

License Management Profile

Document Number:	DCIM1033
Document Type:	Specification
Document Status:	Draft
Document Language:	E
Date:	2017-06-20
Version:	4.0.0

This profile is for informational purposes only and may contain typographical errors and technical inaccuracies. The content is provided as-is, without express or implied warranties of any kind. If there is no 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.

Copyright © 2017 Dell Inc. or its subsidiaries. All rights reserved. Dell, EMC, and other trademarks are trademarks of Dell Inc. or its subsidiaries. Other trademarks may be trademarks of their respective owners.

Contents

1.	Scope	5
2.	Normative References	5
3.	Terms and Definitions	5
3.1.	Can	5
3.2.	Cannot	5
3.3.	Conditional	6
3.4.	Mandatory	6
3.5.	May	6
3.6.	Need not	6
3.7.	Optional	6
3.8.	Referencing profile	6
3.9.	Shall	6
3.10.	Shall not	6
3.11.	Should	6
3.12.	Should not	6
3.13.	Dell Product Licensing Manager	6
3.14.	NULL	6
4.	Symbols and Abbreviated Terms	6
4.1.	CIM	6
4.2.	IDRAC	7
4.3.	FQDD	7
4.4.	License Manager	7
5.	Synopsis	7
6.	Description	7
7.	Implementation Requirements	8
7.1.	DCIM_LicenseManagementService	8
7.1.1.	Resource URIs for WinRM®	9
7.1.2.	Operations	9
7.1.3.	Properties	9
7.2.	DCIM_LicensableDevice	9
7.2.1.	Resource URIs for WinRM®	9
7.2.2.	Operations	10
7.2.3.	Properties	10
7.3.	DCIM_License	11

7.3.1.	Resource URIs for WinRM®	11
7.3.2.	Operations	11
7.3.3.	Properties	12
7.4.	License Management Profile Profile Registration	13
7.4.1.	Resource URIs for WinRM®	13
7.4.2.	Operations	13
7.4.3.	Properties	14
8.	Methods	14
8.1.	DCIM_LicenseManagementService.ImportLicense()	14
8.2.	DCIM_LicenseManagementService.ImportLicense FromNetworkShare()	16
8.3.	DCIM_LicenseManagementService.DeleteLicense()	19
8.4.	DCIM_LicenseManagementService.ExportLicense()	20
8.5.	DCIM_LicenseManagementService.ExportLicenseByDevice()	22
8.6.	DCIM_LicenseManagementService.ExportLicense ToNetworkShare()	23
8.7.	DCIM_LicenseManagementService.ExportLicenseByDevice ToNetworkShare()	25
8.8.	DCIM_LicenseManagementService.ReplaceLicense()	26
9.	Use Cases	28
10.	CIM Elements	28
11.	Privilege and License Requirement	28
12.	Change log	29

1. Scope

The License Management Profile is an autonomous profile for modeling the Dell Product License Manager (License Manager). It extends the management capabilities of referencing profiles by adding the capability to represent the devices and Licenses managed by the Dell Product Licensing Manager. The License Manager is modeled as a collection of devices, licenses and views each class representing devices or feature licenses and a service class serving as a host for the operations (extrinsic methods) associated with the profile. This profile consists of extensions to both the CIM_BIOSService and the CIM_View abstract class. There is typically only one Dell Product License Manager per system platform.

2. Normative References

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

- DMTF DSP1033, *Profile Registration Profile 1.0.0*
- DELL, *Job Control 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*
- *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>
 - o DCIM_LicensableDevice.mof
 - o DCIM_License.mof
 - o DCIM_LicenseManagementService.mof
 - o DCIM_LCElementConformsToProfile.mof
 - o DCIM_LCRegisteredProfile.mof

3. Terms and Definitions

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

3.1. Can

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

3.2. Cannot

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

3.3. Conditional

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

3.4. Mandatory

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

3.5. May

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

3.6. Need not

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

3.7. Optional

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

3.8. Referencing profile

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

3.9. Shall

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

3.10. Shall not

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

3.11. Should

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

3.12. Should not

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

3.13. Dell Product Licensing Manager

Dell feature licensing server

3.14. NULL

a value that indicates that a property or parameter is not defined

4. Symbols and Abbreviated Terms

4.1. CIM

Common Information Model

4.2. IDRAC

Integrated Dell Remote Access Controller

4.3. FQDD

Fully Qualified Device Descriptor – descriptor that uniquely identifies a device within the domain of the management endpoint such as service process.

4.4. License Manager

Dell Product License Manager

5. Synopsis

Profile Name: License Management Profile

Version: 4.0.0

Organization: Dell

CIM Schema Version: 2.41 Final

Central Class: DCIM_LicenseManagementService

Scoping Class: CIM_ComputerSystem

The License Management Profile extends the management capability of the referencing profiles by adding the capability to describe licensable devices and licenses in a very simple fashion. In this profile, license is represented by DCIM_License class, and licensable device represented by DCIM_LicensableDevice class.. An instance of DCIM_LicenseManagementService shall be the Central Instance. The instance of CIM_ComputerSystem with which the Central Instance is associated through the CIM_HostedService association shall be the Scoping Instance.

Table 1 identifies profiles that are related to this profile.

Table 1. Related Profiles

Profile Name	Organization	Version	Relationship
Profile Registration	DCIM	1.0	Reference
Job Control Profile	DCIM	1.0	Reference

6. Description

The License Management Profile describes the management and configuration of the License Manager attribute configuration service and the operations that the service manages. The profile also describes the relationship of the License Manager service to the DMTF/Dell profile version information.

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

The License Manager service in a managed system is represented by the instance of DCIM_LicenseManagementService class. The License Management Profile information is represented with the instance of CIM_RegisteredProfile.

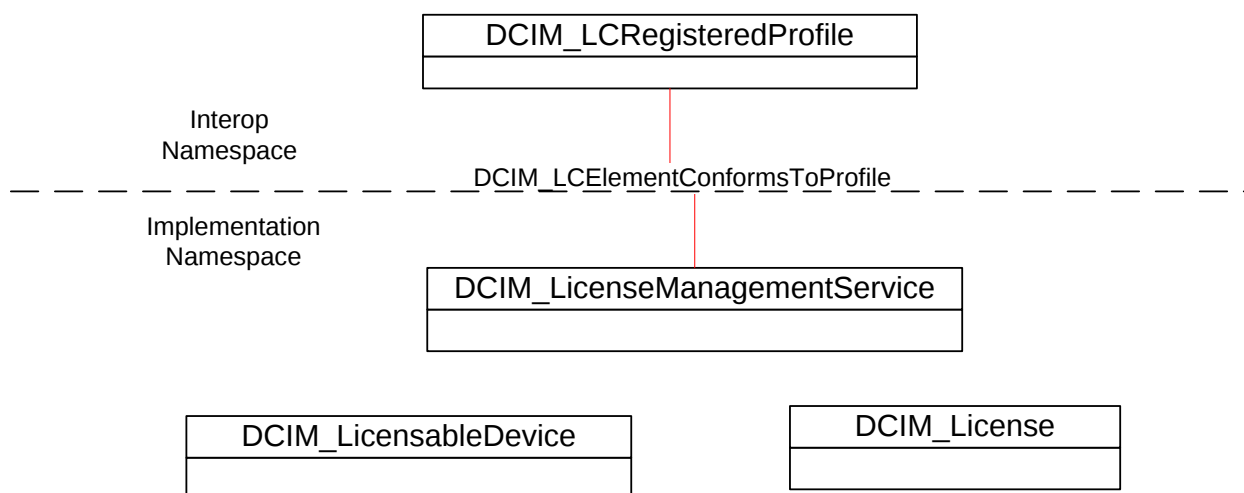


Figure 1. Dell License Management Profile: Class Diagram

7. Implementation Requirements

Requirements and guidelines for propagating and formulating various properties of the classes in the Simple NIC Profile are discussed in this section. Methods are listed in section 8.

Table 2 shows the instances of CIM Elements for this profile. Instances of the CIM Elements shall be implemented as described in section 7.

Table 2. CIM Elements: License Management Profile

Class Name	Requirement	Description
Classes		
DCIM_LicenseManagementService	Mandatory	The class shall be implemented in the <i>Implementation Namespaces</i> . See section 7.1
DCIM_LicensableDevice	Mandatory	The class shall be implemented in the <i>Implementation Namespaces</i> . See section 7.2
DCIM_License	Mandatory	The class shall be implemented in the <i>Implementation Namespaces</i> . See section 7.3
DCIM_LCElementConformsToProfile	Mandatory	The class shall be implemented in both the <i>Interop</i> and <i>Implementation Namespaces</i> . See section 7.1 and 7.4.
DCIM_LCRegisteredProfile	Mandatory	The class shall be implemented in the <i>Interop Namespace</i> . See section 7.4
Indications		
None defined in this profile		

7.1. DCIM_LicenseManagementService

DCIM_LicenseManagementService is derived from CIM_BIOSService. It is used to provide a central class for the Dell License Management Profile and to provide an interface for the extrinsic methods used for license management.

The DCIM_LCElementConformsToProfile association(s) shall reference the DCIM_LicenseManagementService instance.

7.1.1. Resource URIs for WinRM®

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

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

The instance Resource URI for DCIM_LicenseManagementService instance shall be:

“http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_LicenseManagementService?
?__cimnamespace=root/dcim+SystemCreationClassName=DCIM_SPCoMputerSystem+SystemName=systemmc
+CreationClassName=DCIM_LicenseManagementService+Name=DCIM:LicenseManagementService:1”

7.1.2. Operations

The following table details the implemented operations on DCIM_LicenseManagementService.

Table 3. DCIM_LicenseManagementService Operations

Operation	Requirement	Required Input
Get	Mandatory	Instance URI
Enumerate	Mandatory	Class URI
Invoke	Mandatory	Instance URI

7.1.3. Properties

The following table details the implemented properties for an instance of DCIM_LicenseManagementService.

The “Requirements” column shall denote the implementation requirement for the corresponding property. If the column “Name” matches the property name, the property either shall have the value denoted in the corresponding column “Additional Requirements”, or shall be implemented according to the requirements in the corresponding column “Additional Requirements”.

Table 4. DCIM_LicenseManagementService Properties

Properties	Type	Requirement	Additional Requirements
SystemCreationClassName	string	Mandatory	This property value shall be “DCIM_SPCoMputerSystem”
CreationClassName	string	Mandatory	This property value shall be “DCIM_LicenseManagementService”
SystemName	string	Mandatory	This property value shall be “systemmc”
Name	string	Mandatory	This property value shall be “DCIM:LicenseManagementService:1”
ElementName	string	Mandatory	This property value shall be “License Management Service”

7.2. DCIM_LicensableDevice

This section describes the implementation for the DCIM_LicensableDevice class. DCIM_LicensableDevice is derived from CIM_ManagedSystemElement and it models a licensable device in the system that has registered with the license manager on power up of the system.

7.2.1. Resource URIs for WinRM®

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

The key property shall be the InstanceID.

The instance Resource URI for DCIM_LicensableDevice instance shall be:

"http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_LicensableDevice?
?__cimnamespace=root/dcim+InstanceID=<InstanceID>"

7.2.2. Operations

The following table details the implemented operations on DCIM_LicensableDevice.

Table 5. DCIM_LicensableDevice Operations

Operation	Requirement	Required Input
Get	Mandatory	Instance URI
Enumerate	Mandatory	Class URI

7.2.3. Properties

The following table details the implemented properties for an instance of DCIM_LicensableDevice.

The "Requirements" column shall denote the implementation requirement for the corresponding property. If the column "Name" matches the property name, the property either shall have the value denoted in the corresponding column "Additional Requirements", or shall be implemented according to the requirements in the corresponding column "Additional Requirements".

Table 6. DCIM_LicensableDevice Properties

Properties	Type	Requirement	Additional Requirements
InstanceID	string	Mandatory	The property shall have the value of FQDD for the device
DeviceState	uint32	Mandatory	The property value shall represent the current device state and shall have one of the following values: • 1 - DeviceNotReady • 2 - DevicePresent
DeviceType	uint32	Mandatory	The property value shall represent the device type and shall have one of the following values: • 0 - Internal • 1 - External • 2 - Virtual
UniqueID	string	Mandatory	The property value shall represent the device's unique ID as used in the binding information. The information may not be for all the licensable devices. In such cases, the property value shall be NULL.
FQDD	string	Mandatory	The property value shall represent the device's unique FQDD.
DeviceID	string	Mandatory	The property value shall represent the PCI device ID.
VendorID	string	Mandatory	The property value shall represent the PCI vendor/manufacturer ID.
SubsystemID	string	Mandatory	The property value shall represent the PCI subsystem ID.
SubsystemVendorID	string	Mandatory	The property value shall represent the PCI subsystem vendor ID.

Properties	Type	Requirement	Additional Requirements
Model	string	Mandatory	The property value shall represent the model description of the device.
LicenseList[]	string	Mandatory	The property array values shall contain the Entitlement IDs of the licenses that are assigned to this device.
DeviceStatusMessage	string	Mandatory	The property shall have the message string associated with the license status of the device. If the property has value NULL, then there is no message associated with the license status of the device.
DeviceStatusMessageID	string	Mandatory	The property shall represent the message ID associated with the license status of the device. If the property has value NULL, then there is no message associated with the license status of the device.
DevicePrimaryStatus	uint16	Mandatory	The property shall represent the license status for the device: • 0 – Unknown • 1 – OK • 2 – Warning • 3 – Error
RollupStatus	uint16	Mandatory	The property shall represent the accumulative license status for all the sub-devices (the worst case): • 0 – Unknown • 1 – OK • 2 – Warning • 3 – Error

7.3. DCIM_License

This section describes the implementation for the DCIM_License class. DCIM_License is derived from CIM_ManagedSystemElement. It models a feature license installed on a system.

This class shall be instantiated in the Implementation Namespace.

Zero or more instances of DCIM_License shall be instantiated.

7.3.1. Resource URIs for WinRM®

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

The key property shall be the InstanceID.

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

7.3.2. Operations

The following table details the implemented operations on DCIM_License.

Table 7. DCIM_License Operations

Operation	Requirement	Required Input
Get	Mandatory	Instance URI
Enumerate	Mandatory	Class URI

7.3.3. Properties

The following table details the implemented properties for an instance of DCIM_License.

The “Requirements” column shall denote the implementation requirement for the corresponding property. If the column “Name” matches the property name, the property either shall have the value denoted in the corresponding column “Additional Requirements”, or shall be implemented according to the requirements in the corresponding column “Additional Requirements”.

Table 8. DCIM_License Properties

Properties	Type	Requirement	Additional Requirements
InstanceID	string	Mandatory	The property shall have the value of the value the entitlement ID of the License
EntitlementID	string	Mandatory	The property shall have the value of the entitlement ID of the license which is a license unique ID
LicenseDescription	string	Mandatory	The property shall represent the license’s solution set in multiple languages.
LicenseType	uint32	Mandatory	The property shall represent the license type and shall have one of the following values: • 1 - Perpetual • 2 - Leased • 3 - Evaluation • 4 - Site
LicenseAttributes	uint32	Mandatory	The property shall represent bound state to a device and shall have the following value: • 0 - Unbound • 1 - Bound – the license is bound to a specific device and shall only license that specific device. The specific device information is described through DeviceFamilyDeviceID, DeviceFamilyVendorID, DeviceFamilySubsystemID, and DeviceFamilySubsystemVendorID properties. • 4- License if specific to 14G
DeviceFamilyDeviceID[]	string	Mandatory	The property shall represent the PCI DeviceIDs of devices supported by this license
DeviceFamilyVendorID[]	string	Mandatory	The property shall represent the PCI VendorIDs of devices supported by this license.
DeviceFamilySubsystemID[]	string	Mandatory	The property shall represent PCI SubsystemIDs of devices supported by this license
DeviceFamilySubsystemVendorID[]	string	Mandatory	The property shall represent SubsystemVendorIDs of devices supported by this license.
LicenseInstallDate	datetime	Mandatory	The property shall represent the date on which the license was installed.

Properties	Type	Requirement	Additional Requirements
LicenseSoldDate	datetime	Mandatory	The property shall represent the date the factory created the license.
LicenseStartDate	datetime	Optional	The property shall represent the license start date.
LicenseEndDate	datetime	Conditional	The property shall represent the end date of the license. The property shall be implemented, if the LicenseType property has value 2 (Leased) and 4 (Site).
LicenseDuration	uint16	Mandatory	The property shall represent the license period in days and is counted only during the uptime of the system.
EvalLicenseTimeRemaining	uint16	Conditional	The property shall represent the license period remaining in days and is counted only during the uptime of the system. The property shall be implemented, if the LicenseType property has value 3(Evaluation).
AssignedDevices[]	string	Mandatory	The property array shall have values set to FQDDs of devices that have been assigned to this license.
LicenseStatusMessage	string	Mandatory	The property shall have the message string associated with the license status. If the property has value NULL, then there is no message associated with the license status.
LicenseStatusMessageID	string	Mandatory	The property shall represent the message ID associated with the license status. If the property has value NULL, then there is no message associated with the license status.
LicensePrimaryStatus	uint16	Mandatory	The property shall represent the license status: • 0 – Unknown • 1 – OK • 2 – Warning • 3 – Error

7.4. License Management Profile Profile Registration

This section describes the implementation for the DCIM_LCRegisteredProfile class.

This class shall be instantiated in the Interop Namespace.

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

7.4.1. 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:LicenseManagement:1.0.0"

7.4.2. Operations

The following table details the implemented operations on DCIM_LCRegisteredProfile.

Table 9. DCIM_LCRegisteredProfile - Operations

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

7.4.3. Properties

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

Table 10. DCIM_LCRegisteredProfile Properties

Properties	Type	Requirement	Description
InstanceID	string	Mandatory	This property shall have a value of “DCIM:LicenseManagement:4.0.0”.
RegisteredName	string	Mandatory	This property shall have a value of “License Management”.
RegisteredVersion	string	Mandatory	This property shall have a value of “1.0.0”.
RegisteredOrganization	uint16	Mandatory	This property shall have a value of 1 (Other).
OtherRegisteredOrganization	string	Mandatory	This property shall match “DCIM”
AdvertisedTypes[]	Uint16	Mandatory	This property array shall contain [1(Other), 1 (Other)].
AdvertiseTypeDescriptions[]	String	Mandatory	This property array shall contain ["WS-Identify", "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_LicenseManagementService.ImportLicense()

A method used to import License Files to the License Manager. The ImportLicense() method imports the License given in the input parameter to the License Manager Data Store.

Table 11. DCIM_LicenseManagementService.ImportLicense() Method: Return Code Values

Value	Description
2	Error occurred
4096	Job started: REF returned to started CIM_ConcreteJob

Table 12. DCIM_LicenseManagementService.ImportLicense() Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	LicenseFile	String	A base 64 encoded string of the XML License file
IN, REQ	FQDD	String	FQDD of the device to apply the License to
IN, REQ	ImportOptions	Integer	Flag to force or install for all like devices. no options=0, force =1 and all = 2
OUT	MessageID[]	String	Error MessageID is returned if the method fails to execute
OUT	Message[]	String	Error Message in English corresponding to MessageID is returned if the method fails to execute
OUT	MessageArguments[]	String	Substitution variables for dynamic error messages

Table 13. DCIM_LicenseManagementService.ImportLicense() Method: Standard Messages

MessageID (OUT parameter)	Message	MessageArguments[]
LIC001	INFO: License Manager command not supported	
LIC002	INFO: Invalid command	
LIC003	INFO: One or more parameters in request are invalid	
LIC004	WARN: Unable to allocate needed resource(s)	
LIC005	INFO: Maximum number licenses installed	
LIC006	CRITICAL: License Manager unable to allocate needed resources at startup	
LIC007	INFO: Invalid LDSR OID	
LIC008	INFO: LDSR OID not found	
LIC009	INFO: Specified device not found	
LIC010	INFO: The passed Transaction ID is not found	
LIC011	INFO: License file does not exist	
LIC016	CRITICAL: License Manager has detected internal memory data corruption	
LIC017	INFO: License file signature invalid	
LIC018	INFO: License already exists in the system.	
LIC020	INFO: EULA import upgrade warning	
LIC021	INFO: Device binding did not match license binding data	
LIC022	INFO: System service tag did not match license service tag or system service tag not set	
LIC023	INFO: Upgrade license not an upgrade for existing license	

MessageID (OUT parameter)	Message	MessageArguments[]
LIC024	INFO: License not for specified device	
LIC025	INFO: The system model does not support the license	
LIC026	INFO: The system generation does not support license	
LIC027	WARN: No device found suitable for template	
LIC028	INFO: License already imported to device	
LIC029	INFO: May not replace a non-evaluation license with an evaluation license	
LIC030	INFO: License has expired	
LIC031	INFO: One or more license dates are not in ISO 8061 format	
LIC032	INFO: License file missing data	
LIC033	INFO: License file contains unknown data	
LIC034	INFO: License features' dependency are unmet	
LIC035	INFO: License not imported because license start date in the future	
LIC039	WARN: File I/O error occurred when saving Licensable Device Service Record	
LIC041	INFO: File I/O error occurred when saving license XML file	
LIC045	INFO: Restart iDRAC to complete import of license	
LIC047	INFO: I/O error occurred writing evaluation history	
LIC048	INFO: Evaluation license features already licensed	
LIC900	The command was successful	
LIC901	Invalid parameter value for %s	LicenseFile, FQDD, ImportOptions
LIC902	Resource allocation failure	
LIC903	Missing parameter(s) %ss	LicenseFile, FQDD, ImportOptions

8.2. DCIM_LicenseManagementService.ImportLicenseFromNetworkShare()

A method used to import License Files to the License Manager. The ImportLicenseFromNetworkShare() method imports the License given in the location of the share.

The following tables specify ImportLicense return values and parameters.

Table 14. DCIM_LicenseManagementService.ImportLicenseFromNetworkShare() Method: Return Code Values

Value	Description
0	Request was successfully executed.
2	Error occurred

Table 15. DCIM_LicenseManagementService.ImportLicenseFromNetworkShare() Method: Parameters

Qualifiers	Name	Type	Description/Values
IN	Username	String	Username for Network share authentication
IN	Password	String	Password for Network share authentication
IN	Workgroup	String	Name of the workgroup for CIFS share authentication
IN, REQ	ShareType	Uint16	Type of network share. (0 = NFS, 2 = CIFS, 5 = HTTP, 6 = HTTPS)
IN	ShareName	string	Name of the CIFS share or full path to the NFS share. Optional for HTTP/HTTPS share, this may be treated as the path of the directory containing the file.
IN, REQ	IPAddress	string	IP address of the machine hosting the Network share.
IN	LicenseName	String	The file name of the license file to be imported.
IN, REQ	FQDD	String	FQDD of the device to apply the License to
IN, REQ	ImportOptions	Uint16	Flag to force or install for all like devices. no options=0, force =1 and all = 2
IN	ProxySupport	Uint16	It specifies if proxy is to be used or not. Default is 1 (Off). • 1 → Off • 2 → On - use default proxy settings • 3 → On - use passed in parameters for proxy
IN	ProxyType	Uint16	It specifies the proxy type of the proxy server. Default is 0 (HTTP). • 0 → HTTP • 1 → SOCKS
IN	ProxyServer	String	This string represents the IP Address of the proxy server
IN	ProxyUname	String	This string represents the username for proxy server
IN	ProxyPasswd	String	This string represents the password to login into proxy server
IN	ProxyPort	String	This string represents the proxy port for the proxy server
IN	IgnoreCertWarning	Uint16	Specifies if certificate warning is to be ignored when HTTPS is specified. If IgnoreCertWarning is on: certificate will be ignored. Default is 1 (Off). • 1 → Off • 2 → On
OUT	MessageID[]	String	Error MessageID is returned if the method fails to execute
OUT	Message[]	String	Error Message in English corresponding to MessageID is returned if the method fails to execute
OUT	MessageArguments[]	String	Substitution variables for dynamic error messages

Table 16. DCIM_LicenseManagementService.ImportLicenseFromNetworkShare() Method: Standard Messages

MessageID (OUT parameter)	Message	MessageArguments[]
LIC001	INFO: License Manager command not supported	
LIC002	INFO: Invalid command	

MessageID (OUT parameter)	Message	MessageArguments[]
LIC003	INFO: One or more parameters in request are invalid	
LIC004	WARN: Unable to allocate needed resource(s)	
LIC005	INFO: Maximum number licenses installed	
LIC006	CRITICAL: License Manager unable to allocate needed resources at startup	
LIC007	INFO: Invalid LDSR OID	
LIC008	INFO: LDSR OID not found	
LIC009	INFO: Specified device not found	
LIC010	INFO: The passed Transaction ID is not found	
LIC011	INFO: License file does not exist	
LIC016	CRITICAL: License Manager has detected internal memory data corruption	
LIC017	INFO: License file signature invalid	
LIC018	INFO: License already exists in the system.	
LIC020	INFO: EULA import upgrade warning	
LIC021	INFO: Device binding did not match license binding data	
LIC022	INFO: System service tag did not match license service tag or system service tag not set	
LIC023	INFO: Upgrade license not an upgrade for existing license	
LIC024	INFO: License not for specified device	
LIC025	INFO: The system model does not support the license	
LIC026	INFO: The system generation does not support license	
LIC027	WARN: No device found suitable for template	
LIC028	INFO: License already imported to device	
LIC029	INFO: May not replace a non-evaluation license with an evaluation license	
LIC030	INFO: License has expired	
LIC031	INFO: One or more license dates are not in ISO 8061 format	
LIC032	INFO: License file missing data	
LIC033	INFO: License file contains unknown data	
LIC034	INFO: License features' dependency are unmet	
LIC035	INFO: License not imported because license start date in the future	
LIC039	WARN: File I/O error occurred when saving Licensable Device Service Record	
LIC041	INFO: File I/O error occurred when saving license XML file	
LIC045	INFO: Restart iDRAC to complete import of license	

MessageID (OUT parameter)	Message	MessageArguments[]
LIC047	INFO: I/O error occurred writing evaluation history	
LIC048	INFO: Evaluation license features already licensed	
LIC900	The command was successful	
LIC901	Invalid parameter value for %s	UserName,Password,Workgroup,ShareType,ShareName,IPAddress,ImportOptions,FQDD,LicenseName
LIC902	Resource allocation failure	
LIC903	Missing parameter(s) %s	UserName, Password, ShareType, ShareName, IPAddress, ImportOptions, FQDD
LIC904	Could not connect to remote share	
LIC905	LicenseName cannot exceed 64 characters	
LIC906	License file does not exist on remote share	
JCP017	Maximum job limit reached, cannot create new jobs	

8.3. DCIM_LicenseManagementService.DeleteLicense()

A method used to delete assigned licenses. The DeleteLicense() method deletes a specific license from all devices it is assigned to if only the EntitlementID parameter is present, all licenses from a specific device if only the FQDD parameter is present, or a specific license from a specific device if both the EntitlementID and FQDD parameters are present. Either the EntitlementID or FQDD parameter shall be present.

The following tables specify DeleteLicense return values and parameters.

Table 17. DCIM_LicenseManagementService.DeleteLicense() Method: Return Code Values

Value	Description
2	Error occurred
4096	Job started: REF returned to started CIM_ConcreteJob

Table 18. DCIM_LicenseManagementService.DeleteLicense() Method: Parameters

Qualifiers	Name	Type	Description/Values
IN	FQDD	String	Unique device ID
IN	EntitlementID	String	Unique license ID
IN, REQ	DeleteOptions	UInt16	Flag used to force delete or delete license from all like devices. No options =0, force =1 and all = 2
OUT	MessageID[]	String	Error MessageID is returned if the method fails to execute
OUT	Message[]	String	Error Message in English corresponding to MessageID is returned if the method fails to execute

Qualifiers	Name	Type	Description/Values
OUT	MessageArguments[]	String	Substitution variables for dynamic error messages

Table 19. DCIM_LicenseManagementService.DeleteLicense() Method: Standard Messages

MessageID (OUT parameter)	Message	MessageArguments[]
LIC001	INFO: License Manager command not supported	
LIC002	INFO: Invalid command	
LIC003	INFO: One or more parameters in request are invalid	
LIC004	WARN: Unable to allocate needed resource(s)	
LIC006	CRITICAL: License Manager unable to allocate needed resources at startup	
LIC007	INFO: Invalid LDSR OID	
LIC008	INFO: LDSR OID not found	
LIC009	INFO: Specified device not found	
LIC010	INFO: The passed transaction ID is not found	
LIC011	INFO: License file does not exist	
LIC012	INFO: no license assigned to device	
LIC014	INFO: No license with specified EID exist in the system	
LIC015	INFO: No license with specified OID exist in the system	
LIC016	CRITICAL: License Manager has detected internal memory data corruption	
LIC019	INFO: EULA delete warning	
LIC039	WARN: File I/O error occurred when saving Licensable Device Service Record	
LIC046	INFO: Restart iDRAC to complete deletion of license	
LIC047	INFO: I/O error occurred writing evaluation history	
LIC900	The command was successful	
LIC901	Invalid parameter value for %s	FQDD, EntitlementID, DeleteOptions
LIC902	Resource allocation failure	
LIC903	Missing parameter(s) %s	DeleteOptions

8.4. DCIM_LicenseManagementService.ExportLicense()

ExportLicense is a method used to export License files from the iDRAC. The ExportLicense method returns a License, specified by Entitlement ID, as a base64 encoded string. The following tables specify ExportLicense return values and parameters.

Table 20. LicenseManagementService.ExportLicense() Method: Return Code Values

Value	Description
2	Error occurred
4096	Job started: REF returned to started CIM_ConcreteJob

Table 21. DCIM_LicenseManagementService.ExportLicense() Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	EntitlementID	String	Unique license ID
OUT	MessageID	String	Error MessageID is returned if the method fails to execute
OUT	Message[]	String	Error Message in English corresponding to MessageID is returned if the method fails to execute
OUT	MessageArguments[]	String	Substitution variables for dynamic error messages
OUT	LicenseFile	String	Base64 encoded string of the license file contents

Table 22. DCIM_LicenseManagementService.ExportLicense() Method: Standard Messages

MessageID (OUT parameter)	Message	MessageArguments[]
LIC001	INFO: License Manager command not supported	
LIC002	INFO: Invalid command	
LIC003	INFO: One or more parameters in request are invalid	
LIC004	WARN: Unable to allocate needed resource(s)	
LIC006	CRITICAL: License Manager unable to allocate needed resources at startup	
LIC007	INFO: Invalid LDSR OID	
LIC008	INFO: LDSR OID not found	
LIC009	INFO: Specified device not found	
LIC010	INFO: The passed transaction ID is not found	
LIC011	INFO: License file does not exist	
LIC012	INFO: no license assigned to device	
LIC014	INFO: No license with specified EID exist in the system	
LIC016	CRITICAL: License Manager has detected internal memory data corruption	
LIC036	INFO: File I/O error occurred when exporting license file	
LIC038	INFO: License manager unable to create the export license directory in /tmp	
LIC900	The command was successful	
LIC901	Invalid parameter value for %s	EntitlementID
LIC902	Resource allocation failure	
LIC903	Missing parameter(s) %s	EntitlementID

8.5. DCIM_LicenseManagementService.ExportLicenseByDevice()

ExportLicense is a method used to export License files from the iDRAC. The ExportLicense method returns all licenses assigned to a device, specified by FQDD, as a base64 encoded string. The following tables specify ExportLicenseByDevice return values and parameters.

Table 23. LicenseManagementService.ExportLicenseByDevice() Method: Return Code Values

Value	Description
2	Error occurred
4096	Job started: REF returned to started CIM_ConcreteJob

Table 24. DCIM_LicenseManagementService.ExportLicenseByDevice() Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	FQDD	String	Unique device ID
OUT	MessageID[]	String	Error MessageID is returned if the method fails to execute
OUT	Message[]	String	Error Message in English corresponding to MessageID is returned if the method fails to execute
OUT	MessageArguments[]	String	Substitution variables for dynamic error messages
OUT	LicenseFile[]	String	Base64 encoded string of the license file contents

Table 25. DCIM_LicenseManagementService.ExportLicenseByDevice() Method: Standard Messages

MessageID (OUT parameter)	Message	MessageArguments[]
LIC001	INFO: License Manager command not supported	
LIC002	INFO: Invalid command	
LIC003	INFO: One or more parameters in request are invalid	
LIC004	WARN: Unable to allocate needed resource(s)	
LIC006	CRITICAL: License Manager unable to allocate needed resources at startup	
LIC007	INFO: Invalid LDSR OID	
LIC008	INFO: LDSR OID not found	
LIC009	INFO: Specified device not found	
LIC010	INFO: The passed transaction ID is not found	
LIC011	INFO: License file does not exist	
LIC012	INFO: no license assigned to device	
LIC014	INFO: No license with specified EID exist in the system	
LIC016	CRITICAL: License Manager has detected internal memory data corruption	
LIC036	INFO: File I/O error occurred when exporting license file	
LIC038	INFO: License manager unable to create the export license directory in /tmp	
LIC900	The command was successful	
LIC901	Invalid parameter value for %s	FQDD

MessageID (OUT parameter)	Message	MessageArguments[]
LIC902	Resource allocation failure	
LIC903	Missing parameter(s) %s	FQDD

8.6. DCIM_LicenseManagementService.ExportLicenseToNetworkShare()

ExportLicenseToNetworkShare is a method used to export License files from the IDRAC. The ExportLicenseToNetworkShare method exports a License, specified by EntitlementID, to a user defined location. The following tables specify ExportLicenseToNetworkShare return values and parameters.

Table 26. DCIM_LicenseManagementService.ExportLicenseToNetworkShare() Method: Return Code Values

Value	Description
2	Error occurred
4096	Job started: REF returned to started CIM_ConcreteJob

Table 27. DCIM_LicenseManagementService.ExportLicenseToNetworkShare() Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	EntitlementID	String	The unique license ID
IN, REQ	ShareType	UInt16	Type of network share. (0 = NFS, 2 = CIFS, 5 = HTTP, 6 = HTTPS)
IN	Username	String	Username for Network share authentication
IN	Password	String	Password for Network share authentication
IN	Workgroup	String	Name of the workgroup for CIFS share authentication
IN	ShareName	string	Name of the CIFS share or full path to the NFS share. Optional for HTTP/HTTPS share, this may be treated as the path of the directory containing the file.
IN, REQ	IPAddress	string	IP address of the machine hosting the Network share.
IN	FileName	String	If included, the exported license is renamed as <FileName>
IN	ProxySupport	UInt16	It specifies if proxy is to be used or not. Default is 1 (Off). 1 → Off 2 → On - use default proxy settings 3 → On - use passed in parameters for proxy
IN	ProxyType	UInt16	It specifies the proxy type of the proxy server. Default is 0 (HTTP). 0 → HTTP 1 → SOCKS
IN	ProxyServer	String	This string represents the IP Address of the proxy server
IN	ProxyUname	String	This string represents the username for proxy server
IN	ProxyPasswd	String	This string represents the password to login into proxy server
IN	ProxyPort	String	This string represents the proxy port for the proxy server

Qualifiers	Name	Type	Description/Values
IN	IgnoreCertWarning	UInt16	Specifies if certificate warning is to be ignored when HTTPS is specified. If IgnoreCertWarning is on:certificate will be ignored. Default is 1 (Off). • 1 → Off • 2 → On
OUT	MessageID[]	String	Error MessageID is returned if the method fails to execute
OUT	Message[]	String	Error Message in English corresponding to MessageID is returned if the method fails to execute
OUT	MessageArguments[]	String	Substitution variables for dynamic error messages

Table 28. DCIM_LicenseManagementService.ExportLicenseToNetworkShare() Method: Standard Messages

MessageID (OUT parameter)	Message	MessageArguments[]
LIC001	INFO: License Manager command not supported	
LIC002	INFO: Invalid command	
LIC003	INFO: One or more parameters in request are invalid	
LIC004	WARN: Unable to allocate needed resource(s)	
LIC006	CRITICAL: License Manager unable to allocate needed resources at startup	
LIC007	INFO: Invalid LDSR OID	
LIC008	INFO: LDSR OID not found	
LIC009	INFO: Specified device not found	
LIC010	INFO: The passed transaction ID is not found	
LIC011	INFO: License file does not exist	
LIC012	INFO: no license assigned to device	
LIC014	INFO: No license with specified EID exist in the system	
LIC016	CRITICAL: License Manager has detected internal memory data corruption	
LIC036	INFO: File I/O error occurred when exporting license file	
LIC038	INFO: License manager unable to create the export license directory in /tmp	
LIC900	The command was successful	
LIC901	Invalid parameter value for %s	EntitlementID, ShareType, UserName, Password, Workgroup, ShareName, IPAddress, FileName
LIC902	Resource allocation failure	
LIC903	Missing parameter(s) %s	EntitlementID, ShareType, UserName, Password, ShareName, IPAddress

MessageID (OUT parameter)	Message	MessageArguments[]
LIC904	Could not connect to remote share	
JCP017	Maximum job limit reached, cannot create new jobs	

8.7. DCIM_LicenseManagementService.ExportLicenseByDeviceToNetworkShare()

ExportLicenseByDeviceToNetworkShare is a method used to export License files from the iDRAC. The ExportLicenseByDeviceToNetworkShare method exports all licenses from a device, specified by FQDD, to a user defined location. The following tables specify ExportLicenseByDeviceToNetworkShare return values and parameters.

Table 29. DCIM_LicenseManagementService.ExportLicenseByDeviceToNetworkShare() Method: Return Code Values

Value	Description
2	Error occurred
4096	Job started: REF returned to started CIM_ConcreteJob

Table 30. DCIM_LicenseManagementService.ExportLicenseByDeviceToNetworkShare() Method: Parameters

Qualifiers	Name	Type	Description/Values
IN, REQ	FQDD	String	The unique device ID
IN, REQ	ShareType	UInt16	Type of network share. (0 = NFS, 2 = CIFS)
IN, REQ	UserName	String	User name for CIFS share authentication
IN, REQ	Password	String	Password for CIFS share authentication
IN	Workgroup	String	Name of the workgroup for CIFS share authentication
IN, REQ	ShareName	String	Name of the CIFS share or full path to the NFS share
IN, REQ	IPAddress	String	IP Address of the machine hosting the CIFS/NFS share
OUT	MessageID[]	String	Error MessageID is returned if the method fails to execute
OUT	Message[]	String	Error Message in English corresponding to MessageID is returned if the method fails to execute
OUT	MessageArguments[]	String	Substitution variables for dynamic error messages

Table 31. DCIM_LicenseManagementService.ExportLicenseByDeviceToNetworkShare() Method: Standard Messages

MessageID (OUT parameter)	Message	MessageArguments[]
LIC001	INFO: License Manager command not supported	
LIC002	INFO: Invalid command	
LIC003	INFO: One or more parameters in request are invalid	
LIC004	WARN: Unable to allocate needed resource(s)	
LIC006	CRITICAL: License Manager unable to allocate needed resources at startup	
LIC007	INFO: Invalid LDSR OID	

MessageID (OUT parameter)	Message	MessageArguments[]
LIC008	INFO: LDSR OID not found	
LIC009	INFO: Specified device not found	
LIC010	INFO: The passed transaction ID is not found	
LIC011	INFO: License file does not exist	
LIC012	INFO: no license assigned to device	
LIC014	INFO: No license with specified EID exist in the system	
LIC016	CRITICAL: License Manager has detected internal memory data corruption	
LIC036	INFO: File I/O error occurred when exporting license file	
LIC038	INFO: License manager unable to create the export license directory in /tmp	
LIC900	The command was successful	
LIC901	Invalid parameter value for %s	FQDD, ShareType, UserName, Password, Workgroup, ShareName, IPAddress
LIC902	Resource allocation failure	
LIC903	Missing parameter(s) %s	FQDD, ShareType, UserName, Password, ShareName, IPAddress
LIC904	Could not connect to remote share	
JCP017	Maximum job limit reached, cannot create new jobs	

8.8. DCIM_LicenseManagementService.ReplaceLicense()

A method used to replace License Files in the License Manager. The ReplaceLicense() method replaces the license specified by EntitlementID on the device specified by FQDD with the License given in the input parameter LicenseFile.

The following tables specify ReplaceLicense return values and parameters.

Table 32. DCIM_LicenseManagementService.ReplaceLicense() Method: Return Code Values

Value	Description
2	Error occurred
4096	Job started: REF returned to started CIM_ConcreteJob

Table 33. DCIM_LicenseManagementService.ReplaceLicense() Method: Parameters

Qualifiers	Name	Type	Description/Values
IN , REQ	LicenseFile	String	A base64 encoded string of the XML License file
IN , REQ	EntitlementID	String	EntitlementID of the license being replaced
IN , REQ	FQDD	String	FQDD of the device the the license is being replaced on.
IN , REQ	ReplaceOptions	Integer	Flag to force or install for all like devices. no options=0, force =1 and all = 2
OUT	MessageID[]	String	Error MessageID is returned if the method fails to execute

Qualifiers	Name	Type	Description/Values
OUT	Message[]	String	Error Message in English corresponding to MessageID is returned if the method fails to execute
OUT	MessageArguments[]	String	Substitution variables for dynamic error messages

Table 34. DCIM_LicenseManagementService.ReplaceLicense() Method: Standard Messages

MessageID (OUT parameter)	Message	
LIC001	INFO: License Manager command not supported	
LIC002	INFO: Invalid command	
LIC003	INFO: One or more parameters in request are invalid	
LIC004	WARN: Unable to allocate needed resource(s)	
LIC005	INFO: Maximum number licenses installed	
LIC006	CRITICAL: License Manager unable to allocate needed resources at startup	
LIC007	INFO: Invalid LDSR OID	
LIC008	INFO: LDSR OID not found	
LIC009	INFO: Specified device not found	
LIC010	INFO: The passed transaction ID is not found	
LIC011	INFO: License file does not exist	
LIC012	INFO: no license assigned to device	
LIC014	INFO: No license with specified EID exist in the system	
LIC016	CRITICAL: License Manager has detected internal memory data corruption	
LIC017	INFO: License file signature invalid	
LIC018	INFO: License already exists in the system.	
LIC019	INFO: EULA delete warning	
LIC020	INFO: EULA import upgrade warning	
LIC021	INFO: Device binding did not match license binding data	
LIC022	INFO: System service tag did not match license service tag or system service tag not set	
LIC023	INFO: Upgrade license not an upgrade for existing license	
LIC024	INFO: License not for specified device	
LIC025	INFO: The system model does not support the license	
LIC026	INFO: The system generation does not support license	
LIC027	WARN: No device found suitable for template	
LIC028	INFO: License already imported to device	
LIC029	INFO: May not replace a non-evaluation license with an evaluation license	

MessageID (OUT parameter)	Message	
LIC030	INFO: License has expired	
LIC031	INFO: One or more license dates are not in ISO 8061 format	
LIC032	INFO: License file missing data	
LIC033	INFO: License file contains unknown data	
LIC034	INFO: License features' dependency are unmet	
LIC035	INFO: License not imported because license start date in the future	
LIC037	INFO: File I/O error occurred when writing license eval timers file	
LIC039	WARN: File I/O error occurred when saving Licensable Device Service Record	
LIC041	INFO: File I/O error occurred when saving license XML file	
LIC043	INFO: File I/O error occurred when deleting license file	
LIC045	INFO: Restart iDRAC to complete import of license	
LIC046	INFO: Restart iDRAC to complete deletion of license	
LIC047	INFO: I/O error occurred writing evaluation history	
LIC048	INFO: Evaluation license features already licensed	
LIC900	The command was successful	
LIC901	Invalid parameter value for %s	LicenseFile, EntitlementID, FQDD, ReplaceOptions
LIC902	Resource allocation failure	
LIC903	Missing parameter(s) %s	LicenseFile, EntitlementID, FQDD, ReplaceOptions

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 35. Privilege and License Requirements

Class and Method	Operation	User Privilege Required	License Required
DCIM_LicensableDevice	ENUMERATE, GET	Login	LM_LICENSE_UI
DCIM_License	ENUMERATE, GET	Login	LM_LICENSE_UI

Class and Method	Operation	User Privilege Required	License Required
DCIM_LicenseManagementService	ENUMERATE, GET	Login	LM_LICENSE_UI
DCIM_LicenseManagementService. ExportLicense()	INVOKE	Login	LM_LICENSE_UI
DCIM_LicenseManagementService. ExportLicenseToNetworkShare()	INVOKE	Login	LM_LICENSE_UI
DCIM_LicenseManagementService. ExportLicenseByDevice()	INVOKE	Login	LM_LICENSE_UI
DCIM_LicenseManagementService. ExportLicenseByDeviceToNetworkShare()	INVOKE	Login	LM_LICENSE_UI
DCIM_LicenseManagementService. ImportLicenseFromNetworkShare()	INVOKE	Login, System Control, Configure	LM_LICENSE_UI
DCIM_LicenseManagementService. ImportLicense()	INVOKE	Login, System Control, Configure	LM_LICENSE_UI
DCIM_LicenseManagementService. DeleteLicense()	INVOKE	Login, System Control, Configure	LM_LICENSE_UI
DCIM_LicenseManagementService. ReplaceLicense()	INVOKE	Login, System Control, Configure	LM_LICENSE_UI
DCIM_LCRegisteredProfile	ENUMERATE, GET	Login	None.
DCIM_LCElementConformsToProfile	ENUMERATE, GET	Login	None.

12. Change log

Version	Date	Description
4.0.0	20-Jun-2017	LicenseStartDate property of DCIM_License is now Optional. LicenseAttributes has a new ValueMap, 4 – “Valid Subcomponent” (14G specific License)