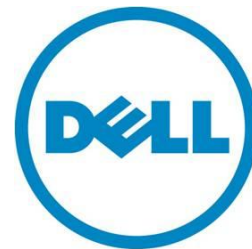

Remote NIC Configuration Management

This Dell Technical White Paper addresses the issue of remotely managing the NIC configuration from a source server to a destination server

Sankara Gara
Divya Vijayvargiya



This document 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.

© 2012 Dell Inc. All rights reserved. Dell and its affiliates cannot be responsible for errors or omissions in typography or photography. Dell, the Dell logo, and PowerEdge are trademarks of Dell Inc. Intel is a registered trademark of Intel Corporation in the U.S. and other countries. Microsoft, Windows, and Windows Server 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

Executive summary.....	4
Introduction.....	4
NIC configuration in general.....	4
Tools available prior to Lifecycle Controller.....	4
Advantages of Lifecycle Controller.....	4
Simple NIC Profile.....	5
Read NIC attributes and settings.....	6
Configure NIC attributes and settings.....	6
NIC configuration methods.....	7
Script work flows.....	7
Pull and Push.....	7
Pull and Diff with saved baseline.....	10
Conclusion.....	12
Learn more.....	12

Executive summary

Dell™ provides a mechanism that enables users to work with NIC configuration settings as a whole, rather than having to work with each setting individually. This enables end users to:

- Save a NIC configuration as baseline
- Compare a configuration to a saved baseline and display their differences
- Set a NIC configuration using a saved baseline

Introduction

In this white paper, we will focus on:

- NIC Management Profile
- Get and set NIC configuration
- Save a NIC configuration as baseline
- Compare a configuration to a saved baseline and display their differences
- Set a NIC configuration using a saved baseline

NIC configuration in general

NIC configuration allows the user to set up or modify configuration settings for supported NIC cards (e.g. Intel, Qlogic, and Broadcom). These configuration settings can be set or updated both locally and remotely using a server management system like the Dell Lifecycle Controller.

Tools available prior to Lifecycle Controller

- Locally on a system using F2 / System setup
- Vendor provided OptionRom
- Hiitest tool
- libsbios, an open source project (NOTE: compatible with Dell servers prior to 12G).

The above tools do not offer true programmatic interfaces for extension or modification of current supported behavior.

Advantages of Lifecycle Controller

Lifecycle Controller provides users with the ability to remotely deploy and manage a Dell PowerEdge™ server platform. Leveraging the embedded OS drivers and the iDRAC service processor, Lifecycle Controller aims at effectively resolving common pain points associated with platform provisioning and deployment.

Using Lifecycle Controller, users can:

- Deploy the operating system
- Manage BIOS and BOOT configuration
- Manage NICs/CNA configurations
- Manage vFlash SD cards

Remote NIC Configuration Management

- Manage RAID configurations
- Manage Auto-Discovery
- Manage certificates
- Perform remote updates
- Manage part replacement
- Use remote firmware inventory
- Retrieve hardware inventory
- View Lifecycle logs

Simple NIC Profile

The table below lists the classes, functions, operations, and methods available under the Simple NIC Profile.

Table 1. Simple NIC Profile

Class Name	Functions	Operations	Methods
DCIM_NICService	Use this central class to modify the NIC attributes.	Get Enumerate Invoke	
DCIM_NICEnumeration	Use this sub-class to display the properties of NIC enumeration instances.	Get Enumerate	SetAttribute() SetAttributes()
DCIM_NICInteger	Use this sub-class to display the properties of NIC integer instances.	Get Enumerate	SetAttribute() SetAttributes()
DCIM_NICString	Use this sub-class to display the properties of NIC string instances.	Get Enumerate	SetAttribute() SetAttributes()
DCIM_NICView	Use this sub-class to display the properties of NIC inventory instances	Get Enumerate	None

Read NIC attributes and settings

The enumerate operation is used to get a list of all the attributes of a given class. A Get operation is used to get a specific attribute of a given class. All NIC attribute instances carry base properties like *'AttributeName'*, *'CurrentValue'*, *'PendingValue'*, etc. In addition to this, each attribute also carries metadata such as *MinLength*, *MaxLength*, *LowerBound*, *UpperBound*, *ValueExpression*, *PossibleValues*, and etc., as appropriate to the attribute type. Attributes that are writable or settable have *'IsReadOnly'* property set to "false".

DCIM_NICEnumeration class represents all attributes that contain enumeration type values:

```
AttributeName = LinkStatus, CurrentValue = Connected  
  
PossibleValues = (Disconnected, Connected)
```

DCIM_NICInteger class represents all attributes that contain integer type value:

```
AttributeName=LnkUpDelayTime, CurrentValue=0, UpperBound=255,  
LowerBound=0
```

DCIM_NICString class represents all attributes that contain string type value:

```
AttributeName=VirtMacAddr, CurrentValue=" 78:2B:CB:54:57:23",  
MinLength=17, MaxLength=17
```

Refer to the "Simple NIC Profile" section of this paper for more details.

Configure NIC attributes and settings

NIC configuration uses the Set (pending values) and Apply (Commit) methodology for configuration. This methodology allows the user to schedule a configuration by staging changes and applying them (pending values) now or at a future time convenient to the user depending on the method parameters stipulated. The following steps are required for configuration:

1. Set one or more NIC attributes using `DCIM_NICService.SetAttribute()` or `DCIM_NICService.SetAttributes()`
2. Read the pending changes using enumeration commands
3. Schedule a job to apply the configuration using `DCIM_NICService.CreateTargetedConfigJob()`
4. Read the status of the job until it completes
5. Read again the updated values using enumeration commands

NIC configuration methods

- The `SetAttribute()` method is used to set or change the value of a NIC attribute.
- The `SetAttributes()` method is used to set or change the values of a group of attributes.
- 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.

Subsequent calls to the `CreateTargetedConfigJob()` method after the first `CreateTargetedConfigJob()` method results in an error until the first job is completed. However, the user can delete the current job and create a new job using `CreateTargetedConfigJob()`.

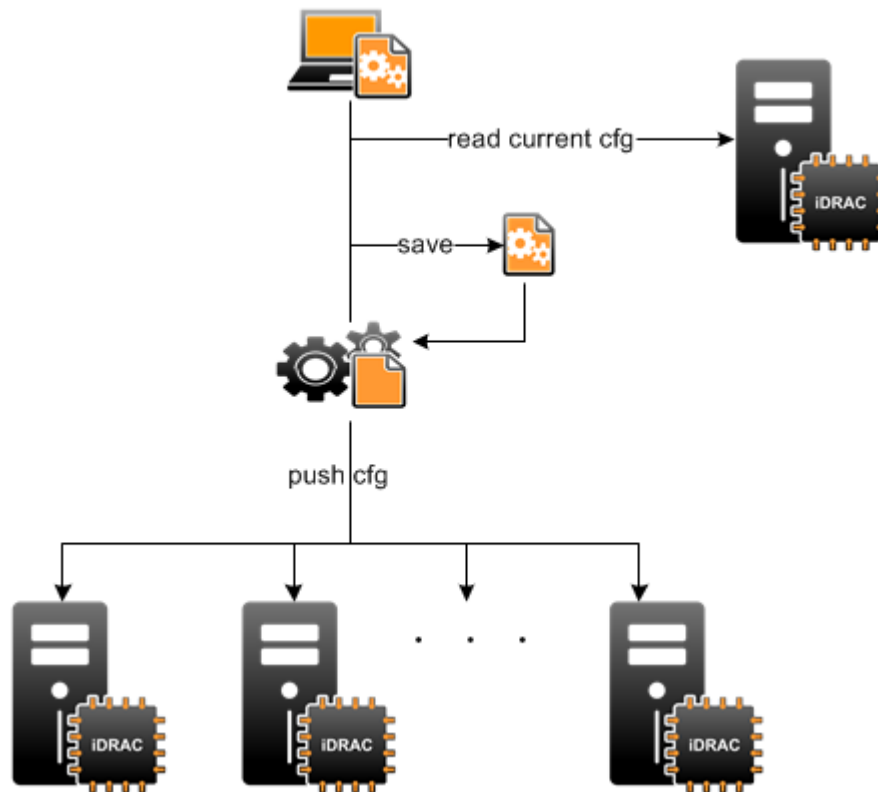
- The `DeletePendingConfiguration()` method cancels the pending configuration (created using the `SetAttribute` and `SetAttributes` methods) changes made before the configuration job is created with `CreateTargetedConfigJob()`.

Script work flows

Pull and Push

This use case demonstrates cloning of the NIC configuration using two-step process: Pull (read from Lifecycle Controller) followed by a Push (write to another Lifecycle Controller).

Figure 1. Work flow: Pull and Push



1. Pull the configuration from a baseline system.

- a. Configure the baseline system to the desired NIC configuration using remote services or any other existing NIC configuration methods.
- b. Pull the NIC configuration from the baseline system using the following command:

```
python ./getnic.py  
USAGE: getnic.py <IPADDRESS> <USERNAME> <USERPASSWORD> (bios | nic)
```

```
python ./getnic.py 10.210.134.33 root calvin nic
```

```
Building the Order Attributes Template File...
```

```
Getting the Attributes ....
```

```
Creating the .ini file ....
```

```
Consolidated Attribute INI created : 10.210.134.33_nic_consolidated.ini
```


Remote NIC Configuration Management

The getnic python script internally enumerates on DCIM_NICEnumeration, DCIM_NICString, and DCIM_NICInteger to create a consolidated list of system configuration.

2. Push the configuration to one or more systems.

Push the baseline NIC configuration to a target system using the following command:

```
./setnic.py
USAGE: setnic.py <IPADDRESS> <USERNAME> <USERPASSWORD> <INIFILE> (bios | nic) Now
For example: python setnic.py 192.168.0.111 admin admin123 file.ini bios Now
```

```
./setnic.py 10.36.0.181 root calvin 10.210.134.33_nic_consolidated.ini nic Now
```

Now setting attributes...

```
|NIC.Integrated.1-1-1
  NIC015:AttributeValue cannot be changed for ReadOnly AttributeName ControllerBIOSVersion
  NIC015:AttributeValue cannot be changed for ReadOnly AttributeName FamilyVersion
|NIC.Integrated.1-4-1
  NIC015:AttributeValue cannot be changed for ReadOnly AttributeName ControllerBIOSVersion
  NIC015:AttributeValue cannot be changed for ReadOnly AttributeName FamilyVersion
|NIC.Integrated.1-2-1
  NIC015:AttributeValue cannot be changed for ReadOnly AttributeName ControllerBIOSVersion
  NIC015:AttributeValue cannot be changed for ReadOnly AttributeName FamilyVersion
|NIC.Integrated.1-3-1
  NIC015:AttributeValue cannot be changed for ReadOnly AttributeName ControllerBIOSVersion
  NIC015:AttributeValue cannot be changed for ReadOnly AttributeName FamilyVersion
```

```
Config job id:JID_925923696439 for:NIC.Integrated.1-1-1 has been created.
Config job id:JID_925923727168 for:NIC.Integrated.1-4-1 has been created.
Config job id:JID_925923759536 for:NIC.Integrated.1-2-1 has been created.
Setting up config job for NIC.Integrated.1-3-1
Config job id:JID_925923791815 for:NIC.Integrated.1-3-1 has been created.
```

Restarting target server to execute configuration jobs.

```
Checking job status:.....
Job:JID_925923696439 completed.

Job:JID_925923727168 completed.

Job:JID_925923759536 completed.

Job:JID_925923791815 completed.
All job executions complete.
Restarting server.
Checking for Remote Services Ready....
.....Server is Ready..
```

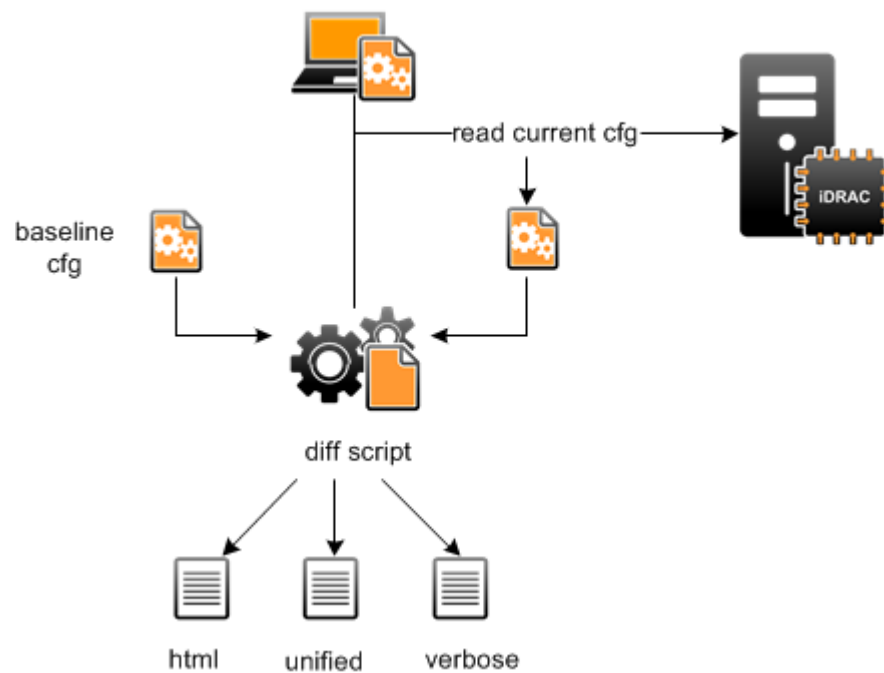
Remote NIC Configuration Management

The setnic script internally uses SetAttributes and CreateTargetedConfigJob to set changed values and create and execute a configuration job to update the target system configuration.

Pull and Diff with saved baseline

This use case demonstrates steps to view differences between NIC configurations of two different systems using a two-step process.

Figure 2. Work flow: Pull and Diff



1. Pull the NIC configuration from one of the systems using the following command:

```
python ./getnic.py 10.210.134.33 root calvin nic
```

```
Building the Order Attributes Template File...
```

```
Getting the Attributes ....
```

```
Creating the .ini file ....
```

```
Consolidated Attribute INI created : 10.210.134.33_nic_consolidated.ini
```

2. Diff the saved configuration with the second system using the following command:

```
./diffattribs.py
```

```
USAGE: diffattribs.py <IPADDRESS> <USERNAME> <USERPASSWORD> <INIFILE> (bios | nic)
```

```
For example: python diffattribs.py 192.168.0.111 admin admin123 file.ini bios
```

Remote NIC Configuration Management

```
./diffattrs.py 10.36.0.181 root calvin 10.210.134.33_nic_consolidated.ini nic
```

```
NIC attributes retrieved.  
Creating the attributes file ....  
File Created: 10.36.0.181_nic.txt  
Adding Attributes to the file...  
Consolidated Attribute INI created : 10.36.0.181_nic.txt_consolidated.ini
```

```
Generating the diff attributes...  
Creating the html diff file: diffattrs.html  
Creating the unified format diff file: diffattrs_unified.txt  
Creating the verbose diff file: diffattrs.txt
```

```
*****  
Attributes That Are Different  
*****
```

```
List of Attributes not present on target but present in: 10.210.134.33_nic_consolidated.ini
```

```
-----  
None
```

```
*****  
List of Attributes present on target but not present in: 10.210.134.33_nic_consolidated.ini
```

```
-----  
None  
*****
```

```
*****  
Attributes That Are Same but have different values  
*****
```

Attribute	Target	10.210.134.33_nic_consolidated.ini
[NIC.Integrated.1-1-1]BusDeviceFunction	02:00:01	01:00:01
[NIC.Integrated.1-1-1]FamilyVersion	7.0.47	7.0.44
[NIC.Integrated.1-1-1]MacAddr	14:FE:B5:FF:B3:27	14:FE:B5:FF:B4:35
[NIC.Integrated.1-2-1]BusDeviceFunction	02:00:01	01:00:01
[NIC.Integrated.1-2-1]FamilyVersion	7.0.47	7.0.44
[NIC.Integrated.1-2-1]MacAddr	14:FE:B5:FF:B3:27	14:FE:B5:FF:B4:35
[NIC.Integrated.1-3-1]BusDeviceFunction	02:00:01	01:00:01
[NIC.Integrated.1-3-1]FamilyVersion	7.0.47	7.0.44
[NIC.Integrated.1-3-1]MacAddr	14:FE:B5:FF:B3:27	14:FE:B5:FF:B4:35
[NIC.Integrated.1-4-1]BusDeviceFunction	02:00:01	01:00:01
[NIC.Integrated.1-4-1]FamilyVersion	7.0.47	7.0.44
[NIC.Integrated.1-4-1]MacAddr	14:FE:B5:FF:B3:27	14:FE:B5:FF:B4:35

```
*****  
Attributes That Are Same  
*****
```

Attribute	Value
[NIC.Integrated.1-1-1]BlinkLeds	0
[NIC.Integrated.1-1-1]ChapAuthEnable	Enabled
[NIC.Integrated.1-2-1]BlinkLeds	0
[NIC.Integrated.1-2-1]ChapAuthEnable	Enabled
[NIC.Integrated.1-3-1]BlinkLeds	0
[NIC.Integrated.1-3-1]ChapAuthEnable	Enabled
[NIC.Integrated.1-4-1]BlinkLeds	0
[NIC.Integrated.1-4-1]ChapAuthEnable	Enabled

```
*****
```

Output of the diff command will be available in text file and html file formats.

Conclusion

The remote service functionality that allows the user to save a particular NIC configuration, clone that NIC configuration on another server with an identical NIC card and firmware, and view the difference between the saved configuration and one from another server enables a system configuration handler to easily view and set up a system with a target configuration. The ability to view the differences between the two configurations provides the flexibility to individually set the specific configuration details, if required, instead of cloning the entire configuration.

Learn more

Visit Dell.com/PowerEdge for more information on Dell's enterprise-class servers.

WSMAN Interface Guide for Linux:

http://attachments.wetpaintserv.us/BMJk79WsVP3F0jwI50xR_w2088275

WSMAN Interface Guide for Windows:

http://attachments.wetpaintserv.us/utYVFQFaHmnfG_LHEnx1YQ2026735

WSMAN command line open source for Linux (Openwsman):

<http://sourceforge.net/projects/openwsman/>

WSMAN command line for Windows (Winrm):

[http://msdn.microsoft.com/en-us/library/windows/desktop/aa384291\(v=VS.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/aa384291(v=VS.85).aspx)

Scripting the Dell Lifecycle Controller:

<http://en.community.dell.com/techcenter/systems-management/w/wiki/scripting-the-dell-lifecycle-controller.aspx>

All about Lifecycle Controller in iDRAC:

http://support.dell.com/support/edocs/software/smusc/smlc/lc_1_5/index.htm