Zero-touch, bare-metal server provisioning by using the Dell EMC iDRAC with Lifecycle Controller Auto Config feature

This technical white paper provides an overview of the Auto Config feature available in the iDRAC with Lifecycle Controller. Examples about configuring Auto Config by using graphical user interface (GUI), RACADM, Redfish, and WS-Man commands are discussed. A list of tentative issues and resolutions is also provided.

Dell Engineering
November 2017

Authors (Dell EMC Server Solutions)

Doug Iler, iDRAC Product Manager
Sanjay Krishna, Firmware-Embedded Solutions
Trey Ramsay, Firmware-Embedded Solutions
Texas Roemer, SW Validation Test
Paul Rubin, Lifecycle Controller Product Manager
David Schmidt, Embedded Management Marketing Director
Virender Sharma, SW Validation Test
Sheshadri PR Rao, (InfoDev)
## Revisions

<table>
<thead>
<tr>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>June 2017</td>
<td>Initial release</td>
</tr>
<tr>
<td>November 2017</td>
<td>Corrected typos in note on page 38</td>
</tr>
</tbody>
</table>

The information in this publication is provided “as is.” Dell Inc. makes no representations or warranties of any kind with respect to the information in this publication, and specifically disclaims implied warranties of merchantability or fitness for a particular purpose.

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

Copyright © June –2017 Dell Inc. or its subsidiaries. All Rights Reserved. Dell, EMC, and other trademarks are trademarks of Dell Inc. or its subsidiaries. Other trademarks may be the property of their respective owners. Published in the USA [11/10/2017]

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

---

2 Zero-touch, bare-metal server provisioning by using the Dell EMC iDRAC with Lifecycle Controller Auto Config feature
Contents
Revisions..................................................................................................................2
Executive summary....................................................................................................4
  1.1 Components of the Auto Config feature in iDRAC with Lifecycle Controller ..........5
  1.1.1 Audience.....................................................................................................5
2 Introduction.............................................................................................................6
  2.1.1 Interactions between components in Auto Config architecture .........................7
  2.1.2 DHCP provisioning options.............................................................................7
  2.1.3 List of PowerEdge servers: 12th, 13th, and 14th generation ..............................8
  2.1.4 DHCP—current and new configuration options .............................................9
  2.1.5 Prerequisites for enabling Auto Config........................................................9
  2.1.6 iDRAC interfaces to enable Auto Config .......................................................9
  2.1.7 Enabling Auto Config by using RACADM (local or remote) ..........................9
  2.1.8 DHCP provisioning enable setting................................................................10
  2.2 Enabling Auto Config using WS-Man (winrm or openwsman) ........................11
3 Enabling Auto Config using iDRAC graphical user interface (GUI) .........................14
  3.1 Getting Server Configuration Profiles..................................................................22
  3.1.1 Getting server configuration profiles by using RACADM commands .............22
4 Getting server configuration profiles by using WS-Man commands .......................24
5 Configuring Windows and Linux DHCP servers to support Auto Config..................26
  5.1 Configuring Windows DHCP server option 43 ..................................................26
  5.2 Configuring Windows DHCP server option 60 ..................................................27
  5.3 Configuring the Linux DHCP server ..................................................................31
  5.4 File naming behavior with iDRAC firmware 2.20.20.20 or later ........................35
  5.4.1 RACADM workflow example using Auto Config server provisioning in a Linux environment..........................................................37
  5.5 WS-Man (winrm) workflow example using Auto Config server provisioning in Windows Server environment 42
6 Troubleshooting Auto Config issues .....................................................................50
7 Conclusion..............................................................................................................53
8 Glossary..................................................................................................................54
Executive summary

While the ever-growing requirement for reducing redundant processes and increasing availability of business-critical services is evident, it is always challenging for the system administrators and datacenter managers to scale out their IT environment. By using DHCP provisioning, the iDRAC with Lifecycle Controller provides the Auto Config feature to seamlessly and effectively onboard servers that are configured in extremely less time. Elaborated examples are provided to configure the Auto Config feature by using different interfaces, OSs, user options, server configuration profiles, and DHCP servers.

The Auto Config feature allows IT administrators to build an environment in which servers can automatically configure all hardware settings as part of the out-of-band network management. This eliminates the necessity of high-touch, manual steps to configure server subsystems such as storage, networking, and BIOS. Administrators can develop configuration profiles for classes of servers and apply those profiles without interacting with individual systems.

The increase in server density combined with the constraint of IT resources has made automated management capabilities a critical component of IT operations. Administrators require repeatable and scalable provisioning capabilities that eliminate error-prone and time-consuming manual processes. The iDRAC firmware for 12th, 13th, and 14th generation PowerEdge servers provides the Auto Config—a zero-touch mechanism—for configuring a bare-metal server from a common server configuration profile.

This technical white paper describes the requirements for using Auto Config, the setup procedures, and the tasks for monitoring the Auto Config workflow by providing detailed input and output examples.
1.1 Components of the Auto Config feature in iDRAC with Lifecycle Controller

![Diagram of Auto Config feature in iDRAC with Lifecycle Controller]

Figure 1  The Auto Config feature in iDRAC with Lifecycle Controller

1.1.1 Audience

This technical white paper is intended for server administrators, architects, and other stakeholders in decision making capacities. The reader is expected to have basic knowledge about server management applications and troubleshooting techniques.
Introduction

AutoConfig concept

Beginning with 12th generation PowerEdge servers, iDRAC with Lifecycle Controller uses Server Configuration Profiles to configure BIOS, iDRAC, PERC, and NIC or HBA settings in a single file.

A DHCP server assigns an IP address from its pool of IP addresses to any DHCP client that requests for an IP. The DHCP server can also be used to provide more information.

The iDRAC Auto Config feature integrates the Server Configuration Profile with the initial iDRAC DHCP process. Computers and other devices (such as iDRAC) commonly use a DHCP server to obtain an IP address. A DHCP server assigns an IP address from its pool of IP addresses to any DHCP client that requests one.

The DHCP server can also be used to provide more information in the form of DHCP server scope options to those clients requesting for that option. In case of Auto Config, the server scope options can specify the location of a Server Configuration Profile.

How Autoconfig works

The Auto Config feature requires an iDRAC Enterprise license. To maximize the full capability of this feature, servers must be ordered with the optional iDRAC DHCP option. With this option, Auto Config is enabled for the first DHCP request only.

When a server is started, iDRAC requests for and obtains an IP address from a DHCP server. iDRAC also requests a DHCP vendor scope option from the DHCP server. The DHCP server can be set up to provide information in the form of a string, along with the IP address, to the requesting iDRAC.

This “Vendor Scope Option” or string from the DHCP server contains a link to the Server Configuration Profile. The profile contains all the parameters that can be provisioned in the iDRAC, BIOS, PERC, and NIC subsystems of a PowerEdge server.

For more information on Server Configuration Profiles, see the Dell white papers on file structure, workflow, and XML schema.

Why Autoconfig for servers

Simplifies bare-metal deployments and scale-out operations by removing the step-by-step complexity of manual configuration.

Instead of manually interacting (pressing F2 or F10), or writing complex scripts, administrators can set up an initial ‘gold’ configuration server and capture the settings into a Server Configuration Profile file. This profile can be modified and then applied across an entire pool of servers.

This white paper describes the vendor scope option and string sent by the DHCP server to iDRAC and the details of the Server Configuration Profile and its location. This information is necessary for rapidly provisioning servers in the pool serviced by the DHCP server.

Starting with the design details of Auto Config, this paper provides specifics on configuring the DHCP server and concludes with detailed examples about using Auto Config for server deployment.

Figure 2 Importance of Auto Config in PowerEdge servers
2.1.1 Interactions between components in Auto Config architecture

The sequence of events performed on a single Dell server iDRAC interacting with the DHCP server for that pool is shown in the figure. This sequence is repeated for all Dell servers in the pool that are enabled for Auto Config.

![Diagram showing interactions between components in Auto Config architecture]

1. iDRAC sends request to the DHCP server
2. DHCP server provides IP address & option fields
3. iDRAC verifies option fields and retrieves Server Configuration profile from remote share
4. Server restart based on required changes

Figure 3 Component interactions in Auto Config architecture in PowerEdge servers

Note: In this technical white paper, user inputs are highlighted in blue color, while the file names, commands, scripts, and methods are indicated in the Courier New font style.
2.1.2 DHCP provisioning options

The AutoConfig → DHCP Provisioning attribute for an iDRAC can be set to one of the four. This attribute depends on DHCP configuration and can be set only if it is enabled. The values set for the attribute can be viewed by using any user privilege level. However, the admin or config iDRAC privileges are required to enable or disable the attribute.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disable</td>
<td>Default state. In this state, the iDRAC does not request for option 43 from the DHCP server, and the feature is turned off. Use when not using DHCP for server Auto Config.</td>
</tr>
<tr>
<td>Enable once</td>
<td>Applies the configuration changes immediately and can be used to configure a server one time with the Server Configuration Profile referenced by the DHCP server. After the DHCP server responds with option tag 43, this attribute turns to the Disable state.</td>
</tr>
<tr>
<td>Enable once after reset</td>
<td>A state in which option tag 43 is not requested until iDRAC is reset. After iDRAC is reset, this state changes to Enable once and the configuration changes are immediately applied.</td>
</tr>
<tr>
<td>Enable always</td>
<td>Used to configure iDRAC every time the iDRAC gets an IP address from the DHCP server. The option tag 43 is requested for every time the server is restarted. When the attribute value is changed from Disable to Enable always, no immediate action takes place. After the iDRAC is reset, changes take place immediately. The DHCP Provisioning option remains in the Enable always state after the changes have been made.</td>
</tr>
</tbody>
</table>

Figure 4  DHCP provisioning options

Note: The Enable always option is not recommended because it may result in the server being reconfigured at any new DHCP request, such as an iDRAC reset or introduction of another DHCP server. This option is not supported on iDRAC 2.10.10.10 and later versions of the 13th generation PowerEdge servers.

2.1.3 List of PowerEdge servers: 12th, 13th, and 14th generation

Visit www.dell.com/poweredge.
2.1.4 DHCP—current and new configuration options

<table>
<thead>
<tr>
<th>Current settings</th>
<th>New settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disable</td>
<td>Disable, Enable once, Enable once after reset, Enable always</td>
</tr>
<tr>
<td>Enable once</td>
<td>Disable, Do nothing, Restart DHCP process, Disable on complete, Wait for iDRAC reset, then Enable once, Enable always</td>
</tr>
<tr>
<td>Enable once after reset</td>
<td>Disable, Restart DHCP process, Disable on complete, Do nothing, Enable always</td>
</tr>
<tr>
<td>Enable always</td>
<td>Disable, Restart DHCP process, Disable on complete, Wait for iDRAC reset, then Enable once, Do nothing</td>
</tr>
</tbody>
</table>

2.1.5 Prerequisites for enabling Auto Config

Before enabling the Auto Config feature, ensure that the following are already set up in the test environment:

- The supported network share (NFS, CIFS, HTTP, or HTTPS) (HTTP and HTTPS for 14th generation servers) that is on the same subnet as the iDRAC and DHCP server. Test this network share first to ensure it can be accessed and that the firewall and user- or share permissions have been set up correctly.

- The Server Configuration profile is exported to the supported network share. For more information, see Getting Server Configuration Profiles in this technical white paper. Also, ensure that all the necessary changes in the profile are made to ensure that proper settings are applied when the Auto Config process is initiated.

- The DHCP server is set up and DHCP configuration updated as required for iDRAC to call DHCP server and run Auto Config feature. For more information, see Configuring Windows and Linux DHCP servers to support Auto Config.

**Note:** The procedure to fulfill these prerequisites is described later in this white paper. Also, a workflow is provided for validation of the Auto Config environment.

2.1.6 iDRAC interfaces to enable Auto Config

By default, the Auto Config feature is disabled in all servers. However, while ordering, you can request for enabling this feature by default at the factory settings. If not enabled, you can enable it by using the iDRAC GUI, RACADM, Redfish, and WS-Man interfaces as discussed in the following sections. You can also refer to the *iDRAC User's Guide* available on the support site and TechCenter.

2.1.7 Enabling Auto Config by using RACADM (local or remote)

RACADM provides the `iDRAC.NIC.AutoConfig` attribute, which can be modified regardless of the DHCP setting on the iDRAC.
2.1.8 DHCP provisioning enable setting

This attribute can be set to one of four values (0–3) and is displayed as a definition when queried:

\[ \text{iDRAC.NIC.AutoConfig (Disable/ Enable once/ Enable once after reset/ Enable always)} \]

Description: Set the Auto Config operation.

Possible Values:

- Disabled (0) — iDRAC does not perform DHCP configuration
- Enable once (1) — iDRAC performs DHCP configuration once
- Enable once after reset (2) — Performs configuration after iDRAC is reset
- Enable Always (3) — Always performs configuration

Default: Disabled

Write Privilege: Configure iDRAC

**Note**: The Enable always option is not recommended because it may result in the server being reconfigured when any new DHCP request is received, such as an iDRAC reset or introduction of another DHCP server. This option is removed from iDRAC 2.10.10.10 and later versions, and is no longer supported in the 13th generation and later PowerEdge servers.

- To set Auto Config by using RACADM, run the RACADM `get` and `set` subcommands.
- An example workflow for enabling Auto Config by using RACADM and a remote SSH session to the iDRAC, and then running the commands from the RACADM command line interface (CLI) is given here.

**Note**: User inputs are highlighted in blue color in this technical white paper. File names, commands, scripts, and methods are indicated by using the Courier New font style.

- These commands can also be run from a remote system by using `racadm -r <IPAddress> -u <UserName> -p <Password>` preceding the commands highlighted here:

1. Enumerate the Auto Config settings
   - `racadm get idrac.nic.autoconfig`
   - `racadm get idrac.nic.autoconfig [Key=idrac.Embedded.1#NIC.1]`
   - Object value modified successfully
   - AutoConfig=Disabled

2. Set the Auto Config settings
   - `racadm set idrac.nic.autoconfig "enable once after reset"`
   - `racadm set idrac.nic.autoconfig "enable once after reset" [Key=idrac.Embedded.1#NIC.1]`
   - Object value modified successfully
3. Validate the Auto Config settings
   - `racadm get idrac.nic.autoconfig`
   - `racadm get idrac.nic.autoconfig`
   - `[Key=idrac.Embedded.1#NIC.1]
     AutoConfig=Enable Once After Reset

   **Note:** After the Auto Config process is started, a system configuration import job ID is created. For more information about how to query for the import job ID and check the configuration results, see [RACADM workflow example using Auto Config server provisioning in a Linux environment](https://www.dell.com/support/article/sln212497).

2.2 Enabling Auto Config using WS-Man (winrm or openwsman)

- **WS-Man** provides the `NIC.1#AutoConfig` attribute which is part of the `DCIM_iDRACCardEnumeration` class. This enum attribute can be set to: Disabled, Enable once, Enable once after reset, or Enable always.


- For information about Linux openwsman, documentation, and setup instructions, see [http://sourceforge.net/projects/openwsman/](http://sourceforge.net/projects/openwsman/).

   **Note:** `NIC.1#AutoConfig` can be modified regardless of the DHCP setting.

<table>
<thead>
<tr>
<th>NIC.1#AutoConfig properties</th>
<th>Class name</th>
<th>Attribute name</th>
<th>IsReadOnly</th>
<th>Default</th>
<th>Possible values</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>iDRACCardEnumeration</td>
<td>NIC.1#AutoConfig</td>
<td>False</td>
<td>Disabled</td>
<td>Disabled (0): iDRAC does not perform DHCP configuration.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Enable once (1): iDRAC performs DHCP configuration once</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Enable once after reset (2): Performs configuration after iDRAC is reset.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Enable always (3): Always performs configuration</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Values = {&quot;0&quot;,&quot;1&quot;,&quot;2&quot;,&quot;3&quot;}</td>
</tr>
</tbody>
</table>
To set NIC.1#AutoConfig by using WS-Man, use the ApplyAttribute method which will set the pending value, create a target configuration job, and then apply the change immediately. Here is an example WS-Man workflow by using the winrm command to set NIC.1#AutoConfig. In this workflow, the current value of NIC.1#AutoConfig will be checked, a job created to set a new value, job status queried to verify success, and then verified that the value of NIC.1#AutoConfig has changed.

1. Get the current Auto Config value:

```
C:\winrm g http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/DCIM_iDRACCardEnumeration?InstanceID=iDRAC.Embedded.1#NIC.1#AutoConfig -u:root -p:calvin -r:https://192.168.0.120/wsman -SkipCNcheck -SkipCAcheck -encoding:utf-8 -a:basic

DCIM_iDRACCardEnumeration

AttributeName = AutoConfiguration enable
CurrentValue = Disabled
DefaultValue = Disabled
Dependency = null
DisplayOrder = 69
FQDD = iDRAC.Embedded.1
GroupDisplayName = NIC Information
GroupID = NIC.1
InstanceID = iDRAC.Embedded.1#NIC.1#AutoConfig
IsReadOnly = false
PendingValue = null
PossibleValues = Disabled, Enable Once, Enable Once After Reset, Enable Always
```

2. Set the updated Auto Config value:

```
C:\winrm i ApplyAttributes http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/root/dcim/DCIM_iDRACCardService?SystemCreationClassName=DCIM_ComputerSystem+CreationClassName=DCIM_iDRACCardService+SystemName=DCIM:ComputerSystem+Name=DCIM:IDRACCardService -u:root -p:calvin -r:https://192.168.0.120/wsman -SkipCNcheck -SkipCAcheck -encoding:utf-8 -a:basic@([Target="iDRAC.Embedded.1";AttributeName="NIC.1#AutoConfig";AttributeValue="Enable Once After Reset"])

ApplyAttributes_OUTPUT
Job

   EndpointReference
Address = http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
ReferenceParameters
```

---

Zero-touch, bare-metal server provisioning by using the Dell EMC iDRAC with Lifecycle Controller Auto Config feature
Zero-touch, bare-metal server provisioning by using the Dell EMC iDRAC with Lifecycle Controller Auto Config feature

3. Verify the job status:

```
```  
```
DCIM_LifecycleJob
    ElapsedTimeSinceCompletion = 0
    InstanceID = JID_307869268181
    JobStartTime = NA
    JobStatus = Completed
    JobUntilTime = NA
    Message = Job successfully Completed
    MessageArguments = NA
    MessageID = JCP007
    Name = iDRACConfig:iDRAC.Embedded.1
    PercentComplete = 100
```

4. Validate the Auto Config changes:

```
```  
```
DCIM_iDRACCardEnumeration
    AttributeDisplayName = Auto Configuration enable
    AttributeName = AutoConfig
    CurrentValue = Enable Once After Reset
    DefaultValue = Disabled
    Dependency = null
    DisplayOrder = 69
    FQDD = iDRAC.Embedded.1
    GroupDisplayName = NIC Information
    GroupID = NIC.1
    InstanceID = iDRAC.Embedded.1#NIC.1#AutoConfig
    IsReadOnly = false
    PendingValue = null
    PossibleValues = Disabled, Enable Once, Enable Once After Reset, Enable Always
```

**Note:** After the Auto Config process is started, a system configuration import job ID is created. For more information about how to query for the import job ID and check the configuration results, see [WS-Man (winrm) workflow example using Auto Config server provisioning in a Windows Server environment](#).
Enabling Auto Config using iDRAC graphical user interface (GUI)

The Network page on the iDRAC web GUI displays the Auto Config feature, enabling the selection of DHCP provisioning options. The Enable DHCP Provisioning option is available only if DHCP is enabled for IPv4. Select the values from the drop-down menu and restart the network to acquire the configuration file. The options available in the Enable DHCP Provisioning drop-down menu are: Disable, Enable Once, Enable once after reset, and Enable Always.

Note: These attributes are available only for DHCP-enabled iDRACs. If the iDRAC IP addressing is set to static, the Auto Config attributes are grayed out. To change the DHCP Provisioning options:

1. Log in to the iDRAC GUI.
2. For iDRAC7 or iDRAC8: On the iDRAC home page, click iDRAC Settings → Network.
3. Under Auto Config, from the Enable DHCP Provisioning drop-down menu, select a DHCP provisioning type.

Figure 5 Selecting DHCP Provisioning option on iDRAC8 GUI
4. For iDRAC9: On the iDRAC home page, click **iDRAC Settings → Connectivity → Network → Auto Config.**

![iDRAC Settings](image)

**Figure 6** Selecting DHCP Provisioning option on iDRAC9 GUI
5. Click **Apply**. The Network page is automatically refreshed.

The selected value is listed under **Auto Config**. An example screen shot is given here.

![Auto Config property selected on iDRAC8 GUI](image)

**Figure 7**  Auto Config property selected on iDRAC8 GUI
Zero-touch, bare-metal server provisioning by using the Dell EMC iDRAC with Lifecycle Controller Auto Config feature
After the Auto Config process is started, an import system configuration job ID is created. To query the job ID status, on the iDRAC7 or iDRAC8 GUI, click **Overview → Server → Job Queue**.

The **Job Queue** page is automatically refreshed until the job ID is marked as completed. The following screen shots show the progress of the Auto Config import system configuration job.

![IDRAC8 Auto Config import system configuration job running](image1.png)

**Figure 9** iDRAC8 Auto Config import system configuration job running

![IDRAC9 Auto Config import system configuration job running](image2.png)

**Figure 10** iDRAC9 Auto Config import system configuration job running
Zero-touch, bare-metal server provisioning by using the Dell EMC iDRAC with Lifecycle Controller Auto Config feature

Figure 11  iDRAC8 Auto Config import system configuration job completed

Figure 12  iDRAC9 Auto Config import system configuration job completed
To view information about an import job:

1. Log in to the iDRAC GUI.

2. For iDRAC7 or iDRAC8: On the iDRAC home page, click **Server** → **Logs** → **Lifecycle Log** → **Export**.

   This will export the complete Lifecycle Log information. Using a text editor, open the log file and search for the import job ID, and then view the configuration results for the import job.

The following screen shots show exporting the Lifecycle Log to a local file and using the WordPad editor to view the configuration results of the import job.
Zero-touch, bare-metal server provisioning by using the Dell EMC iDRAC with Lifecycle Controller Auto Config feature

Figure 15  Viewing Lifecycle Log in text editor

```
<Event AgentID="EM" Category="Configuration"
  Severity="Informational" Timestamp="2015-05-20T21:15:34-0500"
  Sequence="15809">
  <Message>Component configuration successfully changed.
  </Message>
  <MessageID>SYS059</MessageID>
  <FQDD>iDRAC.Embedded.1</FQDD>
  <MessageArgs/>
  <RawEventData/>
  <Comment/>
  <ConfigResults>
    <JobID>VID 32174521592</JobID>
    <FQDD>iDRAC.Embedded.1</FQDD>
    <JobName>Import Configuration</JobName>
    <JobDisplayName>Configure: Import system configuration
XML file</JobDisplayName>
    <Operation name="CHANGE">
      <DisplayValue>Boot Once</DisplayValue>
      <Name>ServerBoot.i#BootOnce</Name>
      <Detail>
        <OldValue>Enabled</OldValue>
```

```
3.1 Getting Server Configuration Profiles

- Getting server configuration profiles by using RACADM commands
- Getting server configuration profiles by using WS-Man commands

The Auto Config feature provides server configuration settings by using a Server Configuration Profile—a file that is typically created by exporting the settings from a "gold" configuration server. A common practice is to create configuration profiles for each specific server model or class of server to be deployed. Server Configuration Profiles can be created from a blank template or more usually, exported from "gold" configuration servers by using either RACADM, Redfish, or WS-Man, and saving the file to a supported NFS or CIFS network share. Auto Config for iDRAC9 also supports HTTP and HTTPS file sharing options.

![Diagram of server configuration flow](image)

Figure 16 Getting Server Configuration Profiles by using Auto Config

**Note:** Make sure the selected file share is set up first and can be accessed without any issues.

For 12th and 13th generation PowerEdge servers, the server configuration profiles are supported in XML format. For 14th generation PowerEdge servers, that support iDRAC 3.00.00.00 and later versions, the server configuration profiles are supported in both the XML and JSON formats.

3.1.1 Getting server configuration profiles by using RACADM commands

An example is given here to describe the process of exporting server configuration profiles to an NFS share by using RACADM commands in an SSH session. The export job is queried until the job is completed. These commands can also be run on a remote system by preceding `racadm -r <IPAddress> -u <UserName> -p <Password>` in the commands highlighted here.

1. Get the Server Configuration Profile.

   ```bash
   racadm>> get -f system_config.xml -t xml -l 192.168.0.130:/nfs_share
   racadm get -f system_config.xml -t xml -l 192.168.0.130:/nfs_share
   ```
RAC976: Export configuration XML file operation initiated.
Use the "racadm jobqueue view -i JID_307853717675" command to view the status of the operation.

**Note**: JSON is supported in iDRAC firmware version 3.00.00.00 or later by specifying “-t json”

2. View the job queue.

```
racadm>> jobqueue view -i JID_307853717675
racadm jobqueue view -i JID_307853717675
------------------------ JOB ------------------------
[Job ID=JID_307853717675]
 Job Name=Export: System configuration XML file
  Status=Completed
 Start Time=[Not Applicable]
 Expiration Time=[Not Applicable]
 Message=[SYS043: Successfully exported system configuration XML file.]
 Percent Complete=[100]
------------------------
```

racadm>>
4 Getting server configuration profiles by using WS-Man commands

Here is an example workflow by using the winrm command to export system configuration profile to an NFS share. The export job is queried until the job is completed.

1. **Export the Server Configuration Profile.**

   ```shell
   C:\> winrm i ExportSystemConfiguration http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/root/dcim/DCIM_LCService?SystemCreationClassName=DCIM_ComputerSystem+CreationClassName=DCIM_LCService+SystemName=DCIM:ComputerSystem+Name=DCIM:LCService -u:root -p:calvin -r:https://192.168.0.130/wsman -SkipCNCheck -SkipCACheck -encoding:utf-8 -a:basic @{IPAddress="192.168.0.130";FileName="system_config.xml";ShareName="/nfs_share";ShareType="0"}
   ```

   **ExportSystemConfiguration_OUTPUT**
   ```
   Job
   EndpointReference
   Address = http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
   ReferenceParameters
   SelectorSet
   Selector:InstanceID = JID_307858869989, cimnamespace = root/dcim
   ReturnValue = 4096
   ```

2. **Check the job status.**

   ```shell
   ```

   **DCIM_LifecycleJob**
   ```
   ElapsedTimeSinceCompletion = 3
   InstanceID = JID_307858869989
   JobStartTime = NA
   JobStatus = Completed
   JobUntilTime = NA
   Message = Successfully exported system configuration XML file.
   MessageArguments = NA
   MessageID = SYS043
   Name = Export Configuration
   PercentComplete = 100
   ```
For more information about the Server Configuration Profiles, see the following technical white papers:

- Server Configuration Profile Workflows
- Server Configuration XML File
- Lifecycle Controller (LC) XML Schema Guide

For information about getting Server Configuration Profiles by using Redfish, see the RESTful Server Configuration with iDRAC REST API white paper available at http://en.community.dell.com/techcenter/extras/m/white_papers/20443207/download.
5 Configuring Windows and Linux DHCP servers to support Auto Config

- Configuring Windows DHCP server option 43
- Configuring Windows DHCP server option 60
- Configuring Linux DHCP server

The following server "roles" are required:

- DHCP server
- File server (must support CIFS or NFS file sharing)

5.1 Configuring Windows DHCP server option 43

1. On the DHCP server, click **Start → Administration Tools → DHCP**. The DHCP server administration tool is started.
2. Find the server and expand all the items in the tree diagram.
3. Right-click **Scope Options** and select **Configure Options**.
4. In the **Scope Options** dialog box, select the **043 Vendor Specific Info** check box.
5. In the **ASCII** column, enter the IP address of the server where the Server Configuration Profile file is hosted. The value you enter is also displayed in the **Binary** column.

![Windows DHCP scope options](image)

Figure 17  Windows DHCP scope options

6. Click **OK** to save the configuration.
5.2 Configuring Windows DHCP server option 60

The Option 60 (Vendor Class) identifies and associates a DHCP client with a particular vendor. Any DHCP server configured to take action based on a client’s vendor ID must have option 43 and option 60 configured. iDRAC identifies itself with the vendor ID called “iDRAC”. Therefore, a new vendor class must be added, a scope option for “code 60” created under the new vendor class, and then the new scope option enabled for the DHCP server. To configure option 60:

1. On the DHCP server, click Start ➔ Administration Tools ➔ DHCP. The DHCP server administration tool is started.
2. Find the server and expand all the items in the tree diagram.
3. Right-click IPv4 and select Define Vendor Classes.
4. Click Add.
5. In the Edit Class dialog box, type the following:
   - **Display name**: Type **iDRAC**.
   - **Description**: Type **Vendor Class**.
   - **ASCII**: Type **iDRAC**.
6. Click OK, and then click Close.

![Windows DHCP Vendor Class dialog box](image)

**Figure 18**  Windows DHCP Vendor Class dialog box

7. Right-click IPv4 from the main DHCP console again, and then select Set Predefined Options.
8. From the Option class drop-down menu, select **iDRAC** as the vendor class, and then click Add.

   The **Option Type dialog** box is displayed.
9. In the **Name** box, enter `iDRAC`, select **String** as the data type and type `60` as the code.

10. Click **OK** to return to the DHCP console.

11. Expand all items under the server name on the DHCP console.

12. Right-click **Scope Options** and select **Configure Options**.

13. Click **Advanced**.

14. From the **Vendor class** drop-down menu, select `IDRAC`.

15. Select the **060 IDRAC** check box under **Available Options**.

![Scope Options](image)

**Figure 19** Setting iDRAC as the Vendor class

16. Under **Data entry**, type the string value that will be sent to the iDRAC to advise to pull the correct Server Configuration Profile file to import.

17. After entering the string, click **Apply**. This string changes depending on whether the Profile file is hosted in an NFS share or a CIFS share.

Examples of parameters and arguments for the string after this step.

**Note:** An empty space " " or whitespace is required in front of the first option such as the "-f" when using Windows DHCP server. Notice in front of the "-f" that there is one whitespace character.

**Note:** If you are using iDRAC 2.20.20.20 or later versions, a white space is not required.
Here is the list of required and optional parameters to be entered as the string value:

- `-f` (Filename): name of exported Server Configuration Profile file (mandatory for iDRAC 2.20.20.20 and earlier versions)

**Note:** If the `-f` parameter is not used, refer to the "File Naming Behavior with iDRAC firmware 2.20.20.20 or later" section.

- `-n` (Sharename): name of network share (mandatory for NFS or CIFS)
- `-s` (ShareType): Type 0 for NFS, or 2 for CIFS (mandatory)
- `-i` (IPAddress): IP address of the network share (mandatory)
- `-u` (Username): username that has access to network share (mandatory only for CIFS)
- `-p` (Password): user password that has access to network share (mandatory only for CIFS)
- `-d` (ShutdownType): type 0 for graceful or 1 for forced (default setting: 0) (optional)
- `-t` (Timetowait): time to wait for the host to shutdown (default setting: 300) (optional)
- `-e` (EndHostPowerState): type 0 for OFF or 1 for ON (default=1) (optional)

Alongside supporting NFS and CIFS-based file sharing, iDRAC firmware 3.00.00.00 or later also supports accessing profile files by using HTTP and HTTPS. The `-s` option flag is updated as follows:

- `-s` (ShareType): type `nfs` or 0 for NFS; `cifs` or 2 for CIFS; `http` or 5 for HTTP; or `https` or 6 for HTTPS (mandatory)
Zero-touch, bare-metal server provisioning by using the Dell EMC iDRAC with Lifecycle Controller Auto Config feature

**Note:** HTTPS Certificates are not supported with Auto Config. Auto Config will ignore certificate warnings.

The following additional option flags are supported in iDRAC 3.00.00.00 and later versions to enable the configuration of HTTP proxy parameters and set the retry timeout for accessing the profile file:

- `-pd` (ProxyDefault): use default proxy setting. (OPTIONAL)
- `-pt` (ProxyType): type `http` or `socks` (default setting `http`) (OPTIONAL)
- `-ph` (ProxyHost): IP address of the proxy host (OPTIONAL)
- `-pu` (ProxyUserName): user name that has access to the proxy server (mandatory for proxy support)
- `-pp` (ProxyPassword): user password that has access to the proxy server (mandatory for proxy support)
- `-po` (ProxyPort): port for the proxy server (default setting 80) (OPTIONAL)
- `-to` (Timeout): indicates the retry timeout in minutes for obtaining Profile file (default setting=60)

For more information about HTTP, see the *14G Support for HTTP and HTTPS across IDRAC/LC Interfaces* white paper available on the TechCenter.

Vendor Class option 60 string examples by using NFS and CIFS network shares:

**NFS:**
```
-f system_config.xml -i 192.168.1.101 -n /nfs_share -s 0 -d 1
```

**CIFS:**
```
-f system_config.xml -i 192.168.1.101 -n cifs_share -s 2 -u <USERNAME>
-p <PASSWORD> -d 1 -t 400
```

"Vendor Class" Option 60 string examples using JSON and HTTP/HTTPS with iDRAC 3.00.00.00 and later versions:

**HTTP:**
```
-f system_config.json -i 192.168.1.101 -s 5
```

**HTTP:**
```
-f http_share/system_config.xml -i 192.168.1.101 -s http
```

**HTTP:**
```
-f system_config.xml -i 192.168.1.101 -s http -n http_share
```

**HTTPS:**
```
-f system_config.json -i 192.168.1.101 -s https
```

Examples for configuring HTTP proxy server with iDRAC firmware 3.00.00.00 and later versions:

**HTTP:**
```
```

**HTTP:**
```
**Note:** Test the network share first and make sure that it can be accessed without any issues.

The figure shows the screen shot of Windows DHCP console scope options after all editing is complete. Both the 043 and 060 options must be listed.

![DHCP console with scope options](image)

**Note:** After configuring the DHCP server, DHCP services must be restarted.

### 5.3 Configuring the Linux DHCP server

1. Open the Linux DHCP configuration file (For example, dhcpd.conf) by using a text editor and add these lines:

   1.1 Add this string at the top of the DHCP configuration file, outside the braces:
   ```
   option myname code 43 = text;
   ```

2. Add these strings inside subnet entry braces:

   ```
   option vendor-class-identifier "iDRAC";
   set vendor-string = option vendor-class-identifier;
   option myname "-f system_config.xml -i 192.168.0.130 -u user -p password -n sambashare/config_files -s 2 -d 0 -t 500";
   ```

   The last string is an example of entering network share information for access to the Server Configuration Profile file. Edit this information by using the filename and network share information.
Required and optional parameters to be entered:

- `f` (Filename): name of exported Server Configuration Profile file (REQUIRED for iDRAC firmware versions prior to 2.20.20.20)

**Note:** If the `-f` parameter is not used, refer to File Naming Behavior with iDRAC firmware 2.20.20.20 or later section.

- `-n` (Sharename): name of network share (mandatory)
- `-s` (ShareType): type 0 for NFS or 2 for CIFS (mandatory)
- `-i` (IPAddress): IP address of the network share (mandatory)
- `-u` (Username): username that has access to network share (mandatory only for CIFS)
- `-p` (Password): user password that has access to network share (mandatory only for CIFS)
- `-d` (ShutdownType): either 0 for graceful or 1 for forced (default setting: 0) (OPTIONAL)
- `-t` (Timetowait): time to wait for the host to shutdown (default setting: 300) (OPTIONAL)
- `-e` (EndHostPowerState): either 0 for OFF or 1 for ON (default=1) (OPTIONAL)

The HTTP and HTTPS file sharing options are supported for iDRAC 3.00.00.00 and later versions. The `-s` option flag is updated as follows:

- `-s` (ShareType): Type nfs or 0 for NFS; cifs or 2 for CIFS; http or 5 for HTTP; or https or 6 for HTTPS (mandatory).

**Note:** HTTPS Certificates are not supported with Auto Config. Auto Config will ignore certificate warnings.

These additional option flags are supported in iDRAC firmware 3.00.00.00 and later versions to enable the configuration of HTTP proxy parameters and set the retry timeout for accessing the Profile file:

- `-pd` (ProxyDefault): Use default proxy setting. (OPTIONAL)
- `-pt` (ProxyType): Type http or socks (default=http) (OPTIONAL)
- `-ph` (ProxyHost): IP address of the proxy host (OPTIONAL)
- `-pu` (ProxyUserName): username that has access to the proxy server (REQUIRED for proxy support)
- `-pp` (ProxyPassword): user password that has access to the proxy server (REQUIRED for proxy support)
- `-po` (ProxyPort): port for the proxy server (default=80) (OPTIONAL)
- `-to` (Timeout): specifies the retry timeout in minutes for obtaining the configuration file (default=60)

For more information about HTTP, see the 14G Support for HTTP and HTTPS across iDRAC/LC Interface white paper available on the TechCenter.
Zero-touch, bare-metal server provisioning by using the Dell EMC iDRAC with Lifecycle Controller Auto Config feature

Parameter string examples for Linux NFS and CIFS network shares:

NFS: 
```
-f system_config.xml -i 192.168.0.130 -n /nfs -s 0 -d 0 -t 500
```

CIFS: 
```
-f system_config.xml -i 192.168.0.130 -n sambashare/config_files -s 2 -u user -p password -d 1 -t 400
```

Parameter string examples using JSON and HTTP/HTTPS for iDRAC 3.00.00.00 and later versions:

- HTTP: 
  ```
  -f system_config.json -i 192.168.1.101 -s 5
  ```

- HTTP: 
  ```
  -f http_share/system_config.xml -i 192.168.1.101 -s http
  ```

- HTTP: 
  ```
  -f system_config.xml -i 192.168.1.101 -s http -n http_share
  ```

- HTTPS: 
  ```
  -f system_config.json -i 192.168.1.101 -s https
  ```

Examples for configuring an HTTP proxy server by using iDRAC 3.00.00.00 and later versions:

- HTTP: 
  ```
  ```

- HTTP: 
  ```
  ```

**Note:** Test the network share first and make certain that it can be accessed without any issues.

**Note:** NFS4 is not supported on iDRAC 3.00.00.00 and earlier versions. Use NFS2 or NFS3 for NFS network share.

Example of edited dhcpd.conf file - lines in bold were added to the existing file:

```
option myname code 43 = text;
subnet 192.168.0.0 netmask 255.255.0.0  {
  default gateway
  option routers          192.168.0.1;
  option subnet-mask      255.255.255.0;
  option nis-domain       "domain.org";
  option domain-name      "domain.org";
  option domain-name-servers 192.168.1.1;
  option time-offset      -18000;    # Eastern Standard Time

  option vendor-class-identifier "iDRAC";
  set vendor-string = option vendor-class-identifier;
  option myname "-f system_config.xml -i 192.168.0.130 -u user -p password -n cifs -s 2 -d 0 -t 500";
  range dynamic-bootp    192.168.0.128 192.168.0.254;
  default-lease-time     21600;
  max-lease-time         43200;
}
```
Example of a static DHCP reservation from a dhcpd.conf file:

```plaintext
# Applying a config to a particular system specified by MAC address
host my_host {
    hardware ethernet b8:2a:72:fb:e6:56;
    fixed-address 192.168.0.211;
    option host-name "my_host";
    option myname "-f r630_raid.xml -i 192.168.0.1 -n /nfs -s 0 -d 0 -t 300";
}
```

**Note:** After editing the dhcpd.conf file, make sure to restart the dhcpd service for the changes to take effect.
5.4  File naming behavior with iDRAC firmware 2.20.20.20 or later

The Filename parameter is most useful when only a single configuration file is required for the servers by using Auto Config. If multiple server configurations are to be supported, the Filename parameter must be updated and the DHCP server restarted for each change in the configuration filename. To simplify the Auto Config process, iDRAC firmware version 2.20.20.20 and later provides dynamic configuration filename generation, making the Filename parameter optional.

![Diagram showing filename generation in iDRAC 2.20.20](image)

**Figure 22  Filename generation in iDRAC 2.20.20**

If the Filename parameter is not present when using iDRAC 2.20.20.20 and later versions, the iDRAC server configuration agent will automatically generate the configuration filename by using the server’s Service Tag, the server's model number, or using the default filename “config.xml.”

Here is the list of file names (in order) that the iDRAC 2.20.20.20 and later server configuration agent will use if the Filename parameter is not provided in the DHCP configuration:

- "<service tag>-config.xml" (Example: CDVH7R1-config.xml)
- "<model number>-config.xml" (Example: R630-config.xml)
- "config.xml"

If none of these files are located on the network share, the server configuration profile import job is indicated as “failed” because the file is not found.

Here are the examples of the dynamic file naming behavior of iDRAC 2.20.20.20 and later Auto Config, When using this feature, the Filename parameter does not have to be changed nor DHCP services restarted to support multiple, unique configuration files.

- Configure by using system Service Tag: Export the server configuration file and rename as `<service tag>-config.xml`. For each server that will be configured, create a unique configuration file by using the servers’ Service Tags. If three new servers with service tags CDVH7R1, CDVH7R2, and CDVH7R3 are to use Auto Config, create three configuration files “CDVH7R1-config.xml”, “CDVH7R2-config.xml” and “CDVH7R3-config.xml”—each file uniquely
configured as required. When these servers, each with iDRAC firmware 2.20.20.20 and later are enabled for Auto Config, the iDRAC server configuration agent will search the network share by using the `<service tag>-config.xml` name.

- **Configure by system model number:** Export the server configuration file and rename as `<model number>-config.xml`. For each server model to be configured, create a unique configuration file by using the server model.
  - To configure a PowerEdge R630, export the R630 configuration file and rename the file “R630-config.xml”.
  - For R730s, export an R730 configuration file and rename “R730-config.xml”.
  - Repeat for every model required. When an R730 with iDRAC firmware 2.20.20.20 and later version is enabled for iDRAC Auto Config, the iDRAC server configuration agent searches the network share for the file “R730-config.xml”, assuming there is no matching Service Tag-based configuration file.

- **Configure with default configuration file:** Export the server configuration file and name `config.xml`. Each server with iDRAC 2.20.20.20 and later version enabled for Auto Config will use this file by assuming that there are no matching Service Tag or model number-based configuration files on the network share.

For iDRAC firmware 3.00.00.00 and later versions, JSON format Profile files are supported. The following file names will be used if the Filename parameter is not present:

- “<service tag>-config.xml” (Example: CDVH7R1-config.xml)
- “<model number>-config.xml” (Example: R630-config.xml)
- “config.xml”
- “<service tag>-config.json” (Example: CDVH7R1-config.json)
- “<model number>-config.json” (Example: R630-config.json)
- “config.json”
5.4.1 RACADM workflow example using Auto Config server provisioning in a Linux environment

In this example, a Red Hat Enterprise Linux (RHEL) version 6.6 DHCP server and NFS share is used. A PowerEdge R730 server is already deployed in this environment with its BIOS, iDRAC, PERC, and NIC settings configured. A new R730 is arriving from the Dell factory and the goal is to apply the configuration settings from the "gold" R730 to the new R730 by using Auto Config. These commands can also be run on a remote system by using racadm –r <IPAddress> -u <UserName> -p <Password> preceding the commands highlighted here:

1. Capture the current Server Configuration Profile from the "gold". Log in to a Linux server, start an SSH session to connect to the R730’s iDRAC.

2. Type racadm which initiates a RACADM command session.

3. Run a "get" command which will export the Server Configuration Profile file to the NFS network share:

   ```
   racadm>> get -f system_config.xml -t xml -l 192.168.0.130:/nfs_share
   ```

   Use the "racadm jobqueue view -i JID_307853717675" command to view the status of the operation.

4. After running the "get" command, a job ID is displayed. Continue to query the job ID until the job is completed.

   ```
   racadm>> jobqueue view -i JID_307853717675
   ```

   Use the "racadm jobqueue view -i JID_307853717675" command to view the status of the operation.

5. On the Linux DHCP or NFS server, view the NFS share and ensure that the Server Configuration Profile file called "system_config.xml" has been exported.

6. Using a text editor, view the Profile file and make any attribute changes required for the new R730. Example below highlights in yellow the file on the NFS share:

   ```
   [root@sandXD nfs]# ls -la
   total 576
   drwxrwxrwx 2 root root 4096 May  5 12:25 .
   dr-xr-xr-x 28 root root 4096 Apr 28 03:39 ..
   -rw-rw-r-- 1 root root 105231 Apr 29 09:16 69T0C42-config.xml
   -rw-rw-r-- 1 root root 124890 Apr 30 10:11 R620-config.xml
   -rw-rw-r-- 1 root root 105232 Apr 29 08:45 R730-config.xml
   ```
Here is an example of the edited Profile file that will be used for this workflow. Observe that unwanted attributes have been removed and storage-related attributes added to create a virtual drive and to place HardDisk.List.1-1 as first device in the BIOS boot order.

```xml
<SystemConfiguration Model="PowerEdge R730" ServiceTag="69T0C42" TimeStamp="Tue May 5 17:26:40 2015">
  <Component FQDD="iDRAC.Embedded.1">
    <Attribute Name="Telnet.1#Enable">Enabled</Attribute>
    <Attribute Name="IPv6.1#Enable">Enabled</Attribute>
  </Component>
  <Component FQDD="RAID.Integrated.1-1">
    <Attribute Name="RAIDrebuildRate">50</Attribute>
    <Component FQDD="Disk.Virtual.1:RAID.Integrated.1-1">
      <Attribute Name="RAIDaction">Create</Attribute>
      <Attribute Name="Name">RAID ZERO</Attribute>
      <Attribute Name="Size">0</Attribute>
      <Attribute Name="StripeSize">128</Attribute>
      <Attribute Name="SpanDepth">1</Attribute>
    </Component>
  </Component>
</SystemConfiguration>
```

7. On the Linux DHCP/NFS server, update the dhcpd.conf file for Auto Config. After editing the dhcpd.conf file, save the changes and restart the DHCP services. The example below is the edited dhcpd.conf file. The entries highlighted in yellow are added to enable the DHCP server and NFS share to support Auto Config:

```bash
option myname code 43 = text;

subnet 192.168.0.0 netmask 255.255.0.0 {  
  default gateway
  option routers 192.168.0.1;
  option subnet-mask 255.255.255.0;
  option nis-domain "domain.org";
  option domain-name "domain.org";
  option domain-name-servers 192.168.1.1;
  option time-offset -18000;  # Eastern Standard Time
  option vendor-class-identifier "iDRAC";
  set vendor-string = option vendor-class-identifier;
  option myname "-f system_config.xml -i 192.168.0.130 -n /nfs -s 0";
  range dynamic-bootp 192.168.0.128 192.168.0.254;
  default-lease-time 21600;
  max-lease-time 43200;
}
```

**Note:** Below steps 8 through 11 are required **only** if the server was ordered from the factory without DHCP Auto Config enabled.
8. When the new R730 server arrives from the factory, connect the A/C power (do not turn on the server but leave in the OFF state) and connect a network cable to the dedicated iDRAC port.

9. Assuming that the network subnet is on the same subnet as the default iDRAC IP address (192.168.0.120), start an SSH session to the iDRAC and initiate a racadm command session.

10. Set the iDRAC.NIC.AutoConfig attribute to “Enable Once” and enable the iDRAC DHCP. After DHCP is enabled, iDRAC will obtain an IP address from the DHCP server and initiate the Auto Config process:

```
[root@sandXD ~]# ssh 192.168.0.120
The authenticity of host '192.168.0.120 (192.168.0.120)' can't be established.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.0.120' (RSA) to the list of known hosts.
root@192.168.0.120's password:
WARNING: Default password is configured. Dell highly recommends changing user root's password immediately
/admin1-> racadm
racadm>> set idrac.nic.autoconfig "enable once"
racadm set idrac.nic.autoconfig "enable once"
[Key=idrac.Embedded.1#NIC.1]
Object value modified successfully
racadm>> get idrac.nic.autoconfig
racadm get idrac.nic.autoconfig
[Key=idrac.Embedded.1#NIC.1]
AutoConfig=Enable Once
racadm>> setniccfg -d
racadm setniccfg -d
DHCP is now ENABLED
racadm>> exit
/admin1->
```

**Note:** If the DHCP subnet is not on the same subnet as the default IP address of the iDRAC (192.168.0.120), use either the server front LCD panel (if the server supports it) or access the servers configuration settings by using the BIOS F2 / iDRAC Settings / Network interface and enable DHCP.

11. After a DHCP IP address is obtained for iDRAC, start an SSH to the iDRAC by using the DHCP IP address and set iDRAC.NIC.AutoConfig to “Enable Once”. After setting Auto Config to “Enable Once”, the Auto Config process will start within a few moments.

12. On the Linux DHCP/NFS server, view the /var/lib/dhcpd/dhcpd.leases file to ensure that a DHCP IP address has been assigned to the iDRAC. If the new server is locally accessible and the server supports an LCD panel, the iDRAC IP address can be verified by using the LCD panel:

```
[root@sandXD ~]# tail -n 11 /var/lib/dhcpd/dhcpd.leases
lease 192.168.0.125 {
```
13. When the Auto Config process starts, a job ID will get created. Using the iDRAC DHCP IP address, start an SSH session to run the RACADM session to query the status of this job ID. If there are multiple jobs already in the job queue, the Auto Config job must be the last job ID listed.

The job can also be identified by looking for “Job Name=Configure: Import system configuration XML file”. Continue to query the job queue until “Status=Completed”.

```
racadm>> jobqueue view

racadm jobqueue view

-------------------------------JOB QUEUE-------------------------------

[Job ID=JID_308867078109]
Job Name=Configure: Import system configuration XML file
Status=Running
Start Time=[Not Applicable]
Expiration Time=[Not Applicable]
Message=[SYS058: Applying configuration changes.]
Percent Complete=[10]

-------------------------------JOB QUEUE-------------------------------

racadm>> jobqueue view -i JID_308867078109

racadm jobqueue view -i JID_308867078109

-------------------------------JOB-------------------------------

[Job ID=JID_308867078109]
Job Name=Configure: Import system configuration XML file
Status=Running
Start Time=[Not Applicable]
Expiration Time=[Not Applicable]
Message=[SYS031: Updating component configuration.]
Percent Complete=[46]

-------------------------------JOB-------------------------------
```
Zero-touch, bare-metal server provisioning by using the Dell EMC iDRAC with Lifecycle Controller Auto Config feature

```
racadm> jobqueue view -i JID_308867078109
racadm jobqueue view -i JID_308867078109

--------------------------- JOB --------------------------
[Job ID=JID_308867078109]
Job Name=Configure: Import system configuration XML file
Status=Running
Start Time=[Not Applicable]
Expiration Time=[Not Applicable]
Message=[SYS031: Updating component configuration.]
Percent Complete=[90]

--------------------------- JOB --------------------------
[Job ID=JID_308867078109]
Job Name=Configure: Import system configuration XML file
Status=Completed
Start Time=[Not Applicable]
Expiration Time=[Not Applicable]
Message=[SYS053: Successfully imported and applied system configuration XML file.]
Percent Complete=[100]
```

14. To view the job details, use the Auto Config import job ID that was just queried in the job queue and validate the configuration results by running the lclog viewconfigresult command. This displays the results of the configuration changes that were applied to the new server. Example of configuration results for an Auto Config job:

```
racadm>> lclog viewconfigresult -j JID_308867078109
racadm lclog viewconfigresult -j JID_308867078109
SeqNumber = 11557
FQDD = NIC.Integrated.1-1-1
DisplayValue = LegacyBootProto
Name = LegacyBootProto
OldValue = NONE
Status = Success
```
5.5 WS-Man (winrm) workflow example using Auto Config server provisioning in Windows Server environment

This example assumes that Windows Server 2012 DHCP server, CIFS share, and multiple currently deployed PowerEdge R730 servers with common BIOS settings are used. Also, there is a separate R730 server, located in a lab on a static network environment that will be relocated to the Windows Server DHCP environment. The goal is to update that R730 to apply the same BIOS configuration as the currently deployed R730 server’s.

1. Begin by retrieving the current BIOS configuration from one of the R730 servers within the DHCP environment. Using WS-Man winrm commands, invoke ExportSystemConfiguration to export the Server Configuration Profile file to a CIFS share.

2. Because the objective is to synchronize the BIOS configuration, use the parameter Target and specify the value “BIOS”.
By using the Target parameter, a selective export will be performed and a Server Configuration Profile file holding only the BIOS attributes is created.

```powershell
C:\winrm i ExportSystemConfiguration
http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/root/dcim/DCIM_LCService?SystemCreationClassName=DCIM_ComputerSystem+CreationClassName=DCIM_LCService+SystemName=DCIM:ComputerSystem+Name=DCIM:LCService -u:root -p:calvin -r:https://192.168.1.201/wsman -SkipCNCheck -SkipCACheck -encoding:utf-8-a:basic 8{IPAddress="192.168.1.101";FileName="bios_config_only.xml";ShareName="cifs_share";ShareType="2";Username="administrator";Password="password";Target ="BIOS"}

ExportSystemConfiguration_OUTPUT

Job
   EndpointReference
      Address = http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
   ReferenceParameters
      SelectorSet
         Selector: InstanceID = JID_313751705727, cimnamespace = root/dcim
   ReturnValue = 4096

3. After invoking the ExportSystemConfiguration method, a job ID is displayed. Query the job ID until it is marked as “Completed”.

```powershell

DCIM_LifecycleJob
   ElapsedTimeSinceCompletion = 0
   InstanceID = JID_313751705727
   JobStartTime = NA
   JobStatus = Completed
   JobUntilTime = NA
   Message = Successfully exported system configuration XML file.
   MessageArguments = NA
   MessageID = SYS043
   Name = Export Configuration
   PercentComplete = 100.

4. Using Windows Explorer, locate the CIFS share and verify that the Server Configuration Profile file named “bios_config_only.xml” has been created.
5. Use a text editor to view the XML file and verify that it contains all of the BIOS attribute settings to be applied. Here are the screen shots of the Profile file on a CIFS share named “cifs_share” along with the Profile file contents.

Figure 23  Configuration Profile file on CIFS

Figure 24  Viewing Configuration Profile file in text editor

6. Configure the Windows DHCP server to use the just-created Server Configuration Profile file. Access the Windows DHCP server, navigate to Server Manager / Tools and start the DHCP tool.
7. Because the Windows DHCP server is already configured for Auto Config, Option 60 must be updated with the correct network share parameters to use the designated CIFS share and the Server Configuration Profile file that was just exported.

Figure 25 Updating Option 60

8. After editing Option 60, restart the DHCP services.

**Note**: If the DHCP server is not already set up to support Auto Config, refer to the Configuring Windows and Linux DHCP servers to support Auto Config section in this white paper. **Note**: A white space is required in front of the first option within the Option 60 string such as the “-f” when using Windows DHCP server.

9. After configuring the Windows DHCP server, the target R730 server must be configured for Auto Config and connect the R730 to the Windows DHCP environment. Because the R730 has been in a static network environment, R730 server’s iDRAC networking must be reconfigured for DHCP, and then Auto Config set to “Enable once after reset”.

These settings are recommended in this case because the R730 will be physically relocated and because A/C power is disconnected from the server, this will cause the R730 server’s iDRAC to initiate both DHCP access and start the Auto Config process when power is restored to the server. Using WS-Man, the ApplyAttributes method will be used to set the iDRAC’s DHCP and Auto Config attributes.

```plaintext
```

ApplyAttributes_OUTPUT

```
Job

EndpointReference
   Address = http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
ReferenceParameters
SelectorSet
```
Zero-touch, bare-metal server provisioning by using the Dell EMC iDRAC with Lifecycle Controller Auto Config feature

Selector: InstanceID = JID_313783436360, __cimnamespace = root/dcim
ReturnValue = 4096

An example XML file provided for the ApplyAttributes method:

```xml
C:\type apply_attributes.xml

    <p:Target>iDRAC.Embedded.1</p:Target>
    <p:AttributeName>NIC.1#AutoConfig</p:AttributeName>
    <p:AttributeValue>Enable Once After Reset</p:AttributeValue>
    <p:AttributeName>IPv4.1#DHCPEnable</p:AttributeName>
    <p:AttributeValue>Enabled</p:AttributeValue>
</p:ApplyAttributes_INPUT>
```

10. After invoking the ApplyAttributes method, a job ID is displayed. Query the job ID status until it is marked as completed.

```bash
```

DCIM_LifecycleJob

- ElapsedTimeSinceCompletion = 0
- InstanceID = JID_313783436360
- JobStartTime = NA
- JobStatus = Completed
- JobUntilTime = NA
- Message = Job successfully Completed
- MessageArguments = NA
- MessageID = JCP007
- Name = iDRACConfig:iDRAC.Embedded.1
- PercentComplete = 100

**Note:** When DHCP is enabled on the iDRAC and power is supplied, the iDRAC will initiate DHCP requests to the DHCP server within 10–15 seconds. After iDRAC interacts with the DHCP server, the iDRAC will begin using its newly supplied IP address. After this, it is necessary to obtain that new IP address to run the `winrm` command to check job status.

11. After verifying the completed job status, disconnect power from the R730 server and move the server to the new Windows Server DHCP environment.

12. Connect a network cable to the iDRAC dedicated network port and connect A/C power but do not turn on the server.

    The iDRAC will start and initiate the Auto Config process. An import Server Configuration Profile job will be created and automatically power on the server to execute the BIOS configuration changes.
Before determining the status of this import job ID, the DHCP IP address assigned to the R730’s iDRAC must be discovered.

The IP address can be determined either from the Windows DHCP server Address Leases directory, from the R730’s front LCD panel, or can note the iDRAC IP reported on the server console during POST. For this example, note the iDRAC IP address 192.168.1.201 reported during POST.

![Image](image.png)

Figure 26 Discovering IP address assigned to iDRAC by DHCP server

13. The new iDRAC IP address is obtained. Query the job queue to find job ID for the Server Configuration Profile import job.

14. Search for a job ID with “Name = Import Configuration”. This normally should be the last job ID listed in the job queue.

15. Continue to query the job queue until it reports a completed job status. For example, the import job ID that was created is “JID_313798193926”.

```
C:\winrm\root\dcim/DCIM_LifecycleJob -u:root -p:calvin -r:https://192.168.1.201/wsman:443 -SkipCNCheck -SkipCACheck -auth:basic -encoding:utf-8
```

DCIM_LifecycleJob

```
ElapsedTimeSinceCompletion = null
InstanceID = JID_CLEARALL
JobStartTime = TIME_NA
JobStatus = Pending
JobUntilTime = TIME_NA
```
Message = NA
MessageArguments = NA
MessageID = NA
Name = CLEARALL
PercentComplete = 0

DCIM_LifecycleJob
ElapsedTimeSinceCompletion = 35
InstanceID = JID_313783436360
JobStartTime = NA
JobStatus = Completed
JobUntilTime = NA
Message = Job successfully completed
MessageArguments = NA
MessageID = JCP007
Name = iDRACConfig:iDRAC.Embedded.1
PercentComplete = 100

DCIM_LifecycleJob
ElapsedTimeSinceCompletion = 6
InstanceID = JID_313798193926
JobStartTime = NA
JobStatus = Completed
JobUntilTime = NA
Message = Successfully imported and applied system configuration XML file.
MessageArguments = NA
MessageID = SYS053
Name = Import Configuration
PercentComplete = 100

More information about the configuration results can be obtained by using the import job ID. Invoking the GetConfigResults method will return all configuration changes that were applied to the server.

16. To more readily view the configuration details, redirect the command output into a file and open that file with a text editor.


GetConfigResults_OUTPUT

ConfigResults =
Zero-touch, bare-metal server provisioning by using the Dell EMC iDRAC with Lifecycle Controller Auto Config feature
## Troubleshooting Auto Config issues

<table>
<thead>
<tr>
<th>Issue</th>
<th>Resolution</th>
</tr>
</thead>
</table>
| The server fails to restart or turn on after enabling Auto Config.  | 1) Check the job queue and determine the status of the Server Configuration Profile import job. If the job failed, check the job status message for indications about why the job failed. Some examples of causes of job failure:  
   - iDRAC is unable to access network share  
   - file not located on the network share  
   2) Check the job queue for an import job created. If no import job ID is created within ten minutes after enabling Auto Config, there was likely an issue reading the DHCP configuration file. Check the Linux DHCP configuration file or Windows DHCP configuration panels for typos and input format. If corrections are made to the Linux or Windows DHCP server, be certain to restart the DHCP services. |
| The import job ID failed because of inability to access the network share. | Check for these common issues:  
   - If a firewall is enabled, ensure that the correct ports are open for NFS and CIFS and that the iDRAC IP address can access that subnet.  
   - Ensure that permissions to access the NFS or CIFS share are properly set.  
   - For NFS shares, Auto Config currently does support secured access; ensure everyone is enabled to access the NFS share.  
   - For CIFS, ensure that the username and password being passed in the DHCP configuration file have read and write access to the share. |
| When using the Windows DHCP server, the Auto Config import job fails by indicating that file is not found. | Check for Option 60 and ensure there is a white space character before the first option. |
- Issue: When using iDRAC 2.20.20.20 and later versions, new log entries for the Lifecycle Controller (LC) log have been added to help track the Auto Config process. Viewing the LC log data can help when monitoring or troubleshooting Auto Config issues. Screen shot shows an example for successful Auto Config workflow with log entries for Auto Config process highlighted:

![Screen shot showing an example for successful Auto Config workflow with log entries for Auto Config process highlighted](image)

Figure 27  Auto Config successfully completed

Screen shot showing an example for failing Auto Config workflow with log entries for Auto Config process highlighted:
Zero-touch, bare-metal server provisioning by using the Dell EMC iDRAC with Lifecycle Controller Auto Config feature

Figure 28  Unsuccessful Auto Config
7 Conclusion

Auto Config is a powerful feature that allows an IT administrator to configure and provision all components in PowerEdge servers. It allows rapid, automated provisioning of multiple servers. Using Auto Config, configuration changes can be made to multiple servers by editing the Server Configuration Profile, storing the file in a CIFS or NFS share, and enabling this feature. Auto Config also supports configuration files that are specific by server Service Tag, specific by server model, or generic for all servers. With iDRAC 3.00.00.00 and later versions, profiles in JSON format are supported as-is Auto Config by using HTTP or HTTPS.

Dell EMC continues to provide its customers with products that simplify and streamline their IT processes, freeing administrator’s time to focus on activities that help grow the business. For more information about iDRAC with Lifecycle Controller, visit the Dell TechCenter.
<table>
<thead>
<tr>
<th><strong>Glossary</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DHCP scope option</strong></td>
</tr>
<tr>
<td><strong>iDRAC</strong></td>
</tr>
<tr>
<td><strong>PERC</strong></td>
</tr>
<tr>
<td><strong>RACADM</strong></td>
</tr>
<tr>
<td><strong>WS-Man</strong></td>
</tr>
<tr>
<td><strong>Vendor-specific DHCP scope options</strong></td>
</tr>
</tbody>
</table>