

---

# Understanding the iDRAC Power State Management Profile

---

*A Dell technical white paper.*

Chris A. Poblete

*Enterprise Product Group*



**This document is for informational purposes only and may contain typographical errors and technical inaccuracies. The content is provided as is, without express or implied warranties of any kind.**

© 2012 Dell Inc. All rights reserved. Dell and its affiliates cannot be responsible for errors or omissions in typography or photography. Dell, the Dell logo, and PowerEdge are trademarks of Dell Inc. Intel and Xeon are registered trademarks of Intel Corporation in the U.S. and other countries. Microsoft, Windows, and Windows Server are either trademarks or registered trademarks of Microsoft Corporation in the United States and/or other countries. Other trademarks and trade names may be used in this document to refer to either the entities claiming the marks and names or their products. Dell disclaims proprietary interest in the marks and names of others.

September 2012 | Rev 1.0

Contents

Executive summary ..... 4

Introduction ..... 4

Before you start ..... 4

Discovering the power state of the system (step 1) ..... 5

Discovering power state change capability (step 2) ..... 6

    What does this mean? ..... 7

Changing the system power state ..... 8

    EPR of the CIM\_ComputerSystem (step 3) ..... 8

    Resource URI for CIM\_PowerStateManagementService (step 4) ..... 9

    Invoking RequestPowerStateChange method (step 5) ..... 9

        I get a ReturnValue of 2, what is wrong? ..... 10

Power state management in CMC ..... 10

Scripting the examples ..... 11

Conclusion ..... 11

Tables

Table 1. PowerChangeCapabilities ..... 7

Table 2. PowerStatesSupported ..... 7

## Executive summary

This document is for system administrators or console application developers who are interested in power state management through the Remote Services API. Learn how to remotely determine the power state of the system and change the power state. This capability is provided by two profiles:

- The Base Server profile can operate on the host system and management controller.
- The Power State Management profile can operate on the host system only. The power state management in this document applies to the Power State Management profile.

## Introduction

Dell PowerEdge servers equipped with Integrated Dell Remote Access Controller (iDRAC) offers an agentless, out-of-band, remote management through the Remote Services API. Remote Services is also called Web Services since the interface is WS-Management (WS-MAN), a SOAP-based protocol that provides CIM-style data access. Remote Services is not an end-user interface but rather an API by which inputs can be a complex type and outputs may require complex processing and interpretation based on an API definition. API definitions are described in a profile specification.

### *DCIM Power State Management Profile*

The functionality described in this document applies to iDRAC and the Chassis Management Controller (CMC) in 11<sup>th</sup> and 12<sup>th</sup> generation Dell PowerEdge servers. The examples in this document use the `openwsman cli (wsmancli)` tool in Linux. When using the examples as a reference, replace the IP address, user name, and user password with real values. In the examples, the Response snippet means some parts of the response are intentionally excluded.

## Before you start

Check <http://support.dell.com> for the latest firmware update for iDRAC, Lifecycle Controller, and BIOS for your system. It is important to update your firmware to the latest version to prevent encountering problems that have already been fixed.

References related to this document:

- Profile specification  
[DCIM Power State Management Profile 1.0](#)
- Lifecycle Controller wiki with links to white papers, scripts, and more  
<http://delltechcenter.com/LC>
- DMTF profile specification documents  
<http://dmtf.org/standards/profiles>
- DCIM (Dell-extended) profile specification documents and accompanying MOFs  
<http://en.community.dell.com/techcenter/systems-management/w/wiki/1839.aspx>

Numbers are used throughout this document as a value for properties. Numerical properties are enumeration type. That is, the numerical value may only make sense when matched to a description

using the referenced profile specification or MOF file. For example, the value of 8 on the property *PowerState* refers to the corresponding profile specification where the value of 8 means the “system power state is off.”

## Discovering the power state of the system (step 1)

When remote, you may want to know whether one or more systems are powered on or off. This section describes how to accomplish this task.

The system power state is reported by the property *PowerState* from an instance of *CIM\_AssociatedPowerManagementService* class. This is an association class that links an instance of *CIM\_PowerManagementService* to an instance of *CIM\_ComputerSystem*.

Sample Command:

```
$ wsman enumerate
'http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_AssociatedPowerManagementService'
-h IPADDRESS -P 443 -u USER -p PASS -v -c server.cer -o -j utf-8 -y basic
```

Sample Response Snippet:

```
<n1:DCIM_CSAssociatedPowerManagementService>
<n1:OtherPowerState xsi:nil="true"/>
<n1:PowerState>8</n1:PowerState>
<n1:ServiceProvided>
  <wsa:Address>
    http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
  </wsa:Address>
  <wsa:ReferenceParameters>
    <wsman:ResourceURI>
      http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_CSPowerManagementService
    </wsman:ResourceURI>
    <wsman:SelectorSet>
      <wsman:Selector Name="CreationClassName">
        DCIM_CSPowerManagementService
      </wsman:Selector>
      <wsman:Selector Name="Name">pwrmtgsvc:1</wsman:Selector>
      <wsman:Selector Name="SystemName">systemmc</wsman:Selector>
      <wsman:Selector Name="SystemCreationClassName">
        DCIM_SPComputerSystem
      </wsman:Selector>
      <wsman:Selector Name="__cimnamespace">root/dcim</wsman:Selector>
    </wsman:SelectorSet>
  </wsa:ReferenceParameters>
</n1:ServiceProvided>
<n1:UserOfService>
  <wsa:Address>
    http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
  </wsa:Address>
  <wsa:ReferenceParameters>
    <wsman:ResourceURI>
      http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_ComputerSystem
    </wsman:ResourceURI>
    <wsman:SelectorSet>
      <wsman:Selector Name="Name">srv:system</wsman:Selector>
      <wsman:Selector Name="CreationClassName">DCIM_ComputerSystem</wsman:Selector>
      <wsman:Selector Name="__cimnamespace">root/dcim</wsman:Selector>
    </wsman:SelectorSet>
```

```
</wsa:ReferenceParameters>
</n1:UserOfService>
</n1:DCIM_CSAssociatedPowerManagementService>
```

The example shows the *PowerState* value of 8, which maps to “system is OFF”. It also shows that this is the power state for the system enclosure, which is identified by the instance property *Name* “srv:system”.

In one version of the firmware, you can see *PowerState* value of 12 which also maps to “system is OFF”. This is confusing because the *PowerState* enumeration is used for reporting power state and for describing the power state change. In the case for 8 and 12, they both have the same meaning when reporting as power state. However, they have different meanings when used for power state change. State change to 8 means to “power off the system” while 12 means to “power off the system but it is proceeded by a request to the host OS to perform its shutdown process”. This is sometimes called “graceful shutdown”.

The *PowerState* may have a value of 1 which maps to “Other.” Typically when you see the value “Other” in an enumeration, it means that the value being described is not included in the current enumeration list. To find the description of the value, look at the *OtherPowerState* property. An example value of this property could be “Powering On.”

Refer to section 7.3.1 of the DMTF profile specification for further explanation of *PowerState* values.

## Discovering power state change capability (step 2)

When remote, you may want to know whether you can power cycle/on/off one or more systems. This section describes how to accomplish this task.

The power state change capability is reported by the properties *PowerChangeCapabilities* and *PowerStatesSupported* from the CIM\_PowerManagementCapabilities class instance associated to the CIM\_PowerManagementService instance.

Sample Command:

```
$ wsman enumerate
'http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_PowerManagementCapabilities'
-h IPADDRESS -P 443 -u USER -p PASS -v -c server.cer -o -j utf-8 -y basic
```

Sample Response Snippet:

```
<wsman:Items>
  <n1:CIM_PowerManagementCapabilities>
    <n1:InstanceID>DCIM:pwrmtgcap1</n1:InstanceID>
    <n1:PowerChangeCapabilities>3</n1:PowerChangeCapabilities>
    <n1:PowerChangeCapabilities>4</n1:PowerChangeCapabilities>
    <n1:PowerChangeCapabilities>8</n1:PowerChangeCapabilities>
    <n1:PowerStatesSupported>2</n1:PowerStatesSupported>
    <n1:PowerStatesSupported>5</n1:PowerStatesSupported>
    <n1:PowerStatesSupported>8</n1:PowerStatesSupported>
    <n1:PowerStatesSupported>11</n1:PowerStatesSupported>
    <n1:PowerStatesSupported>12</n1:PowerStatesSupported>
  </n1:CIM_PowerManagementCapabilities>
</wsman:Items>
```

The example above shows (in blue text) power change capabilities of 3, 4 and 8; and (in green text) the supported power states that are used for RequestPowerStateChange method are 2, 5, 8, 11 and 12.

## What does this mean?

This question is best answered using the tables below. There are differences between versions of the firmware.

Table 1. PowerChangeCapabilities

11 <sup>th</sup> Generation Monolithic	11 <sup>th</sup> Generation Blades	12 <sup>th</sup> Generation	Description
3	3	3	Power state settable
4	4	4	Power cycle support
-	5	-	Timed power on support
-	6	-	Power cycle (hard) support
-	-	7	Hardware reset support
8	8	8	Graceful shutdown support

Similarities across firmware types are highlighted. Unless you have a specific need for capabilities that are *not* common, write code or scripts that work for all firmware versions. Refer to section 7.3.1 of the DMTF profile specification for further explanation of the description.

Table 2. PowerStatesSupported

11 <sup>th</sup> Generation Monolithic	11 <sup>th</sup> Generation Blades	12 <sup>th</sup> Generation	Description
2	2	2	Power on
5	5	5	Power cycle (soft)
-	6	-	Power off (hard)
8	8	8	Power off (soft)
-	9	-	Power cycle (hard)
-	-	10	Master bus reset
11	-	11	Diagnostic interrupt (NMI)
12	-	12	Power off (graceful)

Similarities across firmware types are highlighted. Unless you have a specific need for capabilities that are not common, write code or scripts that work for all firmware versions. Refer to section 7.3.1 of the DMTF profile specification for further explanation of the description.

## Changing the system power state

When remote, you may want to power cycle one or more systems. This section describes how you can accomplish this.

Changing the power state of the system requires the following steps:

1. Know the power state of the system. Refer to [Getting the power state of the system](#).
2. Know the power state change capability of the system. Refer to [Discovering Power State Change Capability](#).
3. Once the capability is determined, get the EPR of the CIM\_ComputerSystem class instance for the system. See [EPR of the CIM\\_ComputerSystem](#).
4. Get the EPR of the associated CIM\_PowerStateManagementService class instance. See [Resource URI for CIM\\_PowerStateManagementService](#).
5. Using the EPRs, run the RequestPowerStateChange() method with desired action. See [Invoking RequestPowerStateChange method](#)

Endpoint References (EPR) contains information needed to address a Web Service endpoint resource. Step 4 returns the resource, which is the instance of CIM\_PowerStateManagementService by which the method is invoked. The method requires the resource, returned by step 2, of the associated CIM\_ComputerSystem from which the action is to be applied.

The following sections describe steps 3, 4 and 5 in more detail.

### EPR of the CIM\_ComputerSystem (step 3)

In iDRAC, the property key *Name* for the host system is "srv:system". Because this is known, use the CQL filter in step 3 to limit the response to the exact instance for which we are looking.

Sample Step 3 Command:

```
$ wsman enumerate -M epr
'http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_ComputerSystem'
--dialect=http://schemas.dmtf.org/wbem/cql/1/dsp0202.pdf
--filter="select * from CIM_ComputerSystem where Name='srv:system'"
-h IPADDRESS -P 443 -u USER -p PASS -v -c server.cer -o -j utf-8 -y basic
```

Sample Step 3 Response Snippet:

```
<wsman:Items>
  <wsa:EndpointReference>
    <wsa:Address>
      http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
    </wsa:Address>
    <wsa:ReferenceParameters>
      <wsman:ResourceURI>
        http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_HostComputerSystem
      </wsman:ResourceURI>
      <wsman:SelectorSet>
        <wsman:Selector Name="__cimnamespace">root/dcim</wsman:Selector>
        <wsman:Selector Name="CreationClassName">DCIM_HostComputerSystem</wsman:Selector>
        <wsman:Selector Name="Name">srv:system</wsman:Selector>
      </wsman:SelectorSet>
    </wsa:ReferenceParameters>
```



```
</wsa:EndpointReference>
</wsman:Items>
```

## Resource URI for CIM\_PowerStateManagementService (step 4)

Using the associated filter in enumeration and the EPR of the CIM\_ComputerSystem in the previous step, get all the associated class instances, and from the response set, extract the EPR from CIM\_PowerStateManagementService class instance required for step 4. In iDRAC however, there is only one instance of CIM\_PowerStateManagementService. In this case, simply enumerate the class and get a single instance from the response.

Sample Step 4 Command:

```
$ wsman enumerate -M epr
'http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_PowerManagementService'
-h IPADDRESS -P 443 -u USER -p PASS -v -c server.cer -o -j utf-8 -y basic
```

Sample Step 4 Response Snippet:

```
<wsman:Items>
  <wsa:EndpointReference>
    <wsa:Address>
      http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
    </wsa:Address>
    <wsa:ReferenceParameters>
      <wsman:ResourceURI>
        http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_CSPowerManagementService
      </wsman:ResourceURI>
      <wsman:SelectorSet>
        <wsman:Selector Name="SystemCreationClassName">
          DCIM_SPCComputerSystem
        </wsman:Selector>
        <wsman:Selector Name="SystemName">systemmc</wsman:Selector>
        <wsman:Selector Name="CreationClassName">
          DCIM_CSPowerManagementService
        </wsman:Selector>
        <wsman:Selector Name="Name">pwrmgtsvc:1</wsman:Selector>
      </wsman:SelectorSet>
    </wsa:ReferenceParameters>
  </wsa:EndpointReference>
</wsman:Items>
```

## Invoking RequestPowerStateChange method (step 5)

Step 5 invokes a method that requires two parameters, one of which is a complex type that cannot be provided in a command line. The first parameter is *PowerState*. The example specifies 5 or “power cycle request.” The allowed values for this parameter come from the *PowerStatesSupported* enumeration that you get from [Discovering the power state of the system](#). The second parameter is the complex type *ManagedElement*. The value for this parameter comes from step 3 in this section and it is a copy-and-paste of the red text from the Response Snippet.

Sample Step 5 Input File “request.xml”:

```
<p:RequestPowerStateChange_INPUT xmlns:p=
  "http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_CSPowerManagementService">
  <p:PowerState>5</p:PowerState>
  <p:ManagedElement xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
    xmlns:wman="http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd">
    <wsa:Address>
```

```
http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
</wsa:Address>
<wsa:ReferenceParameters>
  <wsman:ResourceURI>
    http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_HostComputerSystem
  </wsman:ResourceURI>
  <wsman:SelectorSet>
    <wsman:Selector Name="__cimnamespace">root/dcim</wsman:Selector>
    <wsman:Selector Name="CreationClassName">DCIM_HostComputerSystem</wsman:Selector>
    <wsman:Selector Name="Name">srv:system</wsman:Selector>
  </wsman:SelectorSet>
</wsa:ReferenceParameters>
</p:ManagedElement>
</p:RequestPowerStateChange_INPUT>
```

The command in step 5 includes the input file from above and the resource URI constructed from the response in step 4 (red text in the Response Snippet).

Sample Step 5 Command:

```
$ wsman invoke -a RequestPowerStateChange -J request.xml
"http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_CSPowerManagementService?
SystemCreationClassName=DCIM_SPComputerSystem,SystemName=systemmc,
CreationClassName=DCIM_CSPowerManagementService,Name=pwrmtgsvc:1"
-h IPADDRESS -P 443 -u USER -p PASS -v -c server.cer -o -j utf-8 -y basic
```

Sample Step 5 Response Snippet:

```
<n1:RequestStateChange_OUTPUT>
  <n1:Job xsi:nil="true"/>
  <n1:ReturnValue>0</n1:ReturnValue>
</n1:RequestStateChange_OUTPUT>
```

The response reports a return value of 0 or "success". Refer to the profile specification for a complete mapping of the return value.

### I get a ReturnValue of 2, what is wrong?

If you get a return value of 2, chances are the state of the system is disabled or off. This is expected as the system is only allowed to power cycle when the state is enabled or on. This is why step 1 is important.

## Power state management in CMC

The Chassis Management Controller (CMC) in the Dell PowerEdge M1000e modular enclosure supports a Remote Services API implementation that conforms to DMTF Power State Management Profile. The method described in the previous sections to enumerate power state, enumerate power change capabilities, and invoke power state change also applies here.

As described in [step 3](#), provide the appropriate CIM\_ComputerSystem (CS) instance key property *Name*. Performing [step 1](#) provides you with a power state list and its associated CS instance. You will discover that the blade server power states are reported. This means that without awareness of the blade's service processor (iDRAC), you can power on, power off, or power cycle the blades using CMC.

## Scripting the examples

The examples in this document may be available in scripts for demonstration purposes. The available scripts are posted at the Dell TechCenter:

<http://en.community.dell.com/techcenter/systems-management/w/wiki/1981.scripting-the-dell-lifecycle-controller.aspx>

## Conclusion

From anywhere you have network access to servers that you manage; you can remotely and securely determine the power state and power off/on/or power cycle one or more servers using the Remote Services API in iDRAC. Script the Remote Services API using Powershell, Winrm, and Wsmancli (Linux).