Job Control Profile

Document Number: DCIM1034
Document Type: Specification
Document Status: Published
Document Language: E
Date: 2012-03-08

Version: 1.2.0

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

CONTENTS

Figures

Tables

Job Control Profile

1 Scope

The Dell Job Control Profile extends the management capabilities of referencing profiles by adding the capability to create, schedule, track, and manage jobs that represent platform management operations.

2 Normative References

Refer to the following documents for more information.

NOTE: For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

- DMTF DSP1033, *Profile Registration Profile 1.0.0*
- DMTF DSP0200, *CIM Operations over HTTP 1.2.0*
- DMTF DSP0004, *CIM Infrastructure Specification 2.3.0*
- DMTF DSP1000, *Management Profile Specification Template*
- DMTF DSP1001, *Management Profile Specification Usage Guide*
- DMTF DSP0226, *Web Services for Management (WS-Management) Specification 1.1.0*
- DMTF DSP0227, *WS-Management CIM Binding Specification 1.0.0*
- *Dell Lifecycle Controller Best Practices Guide 1.0, http://en.community.dell.com/techcenter/extras/m/white_papers/20066173.aspx*
- *Dell WSMAN Licenses and Privileges 1.0*
- ISO/IEC Directives, Part 2, *Rules for the structure and drafting of International Standards*, http://isotc.iso.org/livelink/livelink.exe?func=ll&objId=4230456&objAction=browse&sort=subtype
- Unified Modeling Language (UML) from the Open Management Group (OMG), http://www.uml.org
- Dell Tech Center MOF Library: http://www.delltechcenter.com/page/DCIM.Library.MOF
- Related Managed Object Format (MOF) files:
  - DCIM_JobService.mof
  - DCIM_LifeCycleJob.mof
  - DCIM_LCElementConformsToProfile.mof
  - DCIM_LCRegisteredProfile.mof

3 Terms and Definitions

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

3.1

Conditional – Indicates requirements to be followed strictly in order to conform to the document when the specified conditions are met.

3.2

Mandatory – Indicates requirements to be followed strictly in order to conform to the document and from which no deviation is permitted.

3.3

May – Indicates a course of action permissible within the limits of the document.

3.4

Optional – Indicates a course of action permissible within the limits of the document.

3.5

can – Used for statements of possibility and capability, whether material, physical, or causal.

3.6

cannot – Used for statements of possibility and capability, whether material, physical, or causal.

3.7

need not – Indicates a course of action permissible within the limits of the document.

3.8

referencing profile – Indicates a profile that owns the definition of this class and can include a reference to this profile in its “Related Profiles” table.

3.9

shall – Indicates requirements to be followed strictly in order to conform to the document and from which no deviation is permitted.

3.10

shall not – Indicates requirements to be followed strictly in order to conform to the document and from which no deviation is permitted.

3.11

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

3.12

should not – Indicates that a certain possibility or course of action is deprecated but not prohibited

3.13

FQDD – Fully Qualified Device Descriptor is used to identify a particular component in a system.

3.14

Interop Namespace – Interop Namespace is where instrumentation instantiates classes to advertise its capabilities for client discovery.

3.15

Implementation Namespace – Implementation Namespace is where instrumentation instantiates classes relevant to executing core management tasks.

3.16

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

3.17

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

4 Symbols and Abbreviated Terms

4.1

CIM – Common Information Model

4.2

iDRAC – Integrated Dell Remote Access Controller is a management controller for blades and rack and tower servers

4.3

CMC – Chassis Management Controller is a management controller for the modular chassis

4.4

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

5 Synopsis

Profile Name: Job Control

Version: 1.2.0

Organization: Dell

CIM Schema Version: 2.26 Experimental

Central Class: DCIM_JobService

Scoping Class: CIM_ComputerSystem

The Job Control Profile extends the management capability of the referencing profiles by adding the capability to create, schedule, track, and otherwise manage system management tasks, hereafter referred to as “jobs”. In this profile, a job is represented by an instance of a Dell subclass of CIM_ConcreteJob, DCIM_LifecycleJob. DCIM_JobService shall be the Central Class. CIM_ComputerSystem shall be the Scoping Class. Table 1 identifies profiles that are related to this profile.

Table 1 – Related Profiles

Profile Name	Organization	Version	Relationship
Profile Registration	DCIM	1.0	Reference

6 Description

The Job Control Profile describes the job control service, the job types, their behavior, and state transitions and results representation. The profile also describes the relationship of the job service to the profile version information.

Figure 1 represents the class schema for the Job Control Profile. The job service in a managed system is represented by the instance of DCIM_JobService class. The jobs names, status, and error messages are represented by the DCIM_LifeCycleJob class.

The Job Control Profile information is represented with the instance of DCIM_LCRegisteredProfile.

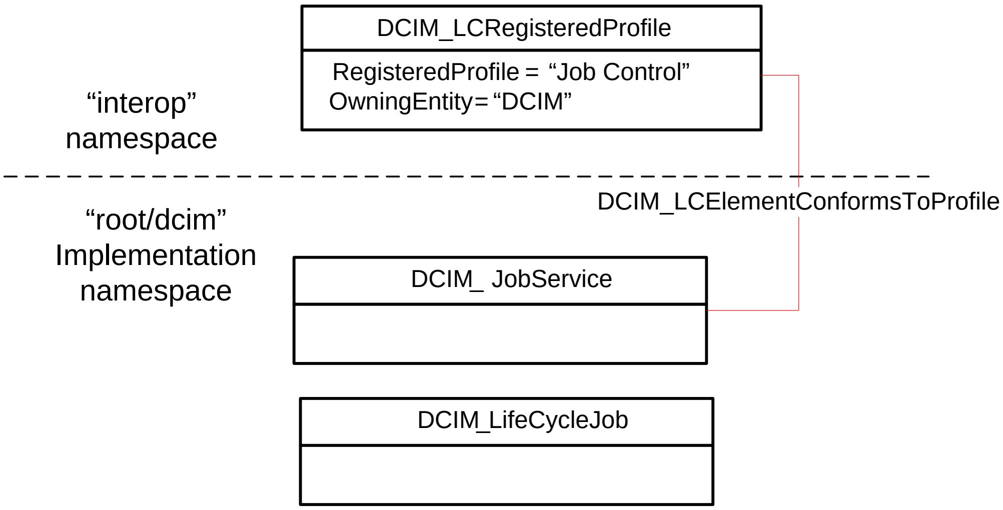
DCIM_LCRegisteredProfile
RegisteredProfile = "Job Control"
OwningEntity= "DCIM"
"interop" namespace
DCIM_LCElementConformsToProfile
"root/dcim" Implementation namespace
DCIM_ JobService
DCIM_LifeCycleJob

Figure 1 – Job Control Profile: Class Diagram

6.1 Job Types

Different kinds of system management tasks, such as firmware updates, reboots, configurations, and so on often take a while to complete. In order to provide a means of performing and tracking tasks and task results, individual tasks are defined as jobs that can be run immediately or at a scheduled time.

Tasks of different types typically have state sequences that are unique to the task performed and the state is represented as the job status. For the purposes of managing these tasks as jobs, several job types have been identified; these include firmware download, firmware update, and reboot. Each job type has a different sequence of states it may pass through. The job may contain one action, or there may be a series of actions taken as part of the job execution.

Refer to Table 7 for a list of job types.

6.2 Creating Jobs

Platform management jobs are created as the result of invoking an extrinsic method for performing a specific task. Jobs that are created as the result of an extrinsic method invocation are initially created without a start time defined and require a management client script or application to subsequently set the scheduled start time by specifying job(s) and start time in a job queue. Note that the reset of the iDRAC itself due to a requested reset or hard reboot does not cause the created jobs to be cancelled. The created jobs will persist and could be executed.

6.3 Grouping Jobs

One or more jobs are be grouped together in job queues to define job order, start time, and to prepare the job grouping for execution in another environment such as Unified Extensible Firmware Interface(UEFI). If the Lifecycle Controller is to perform the system reboot needed to execute update jobs, a reboot job is included in the job queue definition.

6.4 Scheduling Jobs

Jobs, such as firmware updates, that are created as a result of extrinsic method invocations are initially unscheduled. One or more job can be scheduled to start immediately or at a specified start time by setting up a job queue. Job queues are setup by calling the SetupJobQueue() method on the job service.

6.5 Canceling Jobs

Jobs may be canceled by management application. Job can be cancelled either by calling delete instance method on a job instance or through invoking DeleteJobQueue() method but note that using the “JID_CLEARALL” parameter value in the DeleteJobQueue() method will delete all the jobs including the completed or failed ones.

When a pending job is cancelled, all the tasks related to the job will be deleted. If the job is already running, then the instrumentation will delete all the job’s pending tasks and will attempt to undo all the job’s completed tasks. If the job was already completed, cancelling the job will not undo its tasks.

6.6 Auto-Deleting Completed Jobs

Completed jobs will be deleted as a result of the auto-delete policy. Management applications may specify the threshold for activating the auto-delete policy and the time criterion for auto-deleting completed jobs. Once the auto-delete policy is activated, it will delete the jobs that have been completed for more than the specified time criterion. The DCIM_JobService.DeleteOnCompletionTimeout property represents the time criterion for the auto-delete policy with default value of 2880 minutes that can be modified through the SetDeleteOnCompletionTimeout() method. The DCIM_JobService.StartAutoDeleteAtThreshold represents the auto-delete policy activation threshold and has a default value of 50% of the maximum number of jobs possible in the job queue.

6.7 Maintenance Windows

A client script or application that performs platform hardware management, may define specific maintenance time slots. Time slot specification includes being able to specify a scheduled job start time and a duration interval after which, if the job has not been executed, then the job has failed.. The time slot is defined using the StartTimeInterval and Untiltime parameters on the extrinsic method.

6.8 Job Status Lifecycle

LC jobs have a specific workflow associated with their lifecycle. These workflows are provided as a sample to clarify the job lifecycle and the changes to the job status as it progresses from creation to completion. Typically, LC jobs fall within two categories configuration jobs and update jobs.

6.8.1 Configuration Job

The workflow below details the configuration job workflow for setting RAID attributes. NIC and BIOS attribute configuration follows a similar workflow with an additional terminal state, “Completed with Errors”.

Importantly, in all the configuration job workflows, the CreateTargetedConfigJob() method can schedule both the reboot and the created job at the same time. If the CreateTargetConfigJob() method is invoked with parameter values to schedule both the reboot and the created job, then the job state transitions directly from “No Job” to the “JobStatus=Scheduled.”

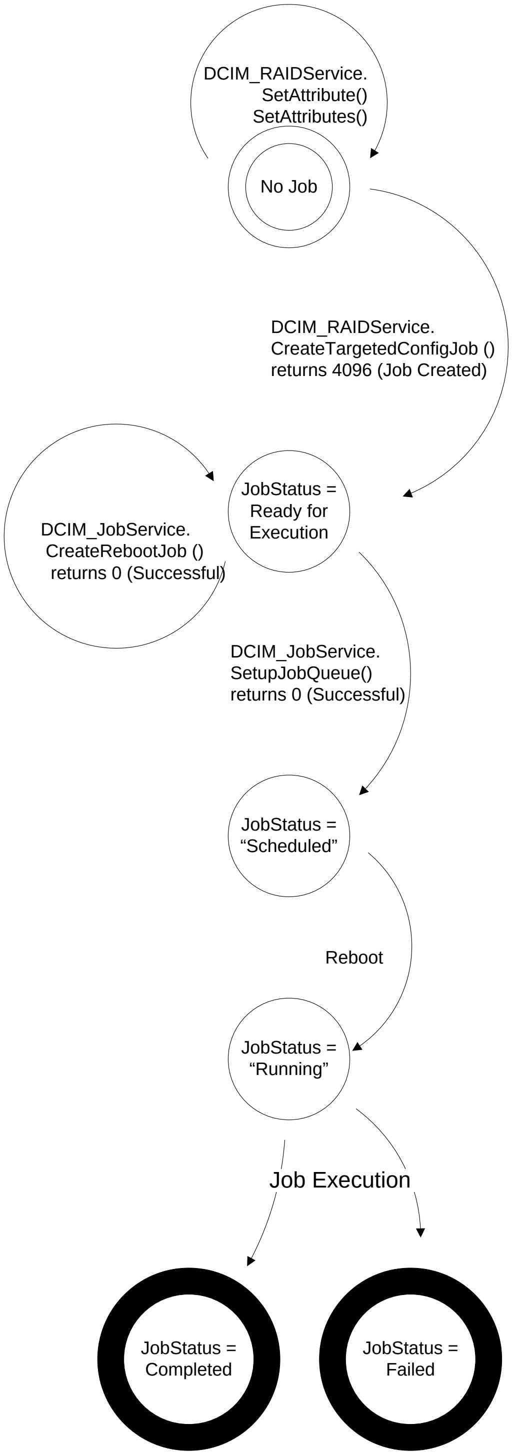

Figure 2 – Job Workflow for RAID Attributes

The workflow below details the configuration job workflow for setting IDRAC card attributes. System attribute configuration follows a similar workflow.

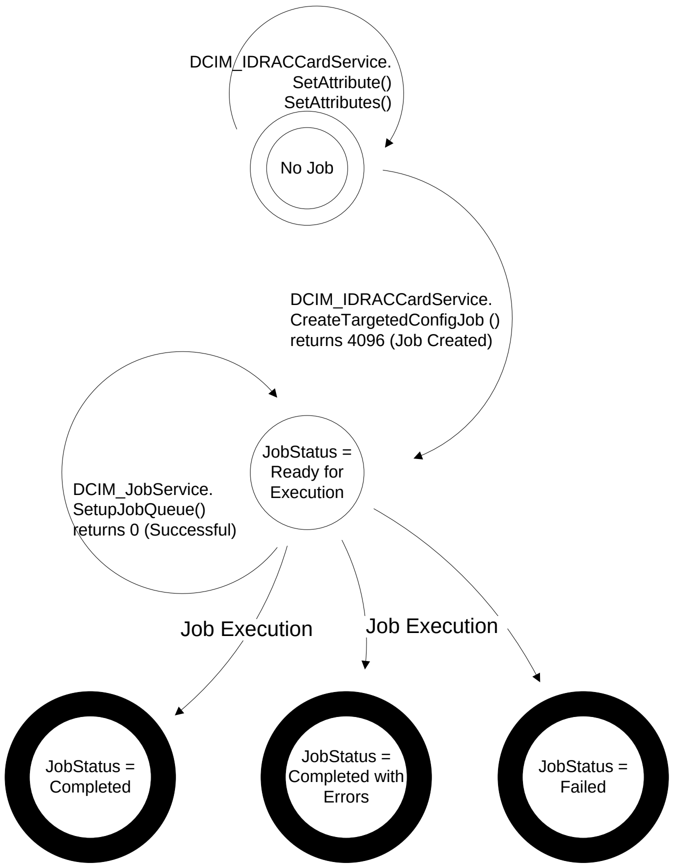

Figure 3 – Job Workflow for IDRAC Card Attributes

Figure 4 shows the timing diagram for configuration job executions. The upper diagram shows the state transitions of a successful configuration job, while the bottom diagram shows the state transitions of the Remote Services (Data Manager) state as affected by the job execution.

For 12G systems, management applications can query for the change (such as enumeration of affected attributes) once the job is completed regardless the Remote Services (Data Manager) state.

NOTE: In 11G, the job completion does not indicate that management applications can query for the completed job's changes but only indicates that all the pending tasks associated with the job have been completed. The Remote Service (Data Manager) needs to reload for the LC interface to reflect new changes. Thus, the Remote Service (Data Manager) status has to transition from "Reloading" to "Ready" to indicate that the LC interface has been updated, and management applications have to check both the job completion as well as the Remote Service (Data Manager) status transition.

6.8.1.1 Lifecycle Log

Configuration job results are also logged in the Lifecycle Log (LCL). Please refer to the Dell Record Log Profile for in detail description on how to retrieve the LCL and its entries through the WSMAN interface.

6.8.1.2 Remote Services Status

The Remote Service (Data Manager) status can be queried using DCIM_LCService.GetRSStatus()and GetRemoteServicesAPIStatus() methods (as described in the Lifecycle Controller (LC) Management Profile). The GetRemoteServicesAPIStatus() method is more robust because it details not only the Remote Service (Data Manager) status (as reflected in the LCStatus output parameter) but also the system's status (as reflected in the ServerStatus output parameter) and overall API status (as reflected in the Status output parameter).

NOTE: In 11G, management applications have to invoke DCIM_LCService.GetRSStatus() or GetRemoteServicesAPIStatus() method (as described in the LC Management Profile) to determine that the completed job changes have been successfully reflected in the LC instrumentation API interface.

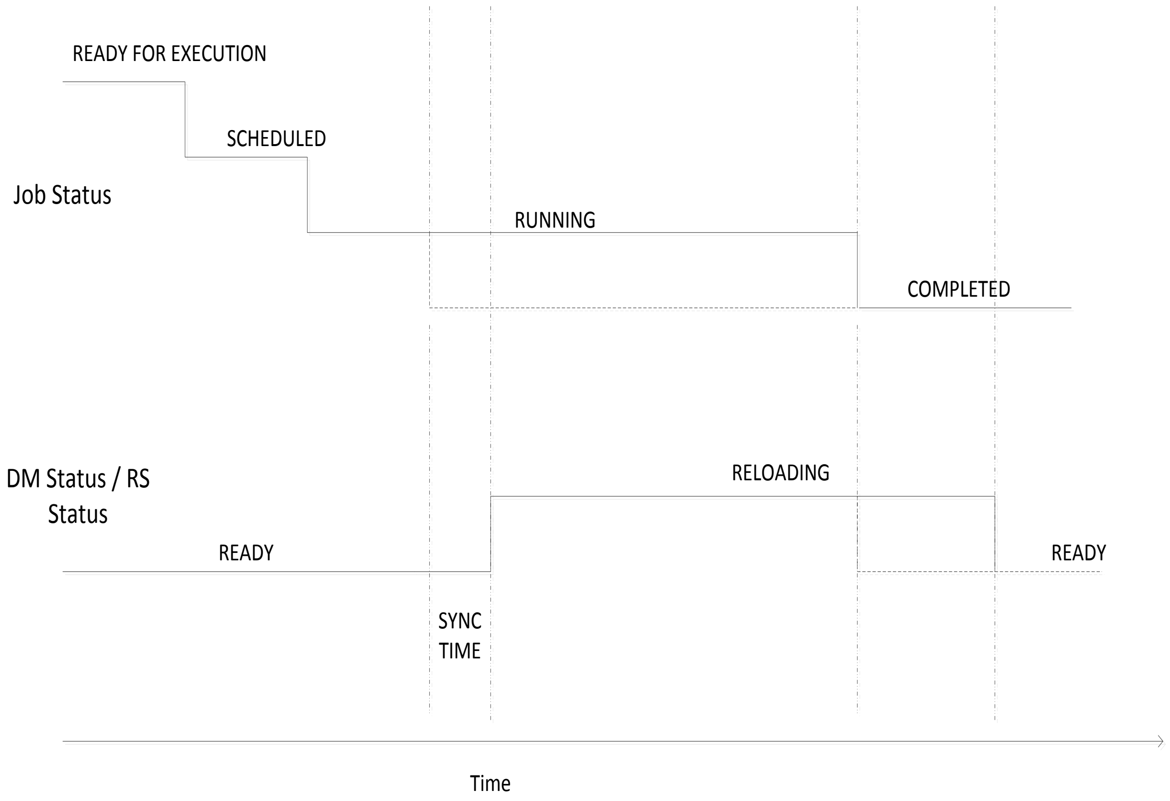

Figure 4 – Configuration Job Timing Diagram

6.8.2 Update Job

The workflow below details the update job lifecycle (DCIM_SoftwareInstallationService.InstallFromURI() invocation). The direct update jobs that do not required reboot transition from "Downloading" state to a terminal state of "Failed" or "Completed". Update jobs that require a reboot will go from state of "Downloading" to a state of "Downloaded" or "Failed". Once in "Downloaded" state, the update job needs to be scheduled using the SetupJobQueue() method in the LC Management Profile (similar to configuration jobs).

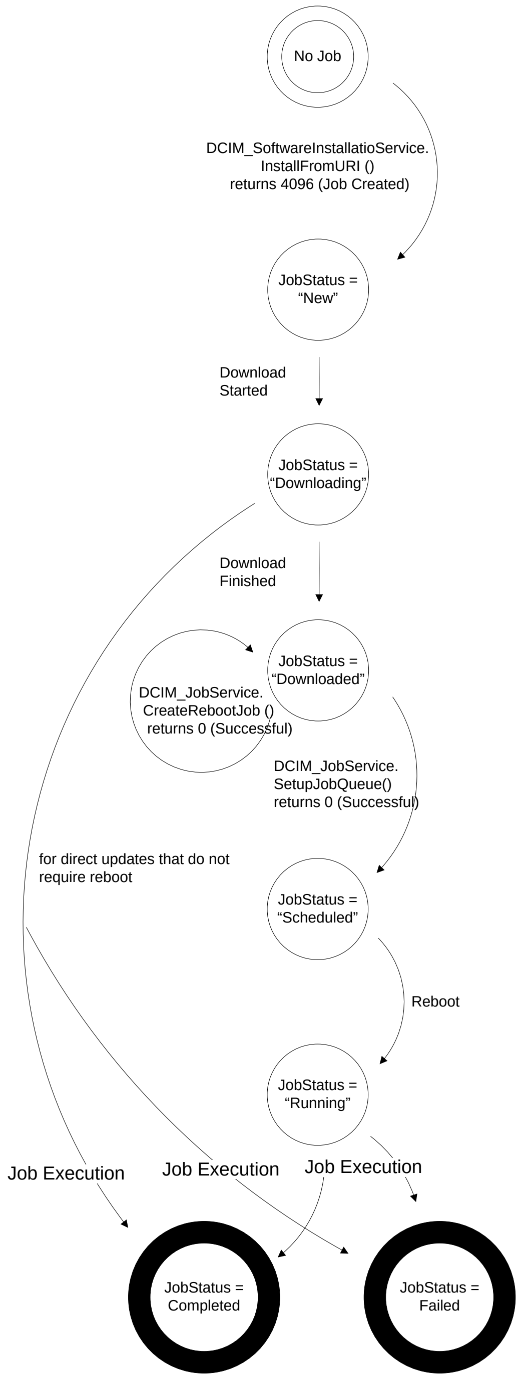

Figure 5 – Update Job Workflow

6.9 Completed Jobs Auto-deletion

Completed jobs will be deleted as a result of the auto-delete policy. Management applications may specify the threshold for activating the auto-delete policy and the time criterion for auto-deleting completed jobs (including failed jobs). Once the auto-delete policy is activated, it will delete the jobs that have been completed for more than the specified time criterion. The DCIM_JobService.DeleteOnCompletionTimeout property represents the time criterion for the auto-delete policy with default value of 2880 minutes that can be modified through the

SetDeleteOnCompletionTimeout() method. The DCIM_JobService.StartAutoDeleteAtThreshold represents the auto-delete policy activation threshold and has a default value of 128 jobs representing the half of the maximum number of jobs possible in the job queue.

7 Implementation Requirements

This section describes the requirements and guidelines for implementing Dell Job Control Profile

Table 2 – CIM Elements: Job Control Profile

Element Name	Requirement	Description
Classes		
DCIM_JobService	Mandatory	The class shall be implemented in the *Implementation Namespace*. See section 7.1
DCIM_LifeCycleJob	Mandatory	The class shall be implemented in the *Implementation Namespace*. See section 7.2
DCIM_LCElementConformsToProfile	Mandatory	The class shall be implemented in the *Implementation Namespace*. See section 7.1 and 7.3
DCIM_LCElementConformsToProfile	Mandatory	The class shall be implemented in the *Interop Namespace*. See section 7.1 and 7.3
DCIM_LCRegisteredProfile	Mandatory	The class shall be implemented in the *Interop Namespace*. See section 7.3
Indications		
None defined in this profile		

7.1 DCIM_JobService

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

7.1.1 Resource URIs for WinRM®

The class Resource URI shall be "http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_JobService?__cimnamespace=root/dcim"

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

The instance Resource URI for DCIM_JobService instance shall be: "http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_JobService?__cimnamespace=root/dcim+SystemCreationClassName=DCIM_ComputerSystem+CreationClassName=DCIM_JobService+ SystemName=Idrac+Name= JobService"

7.1.2 Operations

The following table lists the operations implemented on DCIM_JobService.

Table 3 – DCIM_JobService – Operations

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

7.1.3 Class Properties

Table 4 – Class: DCIM_JobService

Properties	Type	Requirement	Additional Requirments
SystemCreationClassName	String	Mandatory	Key: Value shall be “DCIM_ComputerSystem”
SystemName	String	Mandatory	Key: Value shall be “Idrac”
CreationClassName	String	Mandatory	Key: Value shall be “DCIM_JobService”
Name	String	Mandatory	Key: Value shall be “JobService”
ElementName	String	Mandatory	Value shall be “Job Service”
CurrentNumberOfJobs	uint16	Mandatory	The property shall represent the current number of jobs in the job store.
MaximumNumberOfJobs	uint16	Mandatory	The property shall represent the maximum number of jobs that the job store supports.
DeleteOnCompletionTimeout	uint16	Mandatory	The property shall represent the timeout period in minutes for completed jobs to qualify for auto deletion. If the time taken to complete a job is more than the DeleteOnCompletionTimeout, the job shall be deleted during the auto-delete.
StartAutoDeleteAtThreshold	uint16	Mandatory	The property shall represent the percentage of the MaximumNumberOfJobs which, when reached, shall cause all jobs that have taken more time than the DeleteOnCompletetionTimeout to be deleted.

7.2 DCIM_LifecycleJob

7.2.1 Resource URIs for WinRM®

The class Resource URI shall be “http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_LifecycleJob?__cimnamespace=root/dcim”

The key property shall be the InstanceID.

The instance Resource URI for DCIM_LifecycleJob instance shall be: “http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_LifecycleJob?__cimnamespace=root/dcim+InstanceID=<InstanceID>”

7.2.2 Operations

The following table details the implemented operations on DCIM_LifecycleJob.

Table 5 – DCIM_LifecycleJob - Operations

Operation Name	Requirements	Required Input

Get	Mandatory	Instance URI
Enumerate	Mandatory	Class URI

7.2.3 Class Properties

The following table lists the implemented properties for DCIM_LifecycleJob instance representing a system in a system. The “Requirements” column shall denote whether the property is implemented (for requirement definitions, see section 3). The “Additional Requirements” column shall denote either possible values for the property, or requirements on the value formulation.

Table 6 – Class: DCIM_LifecycleJob

Properties and Methods	Type	Requirement	Additional Requirments
InstanceID	String	Mandatory	Dynamic value returned as an output parameter from the extrinsic operation that created the job.
Name	String	Mandatory	The property shall have value from the “Value” column of Table 7.
JobStatus	String	Mandatory	The property shall have value from the “Status Value” column of Table 8.
JobStartTime	String	Mandatory	The property shall represent the timestamp to start processing the job. JobStartTime shall be in the format: “yyyymmddhhmmss” and the string "TIME_NOW" means immediate
JobUntilTime	String	Mandatory	The property shall represent the time interval after a job has started that it is permitted to run. If the job has not run when the time interval has passed, the job shall be canceled. Canceling a job causes the schedule start time properties to be set to NULL and the JobStatus to be set to Failed. The property value shall be in the following format: “YYYYMMDDHHMMSS”.
PercentComplete	String	Mandatory	The property shall represent the percentage of job completion.
ElapsedTimeSinceCompletion	String	Mandatory	The property shall represent the time interval in minutes since the job has been completed.
Message	String	Mandatory	If an error occurs during the processing of a job, more detailed error information is provided for subsequent inspection of the job completion status. The Message property of the job contains the error message describing the job failure detail.
MessageID	String	Mandatory	An identifier for the error message that can be used to index into Dell Lifecycle Controller Message Registry xml files. The Message Registry files are available in languages such as English, French, Spanish, German, Japanese and Chinese. They also contain more detailed descriptions of the error condition and recommended response actions.
MessageArguments	String	Mandatory	An error message may contain substitution variables that are filled in dynamically at runtime. To support localizing versions of the message, the values of the substitution variables are set in the MessageArguments array in the order they are defined in the Message Registry.

The management tasks requested by the management application shall be represented by instances of a subclass of the DCIM_LifecycleJob class which is a subclass of the CIM_ConcreteJob class. The DCIM_LifecycleJob class is further subclassed as the DCIM_SoftUpdateConcreteJob for BIOS, firmware, and embedded software update jobs. Software update related methods (see the Dell Software Update Profile) return references to the DCIM_SoftUpdateConcreteJob class. Most of the system management tasks require a reboot of the system, which can be scheduled as a job or the reboot can be performed by other means after the job tasks have reached their scheduled start time.

To accomplish these offline platform management actions, several specific job types are defined to represent types of remote enablement actions:

- Image Update
- Image Rollback
- Reboot

7.2.3.1 Name

The DCIM_LifeCycleJob.Name property represents the job type and is formulated as follows:

Table 7 – Job Types

JobType	Description	Value
Update	The flashing of FW into the target device ,	Update:DCIM:InstanceID of SoftwareIdentity
Rollback	The flashing of Available FW into the device.	Rollback:DCIM:InstanceID of SoftwareIdentity
Reboot	Restart of system	• Reboot1 = “PowerCycle” • Reboot2 = “Graceful Reboot without forced shutdown” • Reboot3 = “Graceful Reboot with forced shutdown”
vFlash	Initialize vFlash	VFlashInitialize:Media
vFlash	Create partition	VFlashCreate:Partition<n> Where n is equal to number of vFlash partition indices (1 to 16)
vFlash	Create partition using image	VFlashCreateUsingImage:Partition1<n> Where n is equal to number of vFlash partition indices (1 to 16)
vFlash	Format partition	VFlashFormat:Partition1<n> Where n is equal to number of vFlash partition indices (1 to 16)
vFlash	Attach partition	VFlashAttach:Partition1<n> Where n is equal to number of vFlash partition indices (1 to 16)
vFlash	Detach partition	VFlashDetach:Partition1<n> Where n is equal to number of vFlash partition indices (1 to 16)
vFlash	Export data from partition	VFlashExportData:Partition1<n> Where n is equal to number of vFlash partition indices (1 to 16)
LC Export	LCL log export	LC Export
HW Export	Hardware Inventory export	HW Export
Factory configuration export	Factory configuration export	FACTORY CONFIG Export
RAID configuration	Applying the pending RAID configuration	ConfigRAID:< RAID Controller FQDD> Each RAID controller has an FQDD and is part of the DCIM_ControllerView instance. (DCIM_ControllerView.FQDD) • For example, ConfigRAID :RAID.Integrated.1-1 • For example, DCIM_ControllerView.FQDD = RAID.Integrated.1-1
BIOS configuration	Applying the pending BIOS configuration	ConfigBIOS: BIOS.Setup.1-1
NIC configuration	Applying the pending NIC configuration	ConfigNIC:< NIC FQDD> Each NIC has an FQDD and is part of the DCIM_NICView instance (DCIM_NICView.FQDD). • For example, DCIM_NICView.FQDD = NIC.Embedded.1-1 • For example, ConfigNIC: NIC.Embedded.1-1
iDRAC Card Configuration	Applying the pending iDRAC configuration.	iDRACConfig:<FQDD> • For example, iDRACConfig: iDRAC.Embedded.1

JobType	Description	Value
System Configuration	Applying the pending system configuration.	SYSConfig:<FQDD> • For example, SYSConfig: System.Embedded.1

7.2.3.2 JobStatus

The value of DCIM_LifeCycleJob.JobStatus represents the current state of the specific job. The following table lists the different status values that are valid for different job types, see Table 2 for a list of job types:

Table 8 – JobStatus Property Values

Status Value	Description
Job Type: Update	
New	New Job has been created.
Downloading	Job is Downloading firmware image.
Downloaded	Job Downloaded the firmware image. Note this status is not applicable to the direct update jobs.
Scheduled	Job has been scheduled. Note this status is not applicable to the direct update jobs.
Running	Job is being executed. Note this status is not applicable to the direct update jobs.
Completed	Job has been completed.
Failed	Job failed.
Deleted	Job has been deleted.
Job Type: Rollback	
New	New Job has been created.
Scheduled	Job has been scheduled.
Running	Job is being executed.
Completed	Job has been completed.
Failed	Job has Failed.
Job Type: Reboot	
Pending Reboot	Reboot Pending for this job.
Reboot Completed	Reboot Job completed.
Reboot Failed	Reboot Job failed.
Job Type: vFlash	
New or NEW	New Job has been created.
Completed	Job has been completed.
Failed	Job failed.
Job Type: LC Export, HW Export, FACTORY CONFIG Export	
New	New Job has been created.
Completed	Job has been completed.
Failed	Job failed.
Job Type: RAID Configuration	
New	New Job has been created.
Ready For Execution	Job is ready for execution.
Scheduled	Job has been scheduled.
Running	Job is being executed.

Status Value	Description
Completed	Job has been completed.
Failed	Job failed.
Job Type: NIC Configuration	
New	New Job has been created.
Ready For Execution	Job is ready for execution.
Scheduled	Job has been scheduled.
Running	Job is being executed.
Completed	Job has been completed.
Completed with Errors	Job has been completed with one or more errors.
Failed	Job failed.
Job Type: BIOS Configuration	
New	New Job has been created.
Ready For Execution	Job is ready for execution.
Scheduled	Job has been scheduled.
Running	Job is being executed.
Completed	Job has been completed.
Completed with Errors	Job has been completed with one or more errors.
Failed	Job failed.
Job Type: IDRAC Card Configuration	
New	New Job has been created.
Ready For Execution	Job is ready for execution.
Completed	Job has been completed.
Completed with Errors	Job has been completed with one or more errors.
Failed	Job failed.
Job Type: System Configuration	
New	New Job has been created.
Ready For Execution	Job is ready for execution.
Completed	Job has been completed.
Completed with Errors	Job has been completed with one or more errors.
Failed	Job failed.

7.3 DCIM_LCRegisteredProfile

The CIM_RegisteredProfile class is defined by the Profile Registration Profile. The requirements denoted in this profile are in addition to those mandated by the Profile Registration Profile.

7.3.1 Resource URIs for WinRM®

The class Resource URI shall be "http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_RegisteredProfile?__cimnamespace=root/interop"

The key property shall be the InstanceID property.

The instance Resource URI shall be: “http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_LCRegisteredProfile?__cimnamespace=root/interop+InstanceID= DCIM:JobControl:1.0.0”

7.3.2 Operations

The following table lists the operations implemented on DCIM_SystemView.

Table 9 – DCIM_LCRegisteredProfile - Operations

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

7.3.3 Class Properties

The following table lists the implemented properties for DCIM_LCRegisteredProfile instance representing Job Control Profile implementation. The “Requirements” column shall denote whether the property is implemented (for requirement definitions, see section 3). The “Additional Requirements” column shall denote either possible values for the property, or requirements on the value formulation.

Table 10 – Class: DCIM_LCRegisteredProfile

Properties	Type	Requirement	Additional Requirements
InstanceID	string	Mandatory	This property shall have a value of “DCIM:LCManagement:1.0.0”
RegisteredName	string	Mandatory	This property shall have a value of “Job Control”.
RegisteredVersion	string	Mandatory	This property shall have a value of “1.2.0”.
RegisteredOrganization	uint16	Mandatory	This property shall have a value of 1 (Other).
OtherRegisteredOrganization	string	Mandatory	This property shall match “DCIM”
AdvertiseTypes[]	uint16	Mandatory	This property array shall have values “1(Other)” and “1(Other)”
AdvertiseTypeDescriptions[]	string	Mandatory	This property array shall have values “WS-Identify” and “Interop Namespace”
ProfileRequireLicense[]	string	Mandatory	This property array shall describe the required licenses for this profile. If no license is required for the profile, the property shall have value NULL.
ProfileRequireLicenseStatus[]	string	Mandatory	This property array shall contain the status for the corresponding license in the same element index of the ProfileRequireLicense array property. Each array element shall contain: • “LICENSED” • “NOT_LICENSED” If no license is required for the profile, the property shall have value NULL.

8 Methods

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

8.1 DCIM_JobService.SetupJobQueue()

The SetupJobQueue() method is used for creating a job queue that shall contain one or more DCIM_LyfecycleJobs with a specified order of execution within the queue.

Table 11 – DCIM_JobService.SetupJobQueue() Method: Return Code Values

Value	Description
0	Request was successfully executed.
1	Method is unsupported.
2	Error occurred

Table 12 – DCIM_JobService.SetupJobQueue() Method: Parameters

Qualifiers	Name	Type	Description/Values
IN	JobArray	string[]	Array containing the value of the InstanceID property of the instances of DCIM_LifeCycleJob that represent the set of jobs to add to the job queue. This is an ordered array that represents the sequence in which the jobs are run.
IN	StartTimeInterval	string	Start time for the job execution in format: yyyymmddhhmmss. The string "TIME_NOW" means immediate.
IN	UntilTime	string	End time for the job execution in format: yyyymmddhhmmss. If this parameter is not NULL, then StartTimeInterval parameter shall also be specified..
OUT	Message	string	Error Message
OUT	MessageID	string	Error Message ID- can be used to index into Dell Message registry files.
OUT	MessageArguments	string[]	Substitution variables for dynamic error messages.

Table 13 – DCIM_JobService.SetupJobQueue() Method: Standard Messages

MessageID (OUT parameter)	Message	MessageArguments[]
JCP010	The command was successful	
JCP011	Invalid parameter value	
JCP012	Resource allocation failure	
JCP013	Required parameter not found	
JCP014	Maximum number of jobs per queue exceeded	
JCP015	The job cannot be deleted as it is currently in process	
JCP016	The scheduled time window must be at least 1 hour	
SUP011	Invalid Job ID	
SUP017	Invalid Start Time	
SUP018	Invalid Until Time	
SUP022	JobQueue Exceeds the size limit. Delete unwanted JobID(s)	
SUP023	Duplicate JobID Entries	

8.2 DCIM_JobService.DeleteJobQueue()

The DeleteJobQueue() method is used for deleting one or all jobs from the JobQueue (or job store).

Clearing all the jobs may be accomplished using the keyword JID_CLEARALL for the JobID. Note that the successful execution of this method with the JID_CLEARALL parameter value shall clear all the pending attribute values as well.

When the number of jobs in the JobQueue reaches the maximum limit, jobs in the "Completed" state shall be automatically deleted.

NOTE: In 11G, jobs in the "Failed" state shall not be automatically deleted and shall be manually removed one at a time, or using the keyword JID_CLEARALL as mentioned above.

Table 14 – DCIM_JobService.DeleteJobQueue() Method: Return Code Values

Value	Description
0	Request was successfully executed.
1	Method is unsupported.
2	Error occurred

Table 15 – DCIM_JobService.DeleteJobQueue() Method: Parameters

Qualifiers	Name	Type	Description/Values
IN	JobID	string	The InstanceID property of the instances of DCIM_LifeCycleJob that represent the job to be deleted. The value *"JID_CLEARALL"* for the JobID will clear all the jobs.
OUT	Message	string	Error Message
OUT	MessageID	string	Error Message ID- can be used to index into Dell Message registry files
OUT	MessageArguments	string[]	Substitution variables for dynamic error messages

Table 16 – DCIM_JobService.DeleteJobQueue() Method: Standard Messages

MessageID (OUT parameter)	Message	MessageArguments[]
JCP012	Resource allocation failure	
JCP013	Required parameter not found	
JCP015	The job cannot be deleted as it is currently in process	
SUP011	Invalid Job ID	
SUP020	The specified job was deleted	

8.3 DCIM_JobService.SetDeleteOnCompletionTimeout ()

This method will set the job store auto-delete policy. After successful execution, the jobs that have been completed for longer than the DeleteOnCompletionTimeout parameter specifies shall be deleted when the auto-delete threshold is reached. The auto-delete threshold is specified in the DCIM_JobService.StartAutoDeleteAtThreshold property and represents percentage of maximum jobs.

Table 17 – DCIM_JobService.CreateRebootJob() Method: Return Code Values

Value	Description
0	Request was successfully executed.
1	Method is unsupported.
2	Error occurred

Table 18 – DCIM_JobService.CreateRebootJob() Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	DeleteOnCompletionTimeout	uint16	Specifies wait time for auto job deletion in minutes.
OUT	Message	string	Error Message
OUT	MessageID	string	Error Message ID- can be used to index into Dell Message registry files
OUT	MessageArguments	string[]	Substitution variables for dynamic error messages

Table 19 – DCIM_JobService.CreateRebootJob() Method: Standard Messages

MessageID (OUT parameter)	Message	MessageArguments[]
JCP010	The command was successful	
JCP011	Invalid parameter value	
JCP012	Resoure allocation failure	
JCP013	Required parameter not found	
SUP022	JobQueue Exceeds the size limit. Delete unwanted JobID(s)	

8.4 DCIM_JobService.CreateRebootJob ()

The CreateRebootJob() method is used for creating a reboot job.

Table 20 – DCIM_JobService.CreateRebootJob() Method: Return Code Values

Value	Description
0	Request was successfully executed.
1	Method is unsupported.
2	Error occurred

Table 21 – DCIM_JobService.CreateRebootJob() Method: Parameters

Qualifiers	Name	Type	Description/Values
IN	RebootJobType	uint16	Input Parameter represents the type of Reboot : • 1 = “PowerCycle" • 2 = “Graceful Reboot without forced shutdown” • 3 = “Graceful Reboot with forced shutdown"
OUT	Message	string	Error Message
OUT	MessageID	string	Error Message ID- can be used to index into Dell Message registry files
OUT	MessageArguments	string[]	Substitution variables for dynamic error messages
OUT	Job	DCIM_LifecycleJob REF	Returns the created reboot job.

Table 22 – DCIM_JobService.CreateRebootJob() Method: Standard Messages

MessageID (OUT parameter)	Message	MessageArguments[]
JCP010	The command was successful	
JCP011	Invalid parameter value	
JCP012	Resoure allocation failure	
JCP013	Required parameter not found	
SUP022	JobQueue Exceeds the size limit. Delete unwanted JobID(s)	

9 Use Cases

See *Lifecycle Controller (LC) Integration Best Practices Guide*

10 CIM Elements

No additional details specified.

11 Privilege and License Requirement

The following table describes the privilege and license requirements for the listed operations . For the detailed explanation of the privileges and licenses, refer to the Dell WSMAN Licenses and Privileges specification.

Table 23 – Privilege and License Requirements

Class and Method	Operation	User Privilege Required	License Required
DCIM_LifecycleJob	ENUMERATE, GET	Login	None.
DCIM_JobService	ENUMERATE, GET	Login	None.
DCIM_JobService.SetupJobQueue()	INVOKE	Login, Configure	LM_REMOTE_CONFIGURATION
DCIM_JobService.DeleteJobQueue()	INVOKE	Login, Configure	LM_REMOTE_CONFIGURATION
DCIM_JobService.CreateRebootJob()	INVOKE	Login	LM_REMOTE_CONFIGURATION
DCIM_JobService.DeleteJobs()	INVOKE	Login	LM_REMOTE_CONFIGURATION
DCIM_JobService. SetDeleteOnCompletionTimeout()	INVOKE	Login	LM_REMOTE_CONFIGURATION
DCIM_LCRegisteredProfile	ENUMERATE, GET	Login	None.
DCIM_LCElementConformsToProfile	ENUMERATE, GET	Login	None.