictly in order to conform to the document and from which no deviation is permitted

3.5

may

indicates a course of action permissible within the limits of the document

3.6

need not

indicates a course of action permissible within the limits of the document

3.7

optional

indicates a course of action permissible within the limits of the document

3.8

referencing profile

indicates a profile that owns the definition of this class and can include a reference to this profile in its "Related Profiles" table

3.9

shall

indicates requirements to be followed strictly in order to conform to the document and from which no deviation is permitted

3.10

shall not

indicates requirements to be followed strictly in order to conform to the document and from which no deviation is permitted

3.11

should

indicates that among several possibilities, one is recommended as particularly suitable, without mentioning or excluding others, or that a certain course of action is preferred but not necessarily required

3.12

should not

indicates that a certain possibility or course of action is deprecated but not prohibited

3.13

ENUMERATE

Refers to WS-MAN `ENUMERATE` operation as described in Section 8.2 of DSP0226_V1.1 and Section 9.1 of DSP0227_V1.0

3.14

GET

Refers to WS-MAN GET operation as defined in Section 7.3 of DSP00226_V1.1 and Section 7.1 of DSP0227_V1.0

4 Symbols and Abbreviated Terms

4.1

CIM

Common Information Model

4.2

iDRAC

integrated Dell Remote Access Controller – management controller for blades and monolithic servers

4.3

CMC

Chassis Management Controller – management controller for the modular chassis

4.4

iSCSI

Internet Small Computer System Interface, an Internet Protocol (IP)-based storage networking standard for linking data storage facilities.

4.5

WBEM

Web-Based Enterprise Management

5 Synopsis

Profile Name: Simple NIC

Version: 1.0.0

Organization: Dell Inc.

CIM Schema Version: 2.19.1

Central Class: DCIM_NICService

Scoping Class: CIM_ComputerSystem

The Simple NIC Profile extends the management capability of the referencing profiles by adding the capability to describe NIC controllers in a very simple fashion. In this profile a NIC is represented by a view instance that aggregates zero or more instances of the DCIM_NICAttribute class each representing a NIC controller related configurable property. DCIM_NICService shall be the Central Class. CIM_ComputerSystem shall be the Scoping Class. The instance of DCIM_NICService shall be the Central Instance. The instance of CIM_ComputerSystem with which the Central Instance is associated through the CIM_HostedService association shall be the Scoping Instance.

Table 1 identifies profiles that are related to this profile.

Table 1 – Related Profiles

Profile Name	Organization	Version	Relationship
Profile Registration Profile	DMTF	1.0	Mandatory

6 Description

The Simple NIC Profile describes NIC controller’s representation and configuration. The profile also describes the relationship of the Simple NIC classes to the DMTF/Dell profile version information.

Figure 1 represents the class schema for the Simple NIC Profile. For simplicity, the prefix CIM_ has been removed from the names of the classes.

The DCIM_NICView class is a NIC controller’s representation that contains controllers’ properties. The DCIM_NICAttribute class derives from the CIM_BIOSAttribute class and represents configurable NIC attributes. Each NIC’s configurable attribute is represented by DCIM_NICAttribute. Depending on the data type of the attribute, DCIM_NICAttribute is either instantiated as DCIM_NICEnumeration, DCIM_NICString, or DCIM_NICInteger instance. DCIM_NICView instance represents the NIC has most common properties of the NIC on them. The DCIM_NICService class is used to configure the NIC attributes. The SetAttribute() and SetAttributes() methods on the DCIM_NICService class configure NIC attributes, DCIM_NICAttribute subclass instances.

The Simple NIC profile information is represented with the instance of CIM_RegisteredProfile.

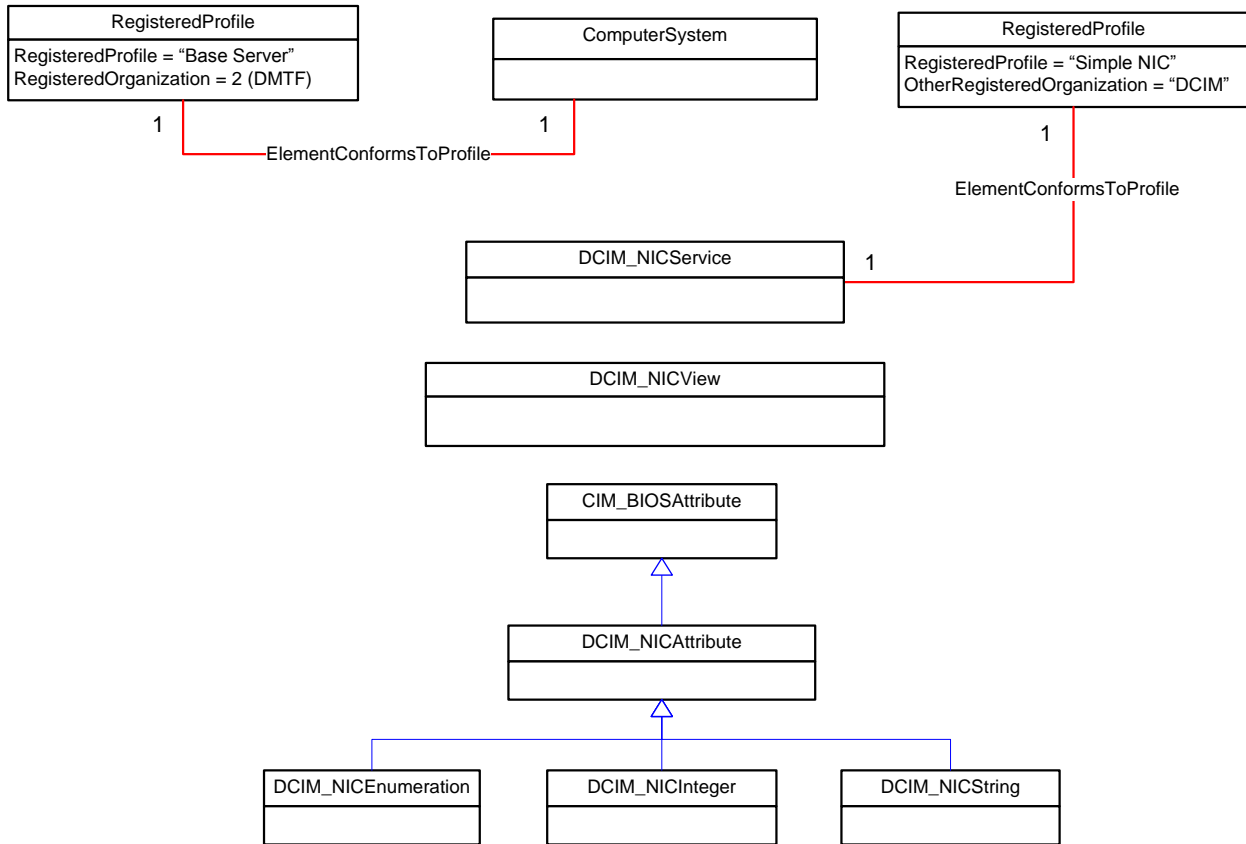


Figure 1 – Simple NIC Profile: Class Diagram

7 Implementation Requirements

Requirements and guidelines for propagating and formulating certain properties of the classes are discussed in this section. Methods are listed in section 8.

Table 2 shows the instances of CIM Elements for this profile. Instances of the CIM Elements shall be implemented as described in Table 2. Sections 7 (“Implementation Requirements” and “Methods”) may impose additional requirements on these elements.

Table 2 – CIM Elements: Simple NIC Profile

Element Name	Requirement	Description
Classes		
DCIM_NICService	Mandatory	The class maybe implemented in the Implementation Namespace. See sections 7.5
DCIM_NICView	Mandatory	The class shall be implemented in the Implementation Namespace. See section 7.1
DCIM_NICEnumeration	Mandatory	The class shall be implemented in the Implementation Namespace. See section 7.2
DCIM_NICInteger	Mandatory	The class shall be implemented in the Implementation Namespace. See section 7.4
DCIM_NICString	Mandatory	The class shall be implemented in the Implementation Namespace. See section 7.3
DCIM_LCElementConformsToProfile	Mandatory	The class shall be implemented in the Implementation Namespace.
DCIM_LCElementConformsToProfile	Mandatory	The class shall be implemented in the Interop Namespace.
DCIM_LCRegisteredProfile	Mandatory	The class shall be implemented in the Interop Namespace. See section 7.6
Indications		
None defined in this profile		

7.1 NIC View

This section describes the implementation for the DCIM_NICView class.

This class shall be instantiated in the Implementation Namespace.

7.1.1 WBEM URIs for WinRM®

The class WBEM URI shall be “http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_NICView?__cimnamespace=<Implementation Namespace>”

The key property shall be the InstanceID.

The instance WBEM URI for DCIM_NICView instance shall be:

“http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_NICView?__cimnamespace=<Implementation Namespace>+InstanceID=<FQDD>”

7.1.2 Operations

The following table details the implemented operations on DCIM_NICView.

Table 3 – DCIM_NICView - Operations

Operation Name	Requirements	Required Input
Get	Mandatory	Instance URI
Enumerate	Mandatory	Class URI

7.1.3 Properties

The following table details the implemented properties for DCIM_NICView instance representing a NIC in a system. The “Requirements” column shall denote the implementation requirement for the corresponding property. If the column “Property Name” matches the property name, the property either shall have the value denoted in the corresponding column “Additional Requirement”, or shall be implemented according to the requirements in the corresponding column “Additional Requirement”.

Table 4 – DCIM_NICView - Properties

Property Name	Requirement	Type	Requirement and description
InstanceID	Mandatory	string	The property value shall be the FQDD property value.
FQDD	Mandatory	string	A string containing the Fully Qualified Device Description a user-friendly name for the object.
BusNumber	Mandatory	uint8	The bus number where this PCI device resides
CurrentMACAddress	Mandatory	string	A string containing the current MAC address.
DataBusWidth	Mandatory	string	DataBusWidth of the PCI.
DeviceNumber	Mandatory	uint8	The device number assigned to this PCI device for this bus.
FunctionNumber	Mandatory	uint8	The function number for this PCI device
PCIDeviceID	Mandatory	string	The property contains a value assigned by the device manufacturer used to identify the type of device.
PCISubDeviceID	Mandatory	string	The property contains a value assigned by the vendor manufacturer used to identify the type of device.
PCISubVendorID	Mandatory	string	Subsystem vendor ID.
PCIVendorID	Mandatory	string	The property contains a value assigned by the PCI SIG used to identify the manufacturer of the device.
PermanentiSCSIMACAddress	Mandatory	string	PermanentiSCSIMACAddress defines the network address that is hardcoded into a port and dedicated to iSCSI usage.
PermanentMACAddress	Mandatory	string	PermanentMACAddress defines the network address that is hardcoded into a port.
ProductName	Mandatory	string	A string containing the product name
SlotLength	Mandatory	string	Slot length of the PCI.

SlotType	Mandatory	string	Slot type of the PCI.
LastSystemInventoryTime	Mandatory	string	This property provides the last time "\"System Inventory Collection On Reboot(CSIOR)\" was performed. The value is represented as yyyyymmddHHMMSS.
LastUpdateTime	Mandatory	string	This property provides the last time the data was updated. The value is represented as yyyyymmddHHMMSS

7.2 DCIM_NICEenumeration

This section describes the implementation for the DCIM_NICEenumeration class.

Each DCIM_NICEenumeration instance is logically associated to a DCIM_NICView instance, where the DCIM_NICEenumeration.FQDD property is equal to the FQDD property on the DCIM_NICView instance.

This class shall be instantiated in the Implementation Namespace.

7.2.1 WBEM URIs for WinRM®

The class WBEM URI shall be "http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_NICEenumeration?__cimnamespace=<Implementation Namespace>"

The key property shall be the InstanceID.

The instance WBEM URI for DCIM_NICEenumeration instance shall be:
 "http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_NICEenumeration?__cimnamespace=<Implementation Namespace>+InstanceID=<FQDD>:<AttributeName>"

7.2.2 Operations

The following table details the implemented operations on DCIM_NICEenumeration.

Table 5 – DCIM_NICEenumeration - Operations

Operation Name	Requirements	Required Input
Get	Mandatory	Instance URI
Enumerate	Mandatory	Class URI
DCIM_NICService.SetAttribute()	Mandatory	See section 8.1
DCIM_NICService.SetAttributes()	Mandatory	See section 8.2

7.2.3 Properties

The following table details the implemented properties for DCIM_NICEenumeration instance representing a NIC controller enumeration attribute. The "Requirements" column shall denote the implementation requirement for the corresponding property. If the column "Property Name" matches the property name, the property either shall have the value denoted in the corresponding column "Additional Requirement", or shall be implemented according to the requirements in the corresponding column "Additional Requirement".

Table 6 – Class: DCIM_NICEnumeration

Properties	Notes	Additional Requirements
InstanceID	Mandatory	The property value shall be formed as follows: “<FQDD property value>:<AttributeName property value>”.
AttributeName	Mandatory	The property value shall be from the “AttributeName” column in Table 7 –
CurrentValue	Mandatory	The property value shall be one of the values in the “PossibleValues” column at the corresponding row in Table 7 – .
PendingValue	Mandatory	The property value shall be one of the values in the “PossibleValues” column at the corresponding row in Table 7 – .
IsReadOnly	Mandatory	The property value shall be FALSE.
FQDD	Mandatory	FQDD of the NIC that the attribute belongs to.
PossibleValues	Mandatory	The property value shall be equal to the array of the values in “PossibleValues” column at the corresponding row in Table 7 – .

The following table describes the requirements for the AttributeName, and PossibleValues properties. The PossibleValues is an array property represented in the table as comma delimited list.

Table 7 – DCIM_NICEnumeration Attributes

AttributeName	Description	PossibleValues
TcpIpViaDHCP	Acquiring TCP/IP configuration via DHCP for iSCSI.	"Disabled", "Enabled"
IscsiViaDHCP	Acquire iSCSI parameters via DHCP.	"Disabled", "Enabled"
ChapAuthEnable	Enable/Disable CHAP authentication for iSCSI.	"Disabled", "Enabled"
IscsiTgtBoot	Enable/Disable booting to iSCSI target after logon.	"Disabled", "Enabled", "One Time Disabled"
TcpTimestmp	Enable/Disable TCP timestamp for iSCSI.	"Disabled", "Enabled"
FirstHddTarget	Enable/Disable target appears as first hard disk drive (HDD) in the system for iSCSI.	"Disabled", "Enabled"
IpVer	IP version support. Modifying this parameter will reset all IP-related fields for iSCSI.	"IPv4", "IPv6"
ConnectFirstTgt	Enable/Disable first target establishment for iSCSI.	"Disabled", "Enabled"
ConnectSecondTgt	Enable/Disable second target establishment for iSCSI.	"Disabled", "Enabled"
LegacyBootProto	Select non-UEFI Boot Protocol: Preboot Execution Environment (PXE)/iSCSI/NONE for Managed Boot Agent.	"PXE", "iSCSI", "NONE"
LnkSpeed	Configure link speed for Managed Boot Agent.	"AutoNeg", "10Mbps Half", "10Mbps Full", "100Mbps Half", "100Mbps Full"
WakeOnLan	Configure preboot Wake on LAN (WOL) for Managed Boot Agent..	"Disabled", "Enabled"
VLanMode	Configure virtual LAN mode for Managed Boot Agent.	"Disabled", "Enabled"
BootRetryCnt	Number of boot retries for Managed Boot Agent.	"No Retry", "1 Retry", "2 Retries", "3 Retries", "4 Retries", "5 Retries", "6 Retries", "Indefinite Retries"
UseIndTgtPortal	Use independent target portal when multipath I/O is enabled.	"Disabled", "Enabled"
UseIndTgtName	Use independent target name when multipath I/O is enabled.	"Disabled", "Enabled"

7.3 DCIM_NICString

This section describes the implementation for the DCIM_NICString class.

Each DCIM_NICString instance is logically associated to a DCIM_NICView instance, where the DCIM_NICString.FQDD property is equal to the FQDD property on the DCIM_NICView instance.

This class shall be instantiated in the Implementation Namespace.

7.3.1 WBEM URIs for WinRM®

The class WBEM URI shall be "http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_NICString?__cimnamespace=<Implementation Namespace>"

The key property shall be the InstanceID.

The instance WBEM URI for DCIM_NICString instance shall be:

"http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_NICString?__cimnamespace=<Implementation Namespace>+InstanceID= <FQDD>"

7.3.2 Operations

The following table details the implemented operations on DCIM_NICString.

Table 8 – DCIM_NICString - Operations

Operation Name	Requirements	Required Input
Get	Mandatory	Instance URI
Enumerate	Mandatory	Class URI
DCIM_NICService.SetAttribute()	Mandatory	See section 8.1
DCIM_NICService.SetAttributes()	Mandatory	See section 8.2

7.3.3 Properties

The following table details the implemented properties for DCIM_NICString instance representing a NIC controller string attribute. The “Requirements” column shall denote the implementation requirement for the corresponding property. If the column “Property Name” matches the property name, the property either shall have the value denoted in the corresponding column “Additional Requirement”, or shall be implemented according to the requirements in the corresponding column “Additional Requirement”.

Table 9 – Class: DCIM_NICString

Properties	Notes	Additional Requirements
InstanceID	Mandatory	The property value shall be formed as follows: “<FQDD property value>:<AttributeName property value>”.
AttributeName	Mandatory	The property value shall be from the “AttributeName” column in Table 10 – .
CurrentValue	Mandatory	The property value shall match the format described in “Value Expression” column at the corresponding row in Table 10 – .
PendingValue	Mandatory	The property value shall match the format described in “Value Expression” column at the corresponding row in Table 10 – .
IsReadOnly	Mandatory	The property value shall be the value in the “IsReadOnly” column at the corresponding row in Table 10 – .
FQDD	Mandatory	FQDD of the NIC that the attribute belongs to.
MinLength	Mandatory	The property value shall be the value in the “MinLength” column at the corresponding row in Table 10 – .
MaxLength	Mandatory	The property value shall be the value in the “MaxLength” column at the corresponding row in Table 10 – .

The following table describes possible DCIM_NICString attributes and the requirements for the AttributeName, MinLength, and MaxLength properties.

The AttributeValue shall be read-only if IsReadOnly property value from the corresponding row in the below table contains “TRUE”.

The AttributeValue string shall have equal or lower number of characters than the MaxLength property value from the corresponding row.

The AttributeValue string shall have equal or higher number of characters than the MinLength property value from the corresponding row.

The AttributeValue shall conform to the ValueExpression form from the corresponding row.

Table 10 – DCIM_NICString Attributes

AttributeName	Description	IsReadOnly	MinLength	MaxLength	Value Expression
ChipMdl	Chip Type/ Revision.	TRUE	0	0	String
MacAddr	Permanent MAC Address	TRUE	0	0	MAC Address
VirtMacAddr	Virtual MAC Address	TRUE	0	0	MAC Address
IscsiMacAddr	iSCSI MAC Address	TRUE	0	0	MAC Address
VirtIscsiMacAddr	Virtual iSCSI MAC Address	TRUE	0	0	MAC Address
DhcpVendId	iSCSI DHCP vendor ID (up to 32 bytes long)	FALSE	0	32	String
IscsiInitiatorIpAddr	iSCSI initiator IP address.	FALSE	2	39	IP Address
IscsiInitiatorSubnet	iSCSI initiator subnet mask.	FALSE	2	39	IP Address
IscsiInitiatorGateway	iSCSI initiator default gateway IP address.	FALSE	2	39	IP Address
IscsiInitiatorPrimDns	iSCSI initiator primary DNS IP address.	FALSE	2	39	IP Address
IscsiInitiatorSecDns	iSCSI initiator secondary DNS IP address.	FALSE	2	39	IP Address
IscsiInitiatorName	iSCSI initiator name.	FALSE	0	128	String
IscsiInitiatorChapId	iSCSI initiator CHAP ID.	FALSE	0	32	String
FirstTgtIpAddress	iSCSI first target IP address.	FALSE	2	39	IP Address
FirstTgtIscsiName	iSCSI first target name.	FALSE	0	128	String
FirstTgtChapId	iSCSI first target CHAP ID.	FALSE	0	32	String
SecondTgtIpAddress	iSCSI second target IP address.	FALSE	2	39	IP address
SecondTgtIscsiName	iSCSI second target name.	FALSE	0	128	String
SecondTgtChapId	iSCSI second target CHAP ID.	FALSE	0	32	String
SecondaryDeviceMac Addr	Secondary device MAC address.	FALSE	17	17	IP address

7.4 DCIM_NICInteger

This section describes the implementation for the DCIM_NICInteger class.

Each DCIM_NICInteger instance is logically associated to a DCIM_NICView instance, where the DCIM_NICInteger.FQDD property is equal to the FQDD property on the DCIM_NICView instance.

This class shall be instantiated in the Implementation Namespace.

7.4.1 WBEM URIs for WinRM®

The class WBEM URI shall be “http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_NICInteger?__cimnamespace=<Implementation Namespace>”

The key property shall be the InstanceID.

The instance WBEM URI for DCIM_NICInteger instance shall be:
“http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_NICInteger?__cimnamespace=<Implementation Namespace>+InstanceID= <FQDD>”

7.4.2 Operations

The following table details the implemented operations on DCIM_NICInteger.

Table 11 – DCIM_NICInteger - Operations

Operation Name	Requirements	Required Input
Get	Mandatory	Instance URI
Enumerate	Mandatory	Class URI
DCIM_NICService.SetAttribute()	Mandatory	See section 8.1
DCIM_NICService.SetAttributes()	Mandatory	See section 8.2

7.4.3 Properties

The following table details the implemented properties for DCIM_NICInteger instance representing a NIC controller integer attribute. The “Requirements” column shall denote the implementation requirement for the corresponding property. If the column “Property Name” matches the property name, the property either shall have the value denoted in the corresponding column “Additional Requirement”, or shall be implemented according to the requirements in the corresponding column “Additional Requirement”.

Table 12 – Class: DCIM_NICInteger

Properties	Notes	Additional Requirements
InstanceID	Mandatory	The property value shall be formed as follows: “<FQDD property value>:<AttributeName property value>”.
AttributeName	Mandatory	The property value shall be from the “AttributeName” column in Table 13
CurrentValue	Mandatory	The property value shall match the format described in “Value Expression” column at the corresponding row in Table 13.
PendingValue	Mandatory	The property value shall match the format described in “Value Expression” column at the corresponding row in Table 13.
IsReadOnly	Mandatory	The property value shall be the value in the “IsReadOnly” column at the corresponding row in Table 13.
FQDD	Mandatory	FQDD of the NIC that the attribute belongs to.
LowerBound	Mandatory	The property value shall be the value in the “LowerBound” column at the corresponding row in Table 13.
UpperBound	Mandatory	The property value shall be the value in the “UpperBound” column at the corresponding row in Table 13.

The following table describes possible DCIM_NICInteger attributes and the requirements for the AttributeName, IsReadOnly, LowerBound, and UpperBound properties.

The AttributeValue shall be read-only if IsReadOnly property value from the corresponding row in the below table contains “TRUE”.

The AttributeValue shall be equal or lower than the UpperBound property value from the corresponding row.

The AttributeValue shall be equal or higher than the LowerBound property value from the corresponding row.

Table 13 – DCIM_NICInteger Attributes

AttributeName	Description	IsReadOnly	LowerBound	UpperBound
BlnkLeds	Blink LEDs for a duration up to 15 seconds.	FALSE	0	15
LnkUpDelayTime	Configure link up delay time in seconds (0..255) for iSCSI.	FALSE	0	255
LunBusyRetryCnt	Configure number of retries in 2 second intervals when LUN is busy (0..60) for iSCSI.	FALSE	0	60
FirstTgtTcpPort	Configure target TCP port number (1..65535) for iSCSI first target.	FALSE	1	65535
FirstTgtBootLun	Configure target boot LUN number (0..255) for iSCSI first target.	FALSE	0	255
SecondTgtTcpPort	Configure target TCP port number (1..65535) for iSCSI second target.	FALSE	1	65535
SecondTgtBootLun	Configure target boot LUN number (0..255) for iSCSI second target.	FALSE	0	255

7.5 DCIM_NICService

This section describes the implementation for the DCIM_NICService class.

This class shall be instantiated in the Implementation Namespace.

The DCIM_LCElementConformsToProfile association(s)' ManagedElement property shall reference the DCIM_NICService instance(s).

7.5.1 WBEM URIs for WinRM®

The class WBEM URI shall be “http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_NICService?__cimnamespace=<Implementation Namespace>”

The key properties shall be the SystemCreationClassName, CreationClassName, SystemName, and Name.

The instance WBEM URI for DCIM_NICService instance shall be:

“http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_NICService?__cimnamespace=<Implementation Namespace>+SystemCreationClassName=DCIM_ComputerSystem+CreationClassName=DCIM_NICService+SystemName=DCIM:ComputerSystem+Name= DCIM:NICService”

7.5.2 Operations

The following table details the implemented operations on DCIM_NICService.

Table 14 – DCIM_NICService – Operations

Operation Name	Requirements	Required Input
Get	Mandatory	Instance URI
Enumerate	Mandatory	Class URI
Invoke	Mandatory	Instance URI

7.5.3 Properties

The following table details the implemented properties for DCIM_NICService instance representing a system in a system. The “Requirements” column shall denote the implementation requirement for the corresponding property. If the column “Property Name” matches the property name, the property either shall have the value denoted in the corresponding column “Additional Requirement”, or shall be implemented according to the requirements in the corresponding column “Additional Requirement”.

Table 15 – Class: DCIM_NICService

Properties and Methods	Requirement	Description
SystemCreationClassName	Mandatory	The property value shall be “DCIM_ComputerSystem”.
CreationClassName	Mandatory	The property value shall be “DCIM_NICService”.
SystemName	Mandatory	The property value shall be “DCIM:ComputerSystem”.
Name	Mandatory	The property value shall be “DCIM:NICService”

7.6 Simple NIC Profile Registration

This section describes the implementation for the DCIM_LCRegisteredProfile class.

This class shall be instantiated in the Interop Namespace.

The DCIM_ElementConformsToProfile association(s)’ ConformantStandard property shall reference the DCIM_LCRegisteredProfile instance.

7.6.1 WBEM URIs for WinRM®

The class WBEM URI shall be "http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_RegisteredProfile?__cimnamespace=<Interop Namespace>"

The key property shall be the InstanceID property.

The instance WBEM URI shall be: “http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_LCRegisteredProfile?__cimnamespace=<InteropNamespace>+InstanceID=DCIM:SimpleNIC:1.0.0”

7.6.2 Operations

The following table details the implemented operations on DCIM_SystemView.

Table 16 – DCIM_LCRegisteredProfile - Operations

Operation Name	Requirements	Required Input
Get	Mandatory	Instance URI
Enumerate	Mandatory	Class URI

7.6.3 Properties

The following table details the implemented properties for DCIM_LCRegisteredProfile instance representing Simple NIC Profile implementation. The “Requirements” column shall denote the implementation requirement for the corresponding property. If the column “Name” matches the property

name, the property either shall have the value denoted in the corresponding column “Additional Requirements”, or shall be implemented according to the requirements in the corresponding column “Additional Requirements”.

Table 17 – Class: CIM_RegisteredProfile

Properties	Requirement	Description
RegisteredName	Mandatory	This property shall have a value of “Simple NIC”.
RegisteredVersion	Mandatory	This property shall have a value of “1.0.0”.
RegisteredOrganization	Mandatory	This property shall have a value of 1 (Other).
OtherRegisteredOrganization	Mandatory	This property shall match “DCIM”

8 Methods

This section details the requirements for supporting intrinsic operations and extrinsic methods for the CIM elements defined by this profile.

8.1 CIM_SimpleNICService.SetAttribute()

The SetAttribute() method is used to set or change the value of a NIC attribute.

Invocation of the SetAttribute() method shall change the value of the DCIM_NICAttribute.CurrentValue or DCIM_NICAttribute.PendingValue property to the value specified by the AttributeValue parameter if the DCIM_NICAttribute.IsReadOnly property is FALSE. Invocation of this method when the DCIM_NICAttribute.IsReadOnly property is TRUE shall result in no change to the value of the DCIM_NICAttribute.CurrentValue property. The results of changing this value is described with the SetResult parameter.

Return code values for the SetAttribute() method are specified in Table 18 – and parameters are specified in Table 19 –. Invoking the SetAttribute() method multiple times can result in the earlier requests being overwritten or lost.

Table 18 – SetAttribute() Method: Return Code Values

Value	Description
0	Completed with no error
1	Not supported
2	Failed

Table 19 – SetAttribute() Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	Target	String	FQDD of the NIC
IN, REQ	AttributeName	String	Shall contain the AttributeName property value for the attribute to be modified.
IN, REQ	AttributeValue[]	String	Shall contain the desired attribute value. If the value is valid, the CurrentValue or PendingValue property of the specified attribute will be modified.

Qualifiers	Name	Type	Description/Values
OUT	SetResult	String	Returns: "Set CurrentValue property" when the attributes current value is set. "Set PendingValue" when the attributes pending value is set.
OUT	RebootRequired	String	Returns: "Yes" if reboot is required, "No" if reboot is not required.
OUT	MessageID	String	Error MessageID
OUT	Message	String	Error Message
OUT	MessageArguments[]	String	Error MessageArguments

8.2 DCIM_NICService.SetAttributes()

The SetAttributes() method is used to set or change the values of a group of attributes.

Invocation of the SetAttributes() method shall change the values of the DCIM_NICAttribute.CurrentValue or PendingValue properties that correspond to the names specified by the AttributeName parameter and the values specified by the AttributeValue parameter if the respective DCIM_NICAttribute.IsReadOnly property is FALSE. Invocation of this method when the respective DCIM_NICAttribute.IsReadOnly property is TRUE shall result in no change to the corresponding value of the DCIM_NICAttribute.CurrentValue property.

Return code values for the SetAttributes() method are specified in Table 20, and parameters are specified in Table 21 – .

Invoking the SetAttributes() method multiple times can result in the earlier requests being overwritten or lost.

Table 20 – SetAttributes() Method: Return Code Values

Value	Description
0	Completed with no error
1	Not supported
2	Failed

Table 21 – SetAttributes() Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	Target	String	FQDD of the NIC
IN, REQ	AttributeName[]	String	Shall contain the AttributeName property value for the attribute to be modified.
IN, REQ	AttributeValue[]	String	Shall contain the desired attribute values. If the value is valid, the CurrentValue or PendingValue property of the specified attribute will be modified.

Qualifiers	Name	Type	Description/Values
OUT	SetResult[]	String	Returns: "Set CurrentValue property" when the attributes current value is set. "Set PendingValue property" when the attributes pending value is set.
OUT	RebootRequired[]	String	Returns: "Yes" if reboot is required, "No" if reboot is not required.
OUT	MessageID[]	String	Error MessageID
OUT	Message[]	String	Error Message
OUT	MessageArguments[]	String	Error MessageArguments

8.3 DCIM_NICService.CreateTargetedConfigJob()

The CreateTargetedConfigJob() method is used to apply the pending values created by the SetAttribute and SetAttributes methods. The successful execution of this method creates a job for application of pending attribute values.

CreateTargetedConfigJob method supports the following optional input parameters

1. RebootJobType: when provided in the input parameters, creates a specific reboot job to "PowerCycle" or "Graceful Reboot without forced shutdown" or "Graceful Reboot with forced shutdown". This parameter only creates the RebootJob and does not schedule it.
2. ScheduledStartTime: When provided in the input parameters, schedules the "configuration job" and the optional "reboot job" at the specified start time. A special value of "TIME_NOW" schedules the job(s) immediately.
3. UntilTime: This parameter has a dependency on "ScheduledStartTime", together "ScheduledStartTime" and "UntilTime" define a time window for scheduling the job(s). Once scheduled, jobs will be executed within the time window.

If CreateTargetedConfigJob method is executed without the 3 optional parameters discussed above, then configuration job is created but not scheduled. However, this configuration job can be scheduled later using the DCIM_JobService.SetupJobQueue () method from the "Job Control Profile". DCIM_JobService.SetupJobQueue () can be executed to schedule several configuration jobs including the reboot job. Refer to "Job Control Profile" for more details.

Return code values for the CreateTargetedConfigJob() method are specified in Table 22, and parameters are specified in Table 23.

Subsequent calls to CreateTargetedConfigJob after the first CreateTargetedConfigJob will result in error until the first job is completed.

Table 22 – CreateTargetedConfigJob() Method: Return Code Values

Value	Description
0	Success
1	Not supported
2	Failed
4096	Job Created

Table 23 – CreateTargetedConfigJob() Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	Target	String	FQDD of the NIC
IN	RebootJobType	Uint16	Shall contain the requested reboot type: 1 - PowerCycle 2 - Graceful Reboot without forced shutdown 3 - Graceful Reboot with forced shutdown.
IN	ScheduledStartTime	String	Start time for the job execution in format: yyyyymmddhhmmss. The string "TIME_NOW" means immediate.
IN	UntilTime	String	End time for the job execution in format: yyyyymmddhhmmss. : If this parameter is not NULL, then ScheduledStartTime parameter shall also be specified.
OUT	Job	CIM_ConcreteJob REF	Reference to the newly created pending value application job.
OUT	MessageID	String	Error MessageID
OUT	Message	String	Error Message
OUT	MessageArguments[]	String	Error MessageArguments

8.4 DCIM_NICService.DeletePendingConfiguration()

The DeletePendingConfiguration() method is used to cancel the pending values created by the SetAttribute and SetAttributes methods. The DeletePendingConfiguration() method cancels the pending configuration changes made before the configuration job is created with CreateTargetedConfigJob(). This method only operates on the pending changes prior to CreateTargetedConfigJob() being called. After the configuration job is created, the pending changes can only be canceled by calling DeleteJobQueue() method in the Job Control profile.

Return code values for the DeletePendingConfiguration() method are specified in Table 24, and parameters are specified in Table 25.

Table 24 – DeletePendingConfiguration() Method: Return Code Values

Value	Description
0	Success
1	Not supported
2	Failed

Table 25 – DeletePendingConfiguration() Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	Target	String	FQDD of the NIC
OUT	MessageID	String	Error MessageID
OUT	Message	String	Error Message

Qualifiers	Name	Type	Description/Values
OUT	MessageArguments[]	String	Error MessageArguments

9 Use Cases

This section contains use cases for the Dell NIC Profile. For the general instance and class URI structure, see Section **Error! Reference source not found.** and Section **Error! Reference source not found.**, respectively.

Note that URIs in this section are in form of WBEM URIs for WinRM®.

9.1 Discovery of NIC profile support

Use one of the two procedures below to confirm the existence of NIC profile support

- A) GET the *DCIM_LCRegisteredProfile* instance using an *InstanceID* of DCIM:SimpleNIC:1.0.0. See section 3.14 for a definition of GET .

Instance URI:

http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/DCIM_LCRegisteredProfile?_cimnamespace=root/interop+InstanceID=DCIM:SimpleNIC:1.0.0

Results for the *InstanceID* of DCIM:SimpleNIC:1.0.0 shown below. If no instance is returned, the profile is not supported.

```
DCIM_LCRegisteredProfile
  AdvertiseTypeDescriptions = WS-Identify, Interop Namespace
  AdvertiseTypes = 1, 1
  InstanceID = DCIM:SimpleNIC:1.0.0
  OtherRegisteredOrganization = DCIM
  RegisteredName = Simple NIC
  RegisteredOrganization = 1
  RegisteredVersion = 1.0.0
```

- B) ENUMERATE the *CIM_RegisteredProfile* class. See section 3.13 for a definition of ENUMERATE .

Class URI:

http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_RegisteredProfile?_cimnamespace=root/interop

Then query the result for the following properties:

```
RegisteredName = Simple NIC, OtherRegisteredOrganization = DCIM, RegisteredVersion = 1.0.0
```


9.2 Inventory of NICs in system

ENUMERATE the *DCIM_NICView* class to view all available instances of the class

Class URI:

http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_NICView?_cimnamespace=root/dcim

The instance information of all available NICs will be returned

9.3 Get the first NIC's information

The URI for getting particular instance information is deterministic (i.e the *InstanceID* will be unique for each instance)

For the first NIC in the system, the instance URI will be:

http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_NICView?_cimnamespace=root/dcim+InstanceID=NIC.Embedded.1-1

The instance of *DCIM_NICView* that contains the information on the first NIC will be returned

9.4 List all NIC attributes

ENUMERATE the *DCIM_NICAttribute* class to view all available attributes and possible values of all NICs

Class URI:

http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_NICAttribute?_cimnamespace=root/dcim

9.5 Setting attributes

- A) ENUMERATE the *DCIM_NICAttribute* class as shown in section 9.4 and identify the applicable instance
- B) Confirm the *IsReadOnly* field is set to false
- C) To invoke the *SetAttribute()* or *SetAttributes()* method, extract the instance information from A) and construct the input parameters per Table 19 or Table 21 – *SetAttributes()*
Method: Parameters
- D) INVOKE the *SetAttribute()* or *SetAttributes()* method

Class URI:

http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_NICService?_cimnamespace=root/dcim

- E) Examine output parameters per Table 18 or Table 20.
- F) Apply the pending values, per section 9.6, using the FQDD obtained from C)
- G) Repeat A) to confirm successful execution of the method

9.6 Apply SetAttribute(s) pending values for a particular NIC

- A) To invoke the CreateTargetedConfigJob() method, construct input parameters per Table 23 and use the particular NIC's FQDD
- B) INVOKE CreateTargetedConfigJob() method
Class URI:
http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_NICService?_cimnamespace=root/dcim
- C) Query the status of the jobID output using the job control profile methods

9.7 Delete SetAttribute(s) pending values for a particular NIC

- A) To invoke the DeletePendingConfiguration() method, construct input parameters per Table 23 and use the particular NIC's FQDD
- B) INVOKE DeletePendingConfiguration() method
Class URI:
http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_NICService?_cimnamespace=root/dcim
- C) If return message indicates success per Table 24, no further action necessary

ANNEX A

(informative)

Related MOF Files

Dell Tech Center MOF Library:

<http://www.delltechcenter.com/page/DCIM.Library.MOF>

Related Managed Object Format (MOF) files:

DCIM_NICService

DCIM_NICView

DCIM_NICEnumeration

DCIM_NICInteger

DCIM_NICString

DCIM_LCElementConformsToProfile

DCIM_LCRegisteredProfile