
Understanding the iDRAC Power State Management in Base Server 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
Power State of the System.....	5
Power State of the Service Processor (iDRAC).....	5
Discovering power state change capability (step 2).....	6
Power Change Capability of the System.....	6
What does this mean?	6
Power Change Capability of the Service Processor (iDRAC).....	7
Changing the system power state	7
Get EPR of CIM_ComputerSystem (step 3)	7
Invoking RequestStateChange Method (step 4).....	8
I get a ReturnValue of 2, what is wrong?	8
Changing the Power State of the Service Processor (iDRAC).....	8
Scripting the Examples	9
Conclusion	9

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

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 Base Server and Physical Asset Profile

The functionality described in this document applies to iDRAC in 11th and 12th 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 Base Server and Physical Asset 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)

Scenario: You are remote and you want to know whether one or more systems you manage are powered on or off. This section describes how you can accomplish this.

The system power state is reported by the *EnabledState* property from an instance of CIM_ComputerSystem class. In iDRAC, there are two instances of this class. One instance refers to the system enclosure and the other instance refers to the service processor (or iDRAC).

Power State of the System

Sample Command:

```
$ wsman enumerate
'http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_ComputerSystem'
--dialect=http://schemas.dmtf.org/wbem/cql/1/dsp0202.pdf
--filter="select EnabledState,ElementName 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 Response Snippet:

```
<wsman:Items>
  <n1:CIM_ComputerSystem>
    <n1:CreationClassName>DCIM_ComputerSystem</n1:CreationClassName>
    <n1:ElementName>Computer System</n1:ElementName>
    <n1:EnabledState>2</n1:EnabledState>
    <n1:Name>srv:system</n1:Name>
  </n1:CIM_ComputerSystem>
</wsman:Items>
```

The power state of the system enclosure comes from the CIM_ComputerSystem instance with the key property *Name* having the value of “srv:system.” In the example above, the state has a value of 2, which maps to “system is ON”. Refer to the profile specification for the complete mapping of this property.

Power State of the Service Processor (iDRAC)

Sample Command:

```
$ wsman enumerate
'http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_ComputerSystem'
--dialect=http://schemas.dmtf.org/wbem/cql/1/dsp0202.pdf
--filter="select EnabledState,ElementName from CIM_ComputerSystem where Name='systemmc'"
-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_ComputerSystem>
    <n1:CreationClassName>DCIM_SPComputerSystem</n1:CreationClassName>
    <n1:ElementName>Service Processor Computer System</n1:ElementName>
    <n1:EnabledState>5</n1:EnabledState>
    <n1:Name>systemmc</n1:Name>
  </n1:CIM_ComputerSystem>
</wsman:Items>
```

```
</n1:CIM_ComputerSystem>  
</wsman:Items>
```

The power state of the service processor comes from the CIM_ComputerSystem instance with key property *Name* value of "systemmc." In the example above, the state has a value of 5 which maps to "Not Applicable". This indicates that the implementation is intentionally not reporting the power state. You may also see a value of 0 which maps to "Unknown". In this case, the power state is currently "Not Available". You may retry your request at a later time.

Discovering power state change capability (step 2)

Scenario: You are remote and you want to know whether you can power cycle/on/off a system or a service processor you manage. This section describes how you can accomplish this.

The power state change capability is reported by the *RequestedStatesSupported* property from the CIM_EnabledLogicalElementCapabilities class instance associated to the CIM_ComputerSystem instance.

Power Change Capability of the System

Sample Command:

```
$ wsman enumerate  
'http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_EnabledLogicalElementCapabilities'  
--dialect=http://schemas.dmtf.org/wbem/cql/1/dsp0202.pdf  
--filter="select RequestedStatesSupported,ElementName from  
    CIM_EnabledLogicalElementCapabilities where InstanceID='DCIM:ComputerCap:1'"  
-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_EnabledLogicalElementCapabilities>  
    <n1:ElementName>Computer System Capabilities</n1:ElementName>  
    <n1:InstanceID>DCIM:ComputerCap:1</n1:InstanceID>  
    <n1:RequestedStatesSupported>2</n1:RequestedStatesSupported>  
    <n1:RequestedStatesSupported>3</n1:RequestedStatesSupported>  
    <n1:RequestedStatesSupported>11</n1:RequestedStatesSupported>  
  </n1:CIM_EnabledLogicalElementCapabilities>  
</wsman:Items>
```

There is one instance of CIM_EnabledLogicalElementCapabilities class associated with CIM_ComputerSystem instance with key property *InstanceID* value of "DCIM:ComputerCap:1". In the example above, the supported states (in green text) are 2, 3, and 11 which correspond to Enabled, Disabled, and Reset.

What does this mean?

It reports "Enabled," which means you are able to power on the system when it is powered off. Likewise, it reports "Disabled," which means you are able to power off the system when it is powered on. Finally, it reports "Reset," which means you are able to cycle the power when it is powered on. Refer to the profile specification for the complete mapping of this property.

The "Reset" state is similar to the "PowerCycle-OffSoft" state in the Power State Management Profile specification.

Power Change Capability of the Service Processor (iDRAC)

The power change capability of the service processor is reported by the same class instance as the power change capability of the system. Refer to the previous section.

Changing the system power state

Scenario: You are remote and you want to be able to power cycle one or more systems you manage. 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 [Discovering 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 [Get EPR of CIM_ComputerSystem](#)
4. Using the EPR, execute the RequestStateChange() method with the desired action. See [Invoking RequestStateChange Method](#)

Endpoint References (EPR) contain information needed to address a Web Service endpoint resource. The resource is the instance of CIM_ComputerSystem by which the method is invoked.

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

Get EPR of CIM_ComputerSystem (step 3)

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

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>https://127.0.0.1:443/wsman</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>
```

Invoking RequestStateChange Method (step 4)

Step 4 invokes a method with one parameter *RequestedState*. The example specifies *11* or “reset power.” The allowed values for this parameter come from the *RequestedStatesSupported* enumeration you get from step 2. The resource URI for the command is constructed from the response in step 3. Notice the red text in step 4 command matches the red text in step 3 response snippet.

Step 4 Command:

```
$ wsman invoke -a RequestStateChange -k RequestedState=11  
"http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_HostComputerSystem?  
__cimnamespace=root/dcim,CreationClassName=DCIM_HostComputerSystem,Name=srv:system"  
-h IPADDRESS -P 443 -u USER -p PASS -v -c server.cer -o -j utf-8 -y basic
```

Sample Step 4 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.

Changing the Power State of the Service Processor (iDRAC)

The steps for power cycling the service processor are similar to power cycling the system described in the previous sections. The only difference is the command in step 3. Instead of 'srv:system', specify 'systemmc' which corresponds to the CIM_ComputerSystem class instance for the service process.

“Power off” and “power on” are not supported for the service processor.

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).