Lifecycle Controller: How to Use Auto Backup and Auto Update for all Supported Interfaces

This Dell Technical white paper discusses how to use the Auto Backup and Auto Update features on Dell PowerEdge servers.

Dell Engineering
June 2014
## Revisions

<table>
<thead>
<tr>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>June 2014</td>
<td>1.0 Release: Initial release</td>
</tr>
</tbody>
</table>

TYPOGRAPHICAL ERRORS AND TECHNICAL INACCURACIES. THE CONTENT IS PROVIDED AS IS, WITHOUT EXPRESS OR IMPLIED WARRANTIES OF ANY KIND.

© 2014 Dell Inc. All rights reserved. Reproduction of this material in any manner whatsoever without the express written permission of Dell Inc. is strictly forbidden. For more information, contact Dell. Dell, the DELL logo, and the DELL badge are trademarks of Dell Inc. Symantec, NetBackup, and Backup Exec are trademarks of Symantec Corporation in the U.S. and other countries. Microsoft, Windows, and Windows Server are 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 any proprietary interest in the marks and names of others.
Contents
Before You Begin ................................................................. 5
Why use Auto Backup? ........................................................... 6
Why use Auto Update? ............................................................ 7
Supported Interfaces .................................................................. 8
Auto Backup Workflow Using iDRAC GUI .................................... 9
Auto Backup Workflow Using WINRM ...................................... 17
Auto Backup Workflow Using RACADM .................................... 24
Auto Update Workflow Using iDRAC GUI .................................... 28
Auto Update Workflow Using WSMAN ..................................... 36
Auto Update Workflow Using RACADM .................................... 48
Conclusion: ........................................................................... 55
Executive summary

iDRAC7 with Lifecycle Controller supports a wide variety of tools and options that enable system administrators to perform firmware updates and backup server profiles. On iDRAC 1.50.50 and later versions, provide the ability to automate the firmware update (Auto Update) and server profile backup (Auto Backup) to run on a recurring basis. This new feature enables you to schedule updates and back up the server profile on a daily, weekly or monthly basis. Once the features are enabled and setup, the system will automatically perform firmware updates and server profile backups at the scheduled recurrence time.
Before you begin

Before you begin, make sure that the following prerequisites are met:

- The target server is a Dell PowerEdge server with iDRAC enabled, configured, and accessible through a network connection.
- iDRAC must either have Express or Enterprise license installed.
- For auto backup, your target server will require access to a supported network share (FTP, NFS or CIFS). Also, make sure that the share has enough space (average backup file size is 100 MB) available.
- A remote repository with firmware updates is accessible through iDRAC. If the iDRAC’s network cannot access ftp://ftp.dell.com, create a custom repository using the latest Dell Repository Manager Dell Repository Manager - Systems Management.
- Once a custom repository is generated, make it accessible through FTP, NFS or CIFS.

For example: Create a custom repository using the Dell Repository manager named “custom_repository” on the root of the C drive which contains the DUPs and catalog.xml. Share this directory on the network as a FTP, NFS or CIFS share.
Why use Auto Backup?

The legacy server profile backup feature provides the option to perform a single backup either immediately or at a specified time in the future. However, it lacks the ability to schedule a recurring backup that occurs automatically at a specified time. For example, if you want a server profile backup to occur every Saturday at 12 AM, the only way to achieve this is to script it external to iDRAC. This process is time consuming. iDRAC 1.50.50 and later versions, allow you to schedule backups on a daily, weekly or monthly basis. By setting up the scheduled backups, an accurate current backup image of the host is always available. This helps when the host goes down and the system board needs to be replaced.

A backup image file contains:

- **Readable**
  - System identification information such as model number and Service Tag. For example, PowerEdge R720 and 1P3H8BS.
  - Date and time the backup was completed.
  - Currently installed hardware inventory information.
  - Firmware for each component.

- **Encrypted**
  - Component configuration information.
  - User name and password for RAID controller and BIOS.
  - Component certificates.
  - Licenses.
  - Signature to validate backup file has not been tampered with and was generated by Lifecycle Controller.

A backup image does not contain:

- Operating system or any data stored on hard disk drives (HDDs) or virtual drives (VDs)
- vFlash SD card partition information
- Lifecycle log
- Dell diagnostics
- Dell OS Driver Pack
Local Key Management (LKM) passphrase if the LKM–based storage encryption is enabled. However, you must provide the LKM passphrase after performing the restore operation. Why use Auto Update?

The legacy firmware updates feature allows you to update one or multiple devices and schedule the update either now or a time in the future. This is a onetime execution and if you want to update the devices again for example in a month, then you need to perform the same manual task again or you will have to create a custom script to handle this work. This effort is time consuming. Starting with iDRAC1.50.50 and later, iDRAC7 simplifies this process by enabling you to schedule recurring firmware updates and iDRAC automatically fetches and installs the updates as and when they are available. You can either use a custom repository or the Dell ftp repository to perform this operation. You no longer have to manually search for the latest firmware packages available. After the scheduled time is elapsed, iDRAC queries the remote repository for a list of available updates and compares it to the versions that are currently installed. If new updates are found, iDRAC downloads the update packages and updates the corresponding devices.

After the updates are processed, one of the following messages are logged to the Lifecycle log. You can also setup iDRAC to send out an email or SNMP notifications.

- The scheduled update from repository job successfully completed. Applicable updates were not found.
- The recurring scheduled update from repository job completed and updates were applied. A system restart was not required.
- The recurring scheduled update from repository job completed and updates are staged to run after the next system restart.
- The recurring scheduled update from repository job completed and updates were staged. The system will now restart to apply the staged updates.
Supported Interfaces
Auto Backup and Auto Updates are supported from the following user interfaces:

- Integrated Dell Remote Access Controller (iDRAC GUI) - Management controller for blades and monolithic servers.
- Web Services Management (WS-MAN) - API that the iDRAC uses for remote management
- RACADM (iDRAC command line)
Auto Backup Workflow Using iDRAC GUI

The workflow below explains how to enable and schedule an auto backup using the iDRAC GUI.

1. Launch the iDRAC GUI.
2. Navigate to Overview -> iDRAC Settings -> Server Profile.
3. On the Backup and Export Server Profile page, click Automatic Backup.
4. Select the Enable Automatic Backup option.
5. In File Settings and Security section, enter the following details:
   - File Location – Select the location where you want to save the backup file image. Select Network to save the backup image file on a CIFS or NFS share. Select vFlash to save the image to a vFlash card.
   - File Name – Type the backup file name such as R620_backup_file.
   - Backup File Passphrase (optional) – Provide the encryption passphrase. This is optional.
   - Confirm Passphrase – Re-enter to confirm the encryption passphrase for the server profile
6. In the File Settings and Security section, if you have selected Network as the file location, then under Network Settings, enter the following details:
   - Number of backup files to preserve (1-50) – Enter a number to denote the number of backup files copies that you would like to preserve.
   - IP Address – Enter the IP address or the host name of the shared folder.
   - Protocol – Select the network share (CIFS or NFS).
   - Share Name – Enter the name of the shared folder.
   - Domain Name – Enter the domain name required to login to the network share.
   - User Name - Enter the user name required to login to the network share.
   - Password - Enter the password required to authenticate the user name.

   **Note:** Click Test network connection to verify if the network share is accessible using the specified user name and password.

7. In the Backup Window Schedule section, enter the following details:
   - Start (24hr format) – Enter the time when the backup operation must start.
   - Recurrence Pattern – Specify the recurrence pattern for the automatic backup:
     - Daily – Select this option to schedule a backup once every x number of days. The valid values are 1-366.
Weekly – Select this option to schedule weekly backups. Enter the week number in the Recur every [1–52] week(s) on field and then select the day of the week. For example, Recur every 10 week(s) on Monday.

Monthly – Select this option to schedule monthly backups based on the following:

- **Day [1–31] of every [1–12] month(s).** For example, Day 4 of every 11 months.
- **The [Last/First] [day of week] of every [1–12] month(s).** For example, The Last Sunday of every 10 months.

8. Click **Schedule Backup.** The input parameters are validated and a success message is displayed. A recurring job is displayed in the job queue with a start date and time of the next scheduled backup operation.

9. Click **Job Queue** to navigate to the job queue page to view and monitor the status of all jobs. A new job - **Automatic Backup: Image** is created. Expand the job ID to view the detailed schedule information about the auto backup job.

After the scheduled start time elapses, the auto backup job starts automatically (backup job usually complete in 20–30 minutes depending on system hardware configuration. Refresh the job queue page or expand the job ID to view the backup job details). A new job for the next recurrence is created and scheduled after the current backup job is initiated.

**Note:** Every time the auto backup process is initiated at the scheduled start time, a new backup file with incremental file names is created and archived in the specified location.

**iDRAC GUI image examples (covers steps 1 thru 9 mentioned above):**

Steps 1 thru 6:
Step 7:
Step 8:
Step 9:
Auto Backup Workflow Using WINRM

The workflow below illustrates how to use WINRM to enable and schedule auto backups to a CIFS share.

1. Run a get on the `DCIM_LCEnumeration` class passing the LifecycleController.Embedded.1#LCAtributes.1#AutoBackup attribute.

**WINRM Command and Output Example:**

```
```

DCIM_LCEnumeration

AttributeName = Automatic Backup Feature
CurrentValue = Disabled
DefaultValue = Disabled
ElementName = LC.emb.1
InstanceID = LifecycleController.Embedded.1#LCAtributes.1#AutoBackup
IsReadOnly = false
PendingValue = null
PossibleValues = Disabled, Enabled

2. To enable the Automatic Backup feature, use the `SetAttributes` and `CreateConfigJob` methods to set the `Automatic Backup Feature` attribute to `Enabled`.

**WSMAN Command and Output Examples:**

```
```

SetAttributes_OUTPUT

Message = The command was successful
MessageID = LC001
RebootRequired = No
ReturnValue = 0
SetResult = Set PendingValue
3. View the current iDRAC time using the **ManageTime** method.

**WSMAN Command and Output Example:**

```plaintext
winrm i ManageTime "cimv2/root/dcim/DCIM_TimeService?SystemCreationClassName=DCIM_SPComputerSystem+SystemName=systemmc+CreationClassName=DCIM_TimeService+Name=DCIM TimeService 1" @{GetRequest="true"} u:IDRAC_USERNAME -p:iDRAC_PASSWORD -r:https://IDRAC_IP /wsman -SkipCNcheck -SkipCAcheck -encoding:utf-8 -a:basic
Message = The command was successful
MessageID = RAC048
ReturnValue = 0
TimeData = 20140102165203.000000-360
```

4. Using your CIFS network credentials, pass in these values for the parameters and invoke the **SetBackUpSchedule** method on the **DCIM_LCService** class:

- IPAddress
- ShareName
- ShareType
- Username
- Password
- ImageName
- Time
- Repeat
- MaxNumberOfBackupArchives

**WSMAN Command and Output Example:**

```
winrm i SetBackupSchedule 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:LC
Service u:IDRAC_USERNAME -p:iDRAC_PASSWORD -r:https://IDRAC_IP/wsman -
SkipCNCheck -SkipCACheck -encoding:utf-8 -a:basic @{IPAddress="172.23.200.26";ShareName="cifs";ShareType="2";Username="administra
";Password="password123";ImageName="backupfile_R620";Time="17:20";Repeat ="12";MaxNumberOfBackupArchives="12"}
SetBackupSchedule_OUTPUT
ReturnValue = 4096
```

5. Run an **enum** on **DCIM_LifecycleJob** class which will return the job queue. An auto backup job ID with the specified job start time is displayed.

**WSMAN Command and Output Example:**

```
winrm e cimv2/root/dcim/DCIM_LifecycleJob u:IDRAC_USERNAME -p:iDRAC_PASSWORD
-r:https://IDRAC_IP/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 = null
    InstanceID = JID_887044678792
    JobStartTime = 20140102172000
    JobStatus = Ready For Backup
```
After the start time elapses, check the job queue again to view the progress on the auto backup job (backup usually takes 20-30 minutes based on system hardware configuration).

WSMAN Command and Output Example:

```
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
```

After it is completed, the auto backup job is marked as **Completed** and a new auto backup job for the same time at a future date is added to the job queue.

WSMAN Command and Output Example:

```
DCIM_LifecycleJob
  ElapsedTimeSinceCompletion = null
  InstanceID = JID_887044678792
  JobStartTime = 20140102172000
  JobStatus = Backup In Progress
  JobUntilTime = 20140102172500
  Message = Collecting Lifecycle Controller Firmware images.
  MessageArguments = NA
  MessageID = BAR063
  Name = Automatic Backup:Image
  PercentComplete = 50
```
A directory with the name you specified (for example, Archive_YourServiceTag) is created in the specified location. The auto backup file with a _1 appended at the end of the file name (for example, R620_backup_file_1) is stored in this directory. Each time the auto backup process passes the scheduled start time, backup files are created and stored in this same directory.

To check the auto backup information, invoke the GetBackupSchedule method on the DCIM_LCService class. This method returns all parameter information that you passed in while invoking the SetBackupSchedule method earlier.

WSMAN Command and Output Example:

```
winrm i GetBackupSchedule http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/root/dcim/DCIM_LCService?SystemCreationClassName=DCIM_ComputerSystem+Creation=DCIM_LCService+SystemName=DCIM:ComputerSystem+Name=DCIM:LCService
-u:IDRAC_USERNAME -p:iDRAC_PASSWORD -r:https://IDRAC_IP/wsman -SkipCNCheck -SkipCACheck -encoding:utf-8 -a:basic
```

GetBackupSchedule_OUTPUT

```
DayOfWeek = *
DayofMonth = *
IPAddress = 172.23.200.26
ImageName = backupfile_R620
```
MaxNumberOfBackupArchives = 12
Repeat = 12
ReturnValue = 4096
ShareName = cifs
ShareType = cifs
Time = 17:20
Username = administrator
WeekofMonth = *

If you do not want the next scheduled job to execute, invoke the **DeleteJobQueue** method in the **DCIM_JobService** class.

**WSMAN Command and Output Example:**

```plaintext
winrm invoke DeleteJobQueue
cimv2/root/dcim/DCIM_JobService?CreationClassName=DCIM_JobService+Name=JobService+SystemName=Idrac+SystemCreationClassName=DCIM_ComputerSystem
@{JobID="JID_887051062764" } -u:root -p:calvin -r:https://192.168.0.120/wsman:443 -SkipCNCheck -SkipCACheck -auth:basic -encoding:utf-8
DeleteJobQueue_OUTPUT
Message = The specified job was deleted
MessageID = SUP020
ReturnValue = 0
```

If you want to disable the auto backup feature, use the **SetAttributes** and **CreateConfigJob** methods to set the **Automatic Backup Feature** attribute to **Disabled**.

**WSMAN Command and Output Examples:**

```plaintext
winrm g http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/root/dcim/DCIM_LCEnumeration?InstanceID=LifecycleController.Embedded.1#LCAttributes.1#AutoBackup
-u:root -p:calvin -r:https://192.168.0.120/wsman -SkipCNcheck -SkipCACheck -encoding:utf-8 -a:basic
DCIM_LCEnumeration
AttributeName = Automatic Backup Feature
CurrentValue = Enabled
DefaultValue = Disabled
ElementName = LC.emb.1
InstanceId = LifecycleController.Embedded.1#LCAttributes.1#AutoBackup
IsReadOnly = false
PendingValue = null
PossibleValues = Disabled, Enabled
```

```plaintext
winrm i SetAttributes 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://192.168.0.120/wsman -SkipCNcheck -SkipCACheck -encoding:utf-8 -a:basic
@{Target="LifecycleController.Embedded.1";AttributeName="Automatic Backup Feature";AttributeValue="Disabled"}
SetAttributes_OUTPUT
```
Message = The command was successful
MessageID = LC001
RebootRequired = No
ReturnValue = 0
SetResult = Set PendingValue

Ø{Target="LifecycleController.Embedded.1"}
CreateConfigJob_OUTPUT

Job

EndpointReference
Address = http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
ReferenceParameters
        Selector: InstanceID = JID_887067839172, __cimnamespace = root/dcim
Returnvalue = 4096

DCIM_LCEnumeration
    AttributeName = Automatic Backup Feature
    CurrentValue = Disabled
    DefaultValue = Disabled
    ElementName = LC.emb.1
    InstanceID = LifecycleController.Embedded.1#LCAttributes.1#AutoBackup
    IsReadOnly = false
    PendingValue = null
    PossibleValues = Disabled, Enabled
Auto Backup Workflow Using RACADM

In this workflow example, I’ll be using RACADM to schedule auto backups to a CIFS share using only the required parameters.

1. Launch a SSH session to the iDRAC.
2. At the admin prompt, type racadm. The RACADM interactive mode is displayed.
3. At the racadm prompt, type ”get LifeCycleController.lcattributes.autobackup” to view the current value. It must be set to Enabled to use the auto backup feature. If set to Disabled, type racadm set lifecyclecontroller.lcattributes.autobackup Enabled to enable the feature.

RACADM Command Example and Output:

```
racadm>>get LifecycleController.lcattributes.autobackup
racadm get LifecycleController.lcattributes.autobackup
Disabled

racadm>>set LifecycleController.lcattributes.autobackup Enabled
racadm set LifecycleController.lcattributes.autobackup Enabled
Object value modified successfully

racadm>>get LifecycleController.lcattributes.autobackup
racadm get LifecycleController.lcattributes.autobackup
Enabled
```  
4. Type getractime to view the current iDRAC time. Now type ”systemconfig backup -f backupfile_RACADM -l //172.23.200.26/cifs -u administrator -p password -time 11:10 -dow "*" -rp 1 -mb 1” which will return a success message. For the racadm options, you want to pass in your CIFS credentials. If needed, type ”systemconfig help backup” which will return help information for this command. It will explain all the options along with showing some examples of executing this command.

RACADM Command Example and Output:

```
racadm>>systemconfig backup -f backupfile_RACADM -l //172.23.200.26/cifs -u administrator -p password123 -time 11:10 -dow "*" -rp 1 -mb 1
racadm systemconfig backup -f backupfile_RACADM -l //172.23.200.26/cifs -u administrator -p password123 -time 11:10 -dow "*" -rp 1 -mb 1
RAC1049: Successfully configured the Automatic Backup (autobackup) feature settings.
```  
5. To view the auto backup job created, type jobqueue view.
6. Type systemconfig GetBackupScheduler to view information of the current scheduled auto backup.
RACADM Command Example and Output:

```
racadm>>jobqueue view
racadm jobqueue view
----------------------------------------------------------
[Job ID=JID_887067839172]
Job Name=Configure: LC.Embedded.1
Status=Completed
Start Time=[Not Applicable]
Expiration Time=[Not Applicable]
Message=[LC001: The command was successful]
----------------------------------------------------------
[Job ID=JID_887686226964]
Job Name=Automatic Backup:Image
Status=Ready For Backup
Start Time=[Fri, 03 Jan 2014 11:10:00]
Expiration Time=[Fri, 03 Jan 2014 11:15:00]
```

The auto backup job is initiated after the specified start time elapses. Type `jobqueue view` to monitor the progress of the backup job (backup may take up to 20-30 minutes depending on system hardware configuration).

RACADM Command and Output Example:

```
racadm>>jobqueue view
racadm jobqueue view
----------------------------------------------------------
[Job ID=JID_887067839172]
Job Name=Configure: LC.Embedded.1
Status=Completed
Start Time=[Not Applicable]
Expiration Time=[Not Applicable]
Message=[LC001: The command was successful]
----------------------------------------------------------
[Job ID=JID_887686226964]
Job Name=Automatic Backup:Image
Status=Backup In Progress
Start Time=[Fri, 03 Jan 2014 11:10:00]
Expiration Time=[Fri, 03 Jan 2014 11:15:00]
```
You can query the job queue until the job is marked as completed. While querying the job queue, you will notice that a new auto backup job is created for the next scheduled time. In the example, you will notice that the same time is reported with a different date.

**RACADM Command and Output Example:**

```
racadm>>jobqueue view
racadm jobqueue view
-------------------------JOB QUEUE------------------------
[Job ID=JID_887067839172]
Job Name=Configure: LC.Embedded.1
Status=Completed
Start Time=[Not Applicable]
Expiration Time=[Not Applicable]
Message=[LC001: The command was successful]

[Job ID=JID_887686226964]
Job Name=Automatic Backup:Image
Status=Completed
Start Time=[Fri, 03 Jan 2014 11:10:00]
Expiration Time=[Fri, 03 Jan 2014 11:15:00]
Message=[BAR007: Export System Profile completed.]

[Job ID=JID_887693048713]
Job Name=Automatic Backup:Image
Status=Ready For Backup
Start Time=[Sat, 04 Jan 2014 11:10:00]
Expiration Time=[Sat, 04 Jan 2014 11:15:00]
```

7. Go to your CIFS network share and you will notice a directory named "Archive_"Your Service Tag" was created. Inside that directory will be the auto backup file. You should see "_1" added to the end of the file name (Example: 'R620_backup_file_1'). Now every time the auto backup process hits the scheduled start time, backup files will be created and stored in this same directory.

If you do not want the next auto backup job to execute, type `jobqueue delete -I JID_xxxxxxxxxxxxxx` to delete a specific job from the job queue.

**RACADM Command and Output Example:**

```
racadm>>jobqueue delete -i JID_887693048713
racadm jobqueue delete -i JID_887693048713
RAC1032: JID_887693048713 job(s) was cancelled by the user.
```

If you want to disable the auto backup feature, use the `set` command.

26
RACADM Command and Output Example:

```bash
racadm>>set LifecycleController.lcattributes.autobackup Disabled
racadm set LifecycleController.lcattributes.autobackup Disabled
Object value modified successfully

racadm>>get LifecycleController.lcattributes.autobackup
racadm get LifecycleController.lcattributes.autobackup
Disabled
```
Auto Update Workflow Using iDRAC GUI

In this workflow example, I’ll be using the iDRAC GUI to schedule auto updates using a custom repository which was created using Dell Repository Manager.

1. Launch the iDRAC GUI.
2. Navigate to **Overview** -> **iDRAC Settings** -> **Update and Rollback**.
3. On the **Firmware Update** page, click **Automatic Update**.
4. Verify if the **Enable Automatic Update** checkbox is selected. If not, select the checkbox to enable the auto update feature.
   
   **Note:** Clear the **Enable Automatic Update** checkbox to disable the auto update feature.

5. Select any of the following **Server Reboot** options to specify if a system reboot is required after the updates are staged:
   - **Schedule Updates** – Stage the firmware updates but do not reboot the server.
   - **Schedule Updates and Reboot Server** – Enables server reboot after the firmware updates are staged.

6. Select any one of the following **File Location** options to specify the location of the firmware images:
   - **Network**
   - **FTP**
   - **TFTP**
   - **HTTP**

7. Based on the selection in step 6, enter the following under **Catalog Location**:
   - **Protocol** – Select the network share (**CIFS** or **NFS**).
   - **IP Address** – Enter the IP address or the host name of the shared folder.
   - **Share Name** – Enter the name of the shared folder.
   - **Domain Name** – Enter the domain name required to login to the network share.
   - **User Name** – Enter the user name required to login to the network share.
   - **Password** – Enter the password required to authenticate the user name.
   - **Catalog location (optional)** – Specify the location of the catalog.xml file.
   - **Catalog name (optional)** – Specify the name of the catalog file.

   **Note:** For the **Catalog Location**, you must either pass in the root directory or the sub directory. For example, a Windows CIFS share named repo is created in the root of the C drive and this share is the custom repository which contains the Dell Update Packages.
and the catalog.xml file. In this case, you must enter the share name as repo and the catalog location as /. However, in case if the catalog.xml file is placed in a sub directory in the directory named repo, then the share name is repo and the catalog location is /catalog.

8. Click Test network configuration to verify if the network share is accessible using the specified user name and password.

9. In the Update Window Schedule section, enter the following:
   - **Start (24hr format)** - Enter the time for the firmware update operation to start.

10. In the Recurrence Pattern section, enter the following details to set the recurrence pattern for the automatic update:
   - **Daily** – Select this option to schedule an update once every x number of days. The valid values are 1-365.
   - **Weekly** – Select this option to schedule weekly updates. Enter the week number in the Recur every [1–52] week(s) on field and then select the day of the week. For example, Recur every 10 week(s) on Monday.
   - **Monthly** – Select this option to schedule monthly updates based on the following:
     - Day [1–31] of every [1–12] month(s). For example, Day 4 of every 11 months.
     - The [day number] [day of week] of every [1–12] month(s). For example, The Last Sunday of every 10 months.

11. Click Schedule Update. A pop-up window with a success message is displayed.

12. To view the status of the job, select Overview -> Server -> Job Queue to view the status of the job created.

   After the scheduled start time elapses, the job status changes from Scheduled to Downloaded. Refresh the page to view the new job IDs created for the devices from your custom repository. For devices that are immediate updates such as diagnostics, USC, driver pack, and so on, the status of the job displays Completed. For the updates that require a host reboot such as BIOS, NIC, RAID, and so on, the status of the job displays as Scheduled. After all job IDs for updates are either marked completed or scheduled, the repository update job is marked as Completed and a new repository job is created to execute at the next scheduled time along with a reboot job ID. The host then reboots to enter Lifecycle Controller to execute the update tasks (if the repository update only contained devices that are immediate updates, then the host will not reboot). After all tasks are done executing, the host will reboot. After the system has completed POST and Collect System Inventory, refresh the Job Queue page. The status of all the device job IDs are displayed as Completed.

13. To view the list of devices updated, select Overview -> Server -> Logs -> Lifecycle Log to view the Lifecycle Controller logs. Select Updates from the Log Type drop-down box to view only the update log entries. A list of all the devices that are updated is displayed. For example, if the BIOS is updated, the Lifecycle Controller log will contain a log entry displaying the BIOS information along with the previous and current versions.
14. If you want to cancel the next auto update job, then select the checkbox next to the job ID and click **Delete** at the bottom of the page.

iDRAC GUI image examples(covers steps 1 thru 8 mentioned above):

Steps 1 - 6:
Tasks are running normally.

Do not reboot or power off the system until system reboots on its own.
Step 7:
Auto Update Workflow Using WS-MAN

In this workflow example, I’ll be using WSMAN to schedule auto updates using a custom repository which has already been created using Dell Repository Manager.

1. Run a get on the `DCIM_LCEnumeration` class passing the `LifecycleController.Embedded.1#LCAttributes.1#AutoUpdate` attribute. Verify that `CurrentValue` is set to “Enabled” (if not set to enabled, use SetAttributes and CreateConfigJob methods to change the current value (refer to step 11 below which shows an example of using these methods).

**WSMAN Command and Output Example:**

```
```

```
DCIM_LCEnumeration
  AttributeName = Automatic Update Feature
  CurrentValue = Enabled
  DefaultValue = Enabled
  ElementName = LC.emb.1
  InstanceID = LifecycleController.Embedded.1#LCAttributes.1#AutoUpdate
  IsReadOnly = false
  PendingValue = null
  PossibleValues = Disabled, Enabled
```

2. Invoke the `ManageTime` method in the `DCIM_TimeService` class to view the iDRAC current time.

**WSMAN Command and Output Example:**

```
winrm i ManageTime
"cimv2/root/dcim/DCIM_TimeService?SystemCreationClassName=DCIM_SPComputerSystem+SystemName=systemmc+CreationClassName=DCIM_TimeService+Name=DCIM_TimeService 1" @{GetRequest="true"} -u:root -p:calvin -r:https://192.168.0.120/wsman -SkipCNcheck -SkipCAcheck -encoding:utf-8 -a:basic
```

```
ManageTime_OUTPUT
  Message = The command was successful
  MessageID = RAC048
  ReturnValue = 0
  TimeData = 20140103132453.000000-360
```

3. Using your CIFS network credentials which contains the custom repository, pass in these values for the parameters and invoke the `SetUpdateSchedule` method on the `DCIM_LCService` class:

- IPAddress
- ShareName
- ShareType
- Username
- Password
- Time
- Repeat
- ApplyReboot

**WSMAN Command and Output Example:**

```bash
winrm invoke SetUpdateSchedule
cimv2/root/dcim/DCIM_SoftwareInstallationService?CreationClassName=DCIM_SoftwareInstallationService+SystemCreationClassName=DCIM_ComputerSystem+SystemName =IDRAC:ID+Name=SoftwareUpdate-u:root -p:calvin -r:https://192.168.0.120/wsman:443 -SkipCNCheck -SkipCACheck -auth:basic -encoding:utf-8
@{IPAddress="172.23.200.26";ShareType="2";ShareName="madone_repository2";Username="administrator";Password="password123";Time="15:15";Repeat="1";ApplyReboot="1"}
SetUpdateSchedule_OUTPUT
  ReturnValue = 4096
```

4. Run an enum on **DCIM_LifecycleJob** class to view the job queue. An auto update job ID with the job start time that you specified is displayed.

**WSMAN Command and Output Example:**

```bash
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
```
5. After the scheduled start time elapses, the job status changes from *Scheduled* to *Downloaded*. Refresh the page to view the new job IDs created for the devices from your custom repository. For devices that are immediate updates such as diagnostics, USC, driver pack, and so on, the status of the job displays *Completed*. For the updates that require a host reboot such as BIOS, NIC, RAID, and so on, the status of the job displays as *Scheduled*. After all job IDs for updates are either marked completed or scheduled, the repository update job is marked as *Completed* and a new repository job is created to execute at the next scheduled time along with a reboot job ID. The host then reboots to enter Lifecycle Controller to execute the update tasks (if the repository update only contained devices that are immediate updates, then the host will not reboot). After all tasks are done executing, the host will reboot.

**WSMAN Command and Output Example:**

```
```

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

```plaintext
DCIM_LifecycleJob
ElapsedTimeSinceCompletion = 1
InstanceID = JID_887834662218
JobStartime = 20140103151500
JobStatus = Downloaded
JobUntilTime = 20140103152000
Message = Package successfully downloaded.
MessageArguments = NA
MessageID = REDO02
```
Name = Repository Update
PercentComplete = NA

DCIM_LifecycleJob
ElapsedTimeSinceCompletion = null
InstanceId = JID_887837513841
JobStartTime = NA
JobStatus = Downloaded
JobUntilTime = NA
Message = Package successfully downloaded.
MessageArguments = NA
MessageID = RED002
Name = update:DCIM:INSTALLED#741__BIOS.Setup.1-1
PercentComplete = NA

DCIM_LifecycleJob
ElapsedTimeSinceCompletion = null
InstanceId = JID_887837772139
JobStartTime = NA
JobStatus = Downloading
JobUntilTime = NA
Message = Downloading package.
MessageArguments = NA
MessageID = RED003
Name = update:DCIM:INSTALLED#802__USC.Embedded.1:LC.Embedded.1
PercentComplete = NA


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 = 10
InstanceId = JID_887834662218
JobStartTime = 20140103151500
JobStatus = Completed
JobUntilTime = 20140103152000
Message = Job completed successfully.
MessageArguments = NA
MessageID = RED001
Name = Repository Update
PercentComplete = 100
DCIM_LifecycleJob
ElapsedTimeSinceCompletion = null
InstanceID = JID_887837513841
JobStartTime = TIME_NOW
JobStatus = Scheduled
JobUntilTime = TIME_NA
Message = Task successfully scheduled.
MessageArguments = NA
MessageID = JCP001
Name = update:DCIM:INSTALLED#741__BIOS.Setup.1-1
PercentComplete = NA

DCIM_LifecycleJob
ElapsedTimeSinceCompletion = 8
InstanceID = JID_887837772139
JobStartTime = NA
JobStatus = Completed
JobUntilTime = NA
Message = Job completed successfully.
MessageArguments = NA
MessageID = RED001
Name = update:DCIM:INSTALLED#802__USC.Embedded.1:LC.Embedded.1
PercentComplete = 100

DCIM_LifecycleJob
ElapsedTimeSinceCompletion = null
InstanceID = JID_887838340497
JobStartTime = TIME_NOW
JobStatus = Scheduled
JobUntilTime = TIME_NA
Message = Task successfully scheduled.
MessageArguments = NA
MessageID = JCP001
Name = update:DCIM:INSTALLED#701__NIC.Slot.1-1-1
PercentComplete = NA

DCIM_LifecycleJob
ElapsedTimeSinceCompletion = 7
InstanceID = JID_887838642881
JobStartTime = NA
JobStatus = Completed
JobUntilTime = NA
Message = Job completed successfully.
MessageArguments = NA
MessageID = RED001
Name = update:DCIM:INSTALLED#802__Diagnostics.Embedded.1:LC.Embedded.1
PercentComplete = 100

DCIM_LifecycleJob
ElapsedTimeSinceCompletion = null
InstanceID = JID_88783800701
JobStartTime = TIME_NOW
JobStatus = Scheduled
JobUntilTime = TIME_NA
Message = Task successfully scheduled.
6. After the system has completed POST and Collect System Inventory, refresh the job queue. The status of the device job IDs is displayed as **Completed**.

**WSMAN Command and Output Example:**

```
```

DCIM LifecycleJob
- ElapsedTimeSinceCompletion = null
- InstanceID = JID_CLEARALL
- JobStartTime = TIME_NOW
- JobStatus = Pending
- JobUntilTime = TIME_NA
- Message = NA
- MessageArguments = NA
- MessageID = NA
- Name = CLEARALL
- PercentComplete = 0

DCIM LifecycleJob
- ElapsedTimeSinceCompletion = 22
- InstanceID = JID_887834662218
- JobStartTime = 20140103151500
- JobStatus = Completed
JobUntilTime = 20140103152000
Message = Job completed successfully.
MessageArguments = NA
MessageID = RED001
Name = Repository Update
PercentComplete = 100

DCIM_LifecycleJob
ElapsedTimeSinceCompletion = 0
InstanceID = JID_887837513841
JobStartTime = TIME_NOW
JobStatus = Completed
JobUntilTime = TIME_NA
Message = Job completed successfully.
MessageArguments = NA
MessageID = PR19
Name = update:DCIM:INSTALLED#741__BIOS.Setup.1-1
PercentComplete = 100

DCIM_LifecycleJob
ElapsedTimeSinceCompletion = 20
InstanceID = JID_887837772139
JobStartTime = NA
JobStatus = Completed
JobUntilTime = NA
Message = Job completed successfully.
MessageArguments = NA
MessageID = RED001
Name = update:DCIM:INSTALLED#802__USC.Embedded.1:LC.Embedded.1
PercentComplete = 100

DCIM_LifecycleJob
ElapsedTimeSinceCompletion = 0
InstanceID = JID_887838340497
JobStartTime = TIME_NOW
JobStatus = Completed
JobUntilTime = TIME_NA
Message = Job completed successfully.
MessageArguments = NA
MessageID = PR19
Name = update:DCIM:INSTALLED#701__NIC.Slot.1-1-1
PercentComplete = 100

DCIM_LifecycleJob
ElapsedTimeSinceCompletion = 19
InstanceID = JID_887838642881
JobStartTime = NA
JobStatus = Completed
JobUntilTime = NA
Message = Job completed successfully.
MessageArguments = NA
MessageID = RED001
Name = update:DCIM:INSTALLED#802__Diagnostics.Embedded.1:LC.Embedded.1
PercentComplete = 100
7. To verify if current firmware versions are installed for all devices on the host, run an enum on the DCIM_SoftwareIdentity class. This output will report the current installed and previous version for each device.

**WSMAN Command Output (part of the output only showing BIOS):**

```
```

```
DCIM_SoftwareIdentity
 BuildNumber = 0
 Classifications = 11
 ComponentID = 159
 ComponentType = BIOS
 DeviceID = null
 ElementName = BIOS
 FQDD = BIOS.Setup.1-1
 IdentityInfoType = OrgID:ComponentType:ComponentID
```
IdentityInfoValue = DCIM:BIOS:159
InstallationDate = NA
InstanceID = DCIM:PREVIOUS#741__BIOS.Setup.1-1
IsEntity = true
MajorVersion = 2
MinorVersion = 0
RevisionNumber = 15
RevisionString = null
Status = Available
SubDeviceID = null
SubVendorID = null
Updateable = true
VendorID = null
VersionString = 2.0.14
impactsTPMmeasurements = true

DCIM_SoftwareIdentity
BuildNumber = 0
Classifications = 11
ComponentID = 159
ComponentType = BIOS
DeviceID = null
ElementName = BIOS
FQDD = BIOS.Setup.1-1
IdentityInfoType = OrgID:ComponentType:ComponentID
IdentityInfoValue = DCIM:BIOS:159
InstallationDate = NA
InstanceID = DCIM:CURRENT#741__BIOS.Setup.1-1
IsEntity = true
MajorVersion = 2
MinorVersion = 0
RevisionNumber = 15
RevisionString = null
Status = Available
SubDeviceID = null
SubVendorID = null
Updateable = true
VendorID = null
VersionString = 2.0.15
impactsTPMmeasurements = true

DCIM_SoftwareIdentity
BuildNumber = 0
Classifications = 11
ComponentID = 159
ComponentType = BIOS
DeviceID = null
ElementName = BIOS
FQDD = BIOS.Setup.1-1
IdentityInfoType = OrgID:ComponentType:ComponentID
IdentityInfoValue = DCIM:BIOS:159
InstallationDate = 2013-12-06T16:39:05Z
InstanceID = DCIM:INSTALLED#741__BIOS.Setup.1-1
IsEntity = true
MajorVersion = 2
8. You can also view the Lifecycle Controller logs, to verify the list of devices that are updated. Run an enum on the `DCIM_LCLogEntry` class. Use the WS-MAN filtering option to filter out the update entries. The example below shows how to filter the Category = Updates, to display only the update entries in the Lifecycle Controller log.

**WSMAN Command and Output Example (part of the output only showing a BIOS update entry):**

```
```

<table>
<thead>
<tr>
<th>DCIM_LCLogEntry</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>AgentID</td>
<td>UEFI SS USC</td>
</tr>
<tr>
<td>Category</td>
<td>Updates</td>
</tr>
<tr>
<td>Comment</td>
<td>[set comment here]</td>
</tr>
<tr>
<td>ConfigResultsAvailable</td>
<td>false</td>
</tr>
<tr>
<td>CreationTimeStamp</td>
<td>20131206163559.000000-360</td>
</tr>
<tr>
<td>ElementName</td>
<td>SUP0518</td>
</tr>
<tr>
<td>FQDD</td>
<td>USC.Embedded.1:LC.Embedded.1</td>
</tr>
<tr>
<td>InstanceID</td>
<td>DCIM:LifeCycleLog:20784</td>
</tr>
<tr>
<td>LogInstanceID</td>
<td>DCIM:LifeCycleLog</td>
</tr>
<tr>
<td>LogName</td>
<td>LifeCycle Log</td>
</tr>
<tr>
<td>Message</td>
<td>Successfully updated the Server BIOS 12G firmware to version 2.0.15</td>
</tr>
<tr>
<td>MessageArguments</td>
<td>Server BIOS 12G, 2.0.15</td>
</tr>
<tr>
<td>MessageID</td>
<td>SUP0518</td>
</tr>
<tr>
<td>OwningEntity</td>
<td>DCIM</td>
</tr>
<tr>
<td>PerceivedSeverity</td>
<td>2</td>
</tr>
<tr>
<td>RawEventData</td>
<td>null</td>
</tr>
<tr>
<td>RecordID</td>
<td>20784</td>
</tr>
<tr>
<td>SequenceNumber</td>
<td>20784</td>
</tr>
</tbody>
</table>

9. To check auto update information, invoke the `GetUpdateSchedule` method on the `DCIM_LCService` class. This command returns the parameter information that you passed while invoking the `SetUpdateSchedule` method.
WSMAN Command and Output Example:

```
winrm invoke GetUpdateSchedule
  cimv2/root/dcim/DCIM_SoftwareInstallationService?CreationClassName=DCIM_SoftwareInstallationService+SystemCreationClassName=DCIM_ComputerSystem+SystemName=IDRAC:ID+Name=SoftwareUpdate -u:root -p:calvin -r:https://192.168.0.120/wsman:443 -SkipCNCheck -SkipCACheck -auth:basic -encoding:utf-8

GetUpdateSchedule_OUTPUT
  ApplyReboot = 1
  IPAddress = 172.23.200.26
  Repeat = 1
  ReturnValue = 4096
  ShareName = madone_repository
  ShareType = cifs
  Time = 16:15
  Username = administrator
```

10. If you do not want the next scheduled job to execute, invoke the **DeleteJobQueue** method on the **DCIM_JobService** class.

**WSMAN Command and Output Example:**

```
winrm invoke DeleteJobQueue
  cimv2/root/dcim/DCIM_JobService?CreationClassName=DCIM_JobService+Name=JobService+SystemName=Idrac+SystemCreationClassName=DCIM_ComputerSystem
  @{JobID="JID_887051062764" } -u:root -p:calvin -r:https://192.168.0.120/wsman:443 -SkipCNCheck -SkipCACheck -auth:basic -encoding:utf-8

DeleteJobQueue_OUTPUT
  Message = The specified job was deleted
  MessageID = SUP020
  ReturnValue = 0
```

To disable the auto update feature, use the **SetAttributes** and **CreateConfigJob** methods to set the **Automatic Update Feature** attribute to disabled.

**WSMAN Command and Output Examples:**

```
winrm g http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/root/dcim/DCIM_LCEnumeration?InstanceID=LifecycleController.Embedded.1#LCAttributes.1#AutoUpdate
  -u:root -p:calvin -r:https://192.168.0.120/wsman
  -SkipCNcheck -SkipCAcheck -encoding:utf-8 -a:basic

DCIM_LCEnumeration
  AttributeName = Automatic Update Feature
  CurrentValue = Enabled
  DefaultValue = Disabled
  ElementName = LC.emb.1
  InstanceID = LifecycleController.Embedded.1#LCAttributes.1#AutoUpdate
  IsReadOnly = false
  PendingValue = null
  PossibleValues = Disabled, Enabled
```
winrm i SetAttributes 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:LC
Service -u:root -p:calvin -r:https://192.168.0.120/wsman -SkipCNcheck -
SkipCAcheck -encoding:utf-8 -a:basic
@{Target="LifecycleController.Embedded.1";AttributeName="Automatic Update
Feature";AttributeValue="Disabled"}
SetAttributes_OUTPUT
Message = The command was successful
MessageID = LC001
RebootRequired = No
ReturnValue = 0
SetResult = Set PendingValue

winrm i CreateConfigJob 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:LC
Service -u:root -p:calvin -r:https://192.168.0.120/wsman -SkipCNcheck -
SkipCAcheck -encoding:utf-8 -a:basic
@{Target="LifecycleController.Embedded.1"}
CreateConfigJob_OUTPUT
Job
EndpointReference
Address =
http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
ReferenceParameters
ResourceURI = http://schemas.dell.com/wbem/wscim/1/cim-
schema/2/DCIM_LifecycleJob SelectorSet
Selector: InstanceID = JID_887067839172, __cimnamespace =
root/dcim
ReturnValue = 4096

DCIM_LCEnumeration
AttributeName = Automatic Update Feature
CurrentValue = Disabled
DefaultValue = Disabled
ElementName = LC.emb.1
InstanceID = LifecycleController.Embedded.1#LCAttributes.1#AutoUpdate
IsReadOnly = false
PendingValue = null
PossibleValues = Disabled, Enabled
Auto Update Workflow Using RACADM

In this workflow example, I’ll be using RACADM to schedule auto updates using a custom repository which was created earlier using Dell Repository Manager.

1. Launch a SSH session in iDRAC.
2. At the admin prompt, type racadm. The RACADM interactive mode is displayed.
3. At the racadm prompt, type getLifeCycleController.lcattributes.autoupdate to view the current value. If set to Disabled, type racadm set lifecyclecontroller.lcattributes.autoupdate Enabled, to enable the feature.

**RACADM Command Example and Output:**

```
racadm>>get LifecycleController.lcattributes.autoupdate
racadm get LifecycleController.lcattributes.autoupdate
Disabled
```
```
racadm>>set LifecycleController.lcattributes.autoupdate Enabled
racadm set LifecycleController.lcattributes.autoupdate Enabled
Object value modified successfully
```
```
racadm>>get LifecycleController.lcattributes.autoupdate
racadm get LifecycleController.lcattributes.autoupdate
Enabled
```

4. Type getractime to view the current iDRAC time.
5. Type AutoUpdateScheduler create -l //192.168.0.120/repository -u username -p password -time 16:46 -rp 1 -a 1 -dow "*" " which will return a success message. For the racadm options, you want to pass in your CIFS credentials. If needed, type help AutoUpdateScheduler which will return help information for this command. It will explain all the options along with showing some examples of executing this command.

**RACADM Command Example and Output:**

```
racadm>>getractime
racadm getractime
Fri Jan  3 16:42:24 2014
```
```
racadm>>AutoUpdateScheduler create -l //172.23.200.26/repository -u administrator -p password123 -time 16:46 -rp 1 -a 1 -dow "*"
```
```
racadm AutoUpdateScheduler create -l //172.23.200.26/repository -u administrator -p password123 -time 16:46 -rp 1 -a 1 -dow "*"
```
```
RAC1041: Successfully configured the Automatic Update (autoupdate) feature settings.
```

6. Type jobqueue view which will return your job queue. You will notice that a new repository update job was created with the correct start time as you just passed in from the previous command. Next type “autoupdatescheduler view” which will return all information about the scheduled auto update.
RACADM Command Example and Output:

```
racadm>>jobqueue view
racadm jobqueue view
-------------------------JOB QUEUE-------------------------
[Job ID=JID_887889621701]
Job Name=Repository Update
Status=Scheduled
Start Time=[Next Reboot]
Expiration Time=[Fri, 03 Jan 2014 16:51:00]
Message=[RED061: The job is successfully scheduled.]

racadm>>autoupdatescheduler view
racadm autoupdatescheduler view
hostname                  =172.23.200.26
sharename                 =madone_repository2
sharetype                 =cifs
time                      =16:46
dayofweek                 =*
repeat                    =1
applyreboot               =0
idracuser                 =racuser
username                  =administrator
```

7. Once the scheduled start time is hit, the job status will change to “Downloaded”. Continue to query the job queue executing the same command. You should start seeing new job IDs getting created for the devices from your custom repository. For the devices that are immediate updates (Examples: DIAGS, USC, Driver Pack), you will see a “Completed” job status. For the devices that require a host reboot (Examples: BIOS, NIC, RAID), you will see a “Scheduled” job status. Once all job IDs for devices are either marked completed or scheduled, the repository update job will be marked as “Completed”. You will also notice that a new repository job is created which will execute at the next scheduled time along with a reboot job ID. The host will reboot to enter Lifecycle Controller to execute the update tasks(if your repository update only contained devices that are immediate updates, then the host will not reboot). Once all tasks are done executing, the host will reboot.

RACADM Command and Output Example:

```
racadm>>jobqueue view
racadm jobqueue view
-------------------------JOB QUEUE-------------------------
[Job ID=JID_887889621701]
Job Name=Repository Update
Status=Downloaded
Start Time=[Next Reboot]
Expiration Time=[Fri, 03 Jan 2014 16:51:00]
```
<table>
<thead>
<tr>
<th>Job ID</th>
<th>Job Name</th>
<th>Status</th>
<th>Start Time</th>
<th>Expiration Time</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>JID_887892142558</td>
<td>Firmware Update: BIOS</td>
<td>Downloaded</td>
<td>[Not Applicable]</td>
<td>[Not Applicable]</td>
<td>[RED002: Package successfully downloaded.]</td>
</tr>
<tr>
<td>JID_887892351852</td>
<td>Firmware Update: Lifecycle Controller</td>
<td>Completed</td>
<td>[Not Applicable]</td>
<td>[Not Applicable]</td>
<td>[RED001: Job completed successfully.]</td>
</tr>
<tr>
<td>JID_887892866760</td>
<td>Firmware Update: NIC</td>
<td>Downloaded</td>
<td>[Not Applicable]</td>
<td>[Not Applicable]</td>
<td>[RED002: Package successfully downloaded.]</td>
</tr>
<tr>
<td>JID_887893221541</td>
<td>Firmware Update: Diagnostics</td>
<td>Completed</td>
<td>[Not Applicable]</td>
<td>[Not Applicable]</td>
<td>[RED001: Job completed successfully.]</td>
</tr>
</tbody>
</table>

racadm>>jobqueue view
racadm jobqueue view
-------------------------JOB QUEUE-------------------------
<table>
<thead>
<tr>
<th>Job ID</th>
<th>Job Name</th>
<th>Status</th>
<th>Start Time</th>
<th>Expiration Time</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>JID_88789621701</td>
<td>Repository Update</td>
<td>Completed</td>
<td>[Next Reboot]</td>
<td>[Fri, 03 Jan 2014 16:51:00]</td>
<td>[RED001: Job completed successfully.]</td>
</tr>
<tr>
<td>JID_887892142558</td>
<td>Firmware Update: BIOS</td>
<td>Scheduled</td>
<td>[Next Reboot]</td>
<td>[Not Applicable]</td>
<td>[JCP001: Task successfully scheduled.]</td>
</tr>
<tr>
<td>JID_887892351852</td>
<td>Firmware Update: Lifecycle Controller</td>
<td>Completed</td>
<td>[Not Applicable]</td>
<td>[Not Applicable]</td>
<td></td>
</tr>
</tbody>
</table>
8. Once the system has completed POST and Collecting Inventory, query the job queue again which now all update job IDs should be marked as “Completed”.

**RACADM Command and Output Example:**

```
racadm>>jobqueue view
racadm jobqueue view
-------------------------JOB QUEUE-------------------------
[Job ID=JID_887892866760]
Job Name=Firmware Update: NIC
Status=Scheduled
Start Time=[Next Reboot]
Expiration Time=[Not Applicable]
Message=[JCP001: Task successfully scheduled.]

[Job ID=JID_887893221541]
Job Name=Firmware Update: Diagnostics
Status=Completed
Start Time=[Not Applicable]
Expiration Time=[Not Applicable]
Message=[RED001: Job completed successfully.]

[Job ID=JID_887893377437]
Job Name=Firmware Update: RAID
Status=Scheduled
Start Time=[Next Reboot]
Expiration Time=[Not Applicable]
Message=[JCP001: Task successfully scheduled.]

[Job ID=JID_887894622089]
Job Name=Repository Update
Status=Scheduled
Start Time=[Next Reboot]
Expiration Time=[Fri, 03 Jan 2014 16:51:00]
Message=[RED001: Job completed successfully.]

[Job ID=RID_887896302796]
Job Name=Reboot: Power cycle
Status=Reboot Completed
Start Time=[Now]
Expiration Time=[Not Applicable]
Message=[RED030: Reboot is complete.]
```
9. Type `lclog view -c updates -n 20` to view the last 20 update entries in Lifecycle log. If needed, run "lclog help view" to see all options.
RACADM Command and Output Example:

```
racadm>1clog view -c updates -n 20
racadm1clog view -c updates -n 20
SeqNumber  = 21502
Message ID = SUP0518
Category    = Updates
AgentID     = UEFI_SS_USC
Severity    = Information
Timestamp   = 2014-01-03 17:03:01
Message     = Successfully updated the 5720 DP 1G ADAPTER firmware to version 7.6.15.
Message Arg 1 = 5720 DP 1G ADAPTER
Message Arg 2 = 7.6.15
FQDD        = USC.Embedded.1:LC.Embedded.1

SeqNumber  = 21501
Message ID = SUP0516
Category    = Updates
AgentID     = UEFI_SS_USC
Severity    = Information
Timestamp   = 2014-01-03 17:01:37
Message     = Updating firmware for 5720 DP 1G ADAPTER to version 7.6.15.
Message Arg 1 = 5720 DP 1G ADAPTER
Message Arg 2 = 7.6.15
FQDD        = USC.Embedded.1:LC.Embedded.1

SeqNumber  = 21500
Message ID = SUP0518
Category    = Updates
AgentID     = UEFI_SS_USC
Severity    = Information
Timestamp   = 2014-01-03 17:01:24
Message     = Successfully updated the PERC H310 Mini Monolithic firmware to version 20.12.0-0004.
Message Arg 1 = PERC H310 Mini Monolithic
Message Arg 2 = 20.12.0-0004
FQDD        = USC.Embedded.1:LC.Embedded.1

SeqNumber  = 21499
Message ID = SUP0516
Category    = Updates
AgentID     = UEFI_SS_USC
Severity    = Information
Timestamp   = 2014-01-03 16:59:18
Message     = Updating firmware for PERC H310 Mini Monolithic to version 20.12.0-0004.
Message Arg 1 = PERC H310 Mini Monolithic
Message Arg 2 = 20.12.0-0004
FQDD        = USC.Embedded.1:LC.Embedded.1
```
10. If you do not want the next auto backup job to execute, type `jobqueue delete -i JID_xxxxxxxxxxxxxx` to delete the specific job from the job queue.

**RACADM Command and Output Example:**

```
racadm>>jobqueue delete -i JID_887693048713
racadm jobqueue delete -i JID_887693048713
RAC1032: JID_887693048713 job(s) was cancelled by the user.
```

11. If you want to disable the auto update feature, use the `set` command.

**RACADM Command and Output Example:**

```
racadm>>set LifecycleController.lcattributes.autoupdate Disabled
racadm set LifecycleController.lcattributes.autoupdate Disabled
Object value modified successfully

racadm>>get LifecycleController.lcattributes.autoupdate
racadm get LifecycleController.lcattributes.autoupdate
Disabled
```
Conclusion:
For more information on Dell’s enterprise-class servers, see Dell.com/PowerEdge.

For more information about the firmware update sequence, see iDRAC and Lifecycle Controller - A Recommended Workflow for Performing Firmware Updates on PowerEdge Servers white paper.

Reference Profiles

Reference MOFs

Best Practices Guide
http://en.community.dell.com/techcenter/extras/m/white_papers/20066173.aspx
Associated Scripts:
http://en.community.dell.com/techcenter/extras/m/white_papers/20066178.aspx

Web Services Interface Guide for Windows
http://en.community.dell.com/techcenter/extras/m/white_papers/20066174.aspx
Associated scripts
http://en.community.dell.com/techcenter/extras/m/white_papers/20066179.aspx

Web Services Interface Guide for Linux
http://en.community.dell.com/techcenter/extras/m/white_papers/20066176.aspx
Associated scripts
http://en.community.dell.com/techcenter/extras/m/white_papers/20066181.aspx

WS-MAN command line for Windows (Winrm)

WSMAN command line open source for Linux (Openwsman)
http://sourceforge.net/projects/openwsman/

Scripting the Dell Lifecycle Controller

About the Authors
This whitepaper has been co-authored by Texas Roemer, Raja Tamilarasan and Ajay Shenoy; they are software development engineers with the Dell Enterprise Servers Group – Firmware group.