



# Creating Management Application using WS-Man API

This Dell Technical White Paper provides information about creating a management application using WS-Man API.

Dell Engineering  
June 2014

Chitrak Gupta  
Abhirup Seal



THIS WHITE PAPER 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.

© 2014 Dell Inc. All rights reserved. Reproduction of this material in any manner whatsoever without the express written permission of Dell Inc. is strictly forbidden. For more information, contact Dell.

Dell, the DELL logo, and the DELL badge are trademarks of Dell Inc. Symantec, NetBackup, and Backup Exec are trademarks of Symantec Corporation in the U.S. and other countries. Microsoft, Windows, and Windows Server are registered trademarks of Microsoft Corporation in the United States and/or other countries. Other trademarks and trade names may be used in this document to refer to either the entities claiming the marks and names or their products. Dell disclaims any proprietary interest in the marks and names of others.

Performance of network reference architectures discussed in this document may vary with differing deployment conditions, network loads, and the like. Third party products may be included in reference architectures for the convenience of the reader. Inclusion of such third-party products does not necessarily constitute Dell's recommendation of those products. Please consult your Dell representative for additional information.

Trademarks used in this text:

Dell™, the Dell logo, Dell Boomi™, Dell Precision™, OptiPlex™, Latitude™, PowerEdge™, PowerVault™, PowerConnect™, OpenManage™, EqualLogic™, Compellent™, KACE™, FlexAddress™, Force10™ and Vostro™ are trademarks of Dell Inc. Other Dell trademarks may be used in this document. Cisco Nexus®, Cisco MDS®, Cisco NX-OS®, and other Cisco Catalyst® are registered trademarks of Cisco System Inc. EMC VNX®, and EMC Unisphere® are registered trademarks of EMC Corporation. Intel®, Pentium®, Xeon®, Core® and Celeron® are registered trademarks of Intel Corporation in the U.S. and other countries. AMD® is a registered trademark and AMD Opteron™, AMD Phenom™ and AMD Sempron™ are trademarks of Advanced Micro Devices, Inc. Microsoft®, Windows®, Windows Server®, Internet Explorer®, MS-DOS®, Windows Vista® and Active Directory® are either trademarks or registered trademarks of Microsoft Corporation in the United States and/or other countries. Red Hat® and Red Hat® Enterprise Linux® are registered trademarks of Red Hat, Inc. in the United States and/or other countries. Novell® and SUSE® are registered trademarks of Novell Inc. in the United States and other countries. Oracle® is a registered trademark of Oracle Corporation and/or its affiliates. Citrix®, Xen®, XenServer® and XenMotion® are either registered trademarks or trademarks of Citrix Systems, Inc. in the United States and/or other countries. VMware®, Virtual SMP®, vMotion®, vCenter® and vSphere® are registered trademarks or trademarks of VMware, Inc. in the United States or other countries. IBM® is a registered trademark of International Business Machines Corporation. Broadcom® and



NetXtreme® are registered trademarks of Broadcom Corporation. Qlogic is a registered trademark of QLogic Corporation. Other trademarks and trade names may be used in this document to refer to either the entities claiming the marks and/or names or their products and are the property of their respective owners. Dell disclaims proprietary interest in the marks and names of others.



# Contents

Executive Summary .....	5
1 Introduction .....	5
2 Programmatic Interface .....	6
2.1 What is a programmatic interface? .....	6
2.2 A Real Time Scenario of a Web Services Based Application .....	6
2.3 Programmatic Interface versus Command Line Interface .....	8
3 WS-Man APIs .....	9
3.1 Example of A Fan device Class .....	9
3.2 MOF and WSDL Files .....	10
4 Understanding WSDLs .....	11
4.1 WSDL components .....	11
4.1.1 WSDL Ports .....	11
4.1.2 WSDL Messages .....	12
4.1.3 WSDL Types .....	12
4.1.4 WSDL Bindings .....	12
4.2 Example of WSDL .....	13
4.3 Parsing a WSDL to Use in Scripting .....	14
4.3.1 CIM Namespace .....	14
4.3.2 Resource URI .....	14
4.3.3 Methods and Method Parameters .....	14
4.3.4 Purpose of the WSDL .....	15
4.4 Example of a Python Code for Parsing WSDL .....	15
5 WSDL to SOAP XML .....	15
5.1 Example of Enumerate with Filter .....	16
5.2 Example of Invoke .....	17
6 Conclusion .....	18
7 Appendix-A .....	19
7.1 Sample Python Code to Parse a WSDL Document .....	19
8 Appendix-B .....	21
8.1 CIM_AuthorizationService.wsdl .....	21
9 Appendix C .....	38



# Executive Summary

This technical white paper is for the customers who use WS-MAN APIs to create scripts and management applications. It provides any management application developer, the required knowledge for using the WS-Man APIs and create their own management applications. The aim is to enable the developers to formulate a simple python based management application.

## 1 Introduction

In a large heterogenous data center, a management application helps to manage the data center assets by maintaining inventory, periodically collecting the health statistics, and providing incident management methods. All these are achieved through standard protocols because of the heterogeneity of the assets being managed. SNMP (Simple Network Management Protocol) is the most commonly used protocol by the administrators over the years. Traditionally utilized for inventory and monitoring, SNMP is not considered to be secure enough for configuration and does not have a mechanism for method invocation. DMTF (Distributed Management Task Force) has developed another protocol that provides a management mechanism in terms of object oriented modeling. This architecture is called SMASH (Systems Management Architecture for Server Hardware) and the protocol to access the object oriented Common Information Model (CIM) data based on standard web services is called WS-Man (Web services for management).

The management applications around the world have started using WS-Man for the ease of modeling and management of data center assets. For a description of how SNMP and WBEM (web based enterprise management) differs, see [Adoption of Wbem whitepaper](#).

This whitepaper aims to enable the management application developers to understand the WS-Man Application Programming Interface (API)s and the method to create an application for data center management using the programmatic interface. The whitepaper will describe the way a management application will parse Web Services Description Language (WSDL) files to understand the schema of the various SOAP requests to be sent for enumeration of data and invocation of methods. It will represent the inventory of data and invocation of methods using sample python code.

## 2 Programmatic Interface

### 2.1 What is a programmatic interface?

The Web Services for Management (WS-Management) specification describes a general Web services protocol based on SOAP for managing systems such as PCs, servers, devices, Web services and other applications, and other manageable entities. Services can display only a WS-Management interface or compose the WS-Management service interface with other Web service specifications. As per DMTF SMASH specification, it is one of the defined methods to access CIM data objects from a managed entity. The command line and the web services based protocols are defined in the SMASH architecture. The WS-Man specification (DSP-0226) describes the methods to manage an entity using SOAP requests. The WS-Man specification describes schema of different SOAP request packets that form the structure of WS-Man APIs. For example, if a management application wants to enumerate the instances of a class, it forms the SOAP request for an intrinsic\* enumeration method and sends the packet to the WS-Man endpoint. If the management application wants to use an extrinsic\* method, an intrinsic method is called, "InvokeMethod" is called and embed the details of the extrinsic method inside the body. These are standard request SOAP packets and hence any management application which has the capability to create these standard SOAP packet, based on the schema provided, is able to use the WS-Man infrastructure.

\*Note: Extrinsic method vs Intrinsic Method

- *Intrinsic method is defined by the CIM Operations over HTTP Specification for the purpose of modeling a CIM operation, Intrinsic methods are further characterized on the basis that they are made against a CIM Namespace;*
- *Extrinsic is defined as a method on a CIM Class in some Schema. Extrinsic methods are invoked on a CIM Class (if static) or Instance.*

### 2.2 A Real Time Scenario of a Web Services Based Application

The below figure represents a dynamic application that is created by reading WSDLs from two different web service providers.

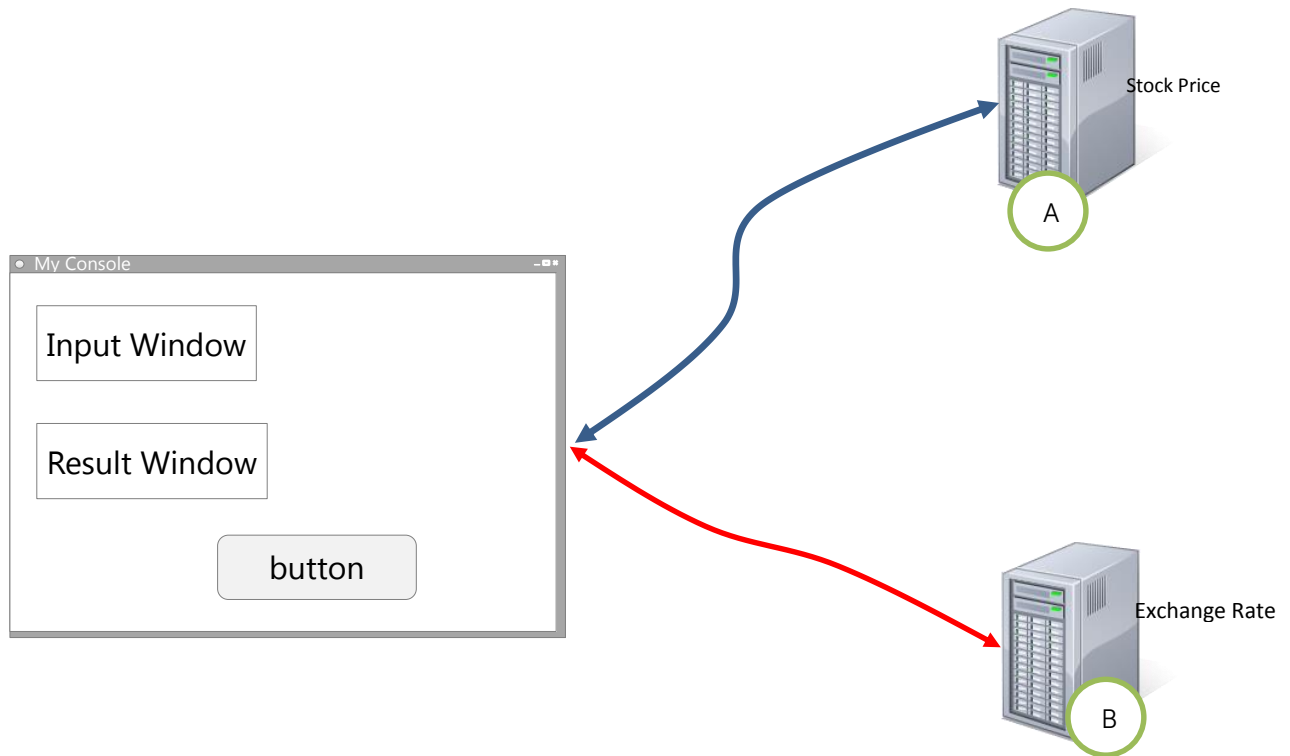


Figure 1 A Sample application

In Figure 1., Server A provides stock prices for the required stock name as input and the Server B provides currency exchange value from euros to dollar (considering the scenario with two currencies for simplification). The application knows the URL for both the servers. It is designed to be dynamic. When it connects to Server A, the “**button.text**” will changes to **Get Stock Price** and when it connects to Server B, the “**button.text**” changes to **Get Exchange Rate**. When the application connects to the servers A and B, it requests a web services API header file. The header file may be in a XML format as given below.

Server A	Server B
<Function-Header> <Function-Name> Get Stock Price </Function-Name> <Function Parameters> <Param1 type="string"> Stock-Name </Param1> </Function-Header>	<Function-Header> <Function-Name> Get Exchange Rate </Function-Name> <Function Parameters> <Param1 type="float"> Currency Value </Param1> </Function-Header>

The above XML is a generic representation on how the API header might look. When the application receives this XML, it parses out the data with the following pseudo code:

```
Button1.text=getXMLTag("Function-Name");  
Button1.OnClick("CreateRequest", Function-Name, text1.text);
```

On parsing the XML file , the Function-Name is derived. The application has a generic function to create a request which takes parameters as a function name and the argument derived from the input text. With this functionality, when the application receives API header it creates a request for a particular server. This becomes totally programmatic and dynamic. The header file in the XML format is formalized as WSDL (Web services description language) that enables management applications to create dynamic applications as described above.

## 2.3 Programmatic Interface versus Command Line Interface

The following table provides differences between the features of a Programmatic Interface and the command line interface.

Programmatic Interface	Command Line Interface
Programmatic interface provides management application and script developers with APIs to use for achieving the desired functionality.	Command line interfaces do not have APIs. They have commands following a standard grammar. Any management application or a script writer will use the command line calls directly.
Programmatic interfaces provide structured definitions of the APIs and they may be used in the management application following the appropriate documentation. Similar to any API, a complex work flow can be achieved by combining multiple APIs	Command line interfaces are distinct commands. Each command should be called Execution return and error code parsed from the shell stdout and stderr.
Based on web services, programmatic interfaces follow a standard schema definition of API headers in WSDL files.	Generally, command line interfaces are all proprietary and non-standard.
Complexity may be greater and API documentation is required.	Command line interfaces are usually associated with help and is quicker for a user to get started.



## 3 WS-Man APIs

WS-Man interface is a DMTF specification for systems management through Web Services architecture. WS-Man interface can be used to manage the server through a series of Simple Object Access Protocol (SOAP) encapsulated Web-Based Enterprise Management (WBEM) commands. **Enumeration** is used to inventory the devices under management and **Invoke** is used to change the states and configurations of a managed entity. Dell provides WS-Man APIs for management applications to enable systems management. For more information, see the [API documentation](#) and [best practices guide](#) available at Dell TechCenter.

WS-Man as an interface provides web services to access CIM data. All the managed entities in the server are represented by a class that includes data elements corresponding to properties for the device, and methods to control the behavior of the managed device.

### 3.1 Example of A Fan device Class

The following shows an example representation of a Fan device class.

```
class CIM_Fan : CIM_CoolingDevice
{
    boolean VariableSpeed;
    uint64 DesiredSpeed;
    uint16 ControlModesSupported[];
    uint16 ControlMode;
    uint16 DesiredControlMode;
    uint32 SetSpeed(uint64 DesiredSpeed);
};
```

The above is a class description of the fan device. It is derived from the CIM\_CoolingDevice class. The properties of the class shows different capabilities of the device. The method *SetSpeed* is used to control the speed of the fan in the system.

A management application that controls the fans on the server will “enumerate” the instances of the CIM\_Fan class. If the server has multiple fans, the same number of instances for the multiple fans are displayed. Each instance will have the properties populated and the *SetSpeed* method can be invoked on them to control their speeds.

The information about the class is available to the management applications in the WSDL format to use. The management applications can download the [WSDL files](#) and use them to create the workflows required for systems management.



## 3.2 MOF and WSDL Files

MOF (Managed Object Format) files are textual representation of the device classes and/or instances. The MOF files consists of the class names, their hierarchy, the properties and the methods. Qualifiers in the MOF files describe class, the properties and the methods.

The MOF files are compiled to form the CIM schema. Requests from the management applications are checked against the CIM schema for the validity of the request. To call the CIM objects accurately, the management applications need to have the knowledge of the classes. If the management application wants to call a method for a class, it needs to know the method prototype. That prototype information can be derived from the WSDL files.



## 4 Understanding WSDLs

WSDL (pronounced as WISDLE) expanding to Web Services Description Language is a document written in XML. The document describes a Web service, specifies the location of the service, and the operations (or methods) the service displays. WSDL files are XML files with the WSDL components:

### 4.1 WSDL components

A WSDL document describes a web service using the following major elements:

Element	Description
<b>&lt;types&gt;</b>	A container for data type definitions used by the web service
<b>&lt;message&gt;</b>	A typed definition of the data being communicated
<b>&lt;portType&gt;</b>	A set of operations supported by one or more endpoints
<b>&lt;binding&gt;</b>	A protocol and data format specification for a particular port type

The XML structure is:

```
<definitions>

<types>
  data type definitions.....
</types>

<message>
  definition of the data being communicated....
</message>

<portType>
  set of operations.....
</portType>

<binding>
  protocol and data format specification....
</binding>

</definitions>
```

#### 4.1.1 WSDL Ports

The <portType> element is the most important WSDL element. It describes a web service, the operations that can be performed, and the messages that are involved. The <portType> element can be compared to a function library (a module or a class) in a traditional programming language.



## 4.1.2 WSDL Messages

The <message> element defines the data elements of an operation.

Each message can include one or more parts. The parts can be compared to the parameters of a function call in a traditional programming language.

## 4.1.3 WSDL Types

The <types> element defines the data types that are used by the web service. For maximum platform neutrality, WSDL uses XML Schema syntax to define the data types.

## 4.1.4 WSDL Bindings

The <binding> element defines the data format and protocol for each port type.

For more information about WSDLs, see the link:

[http://www.w3schools.com/webservices/ws\\_wSDL\\_intro.asp](http://www.w3schools.com/webservices/ws_wSDL_intro.asp)



## 4.2 Example of WSDL

```
1. <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2. <wsdl:definitions ... targetNamespace="root/dcim/CIM_AuthorizationService">
3. ...
4. <wsdl:types>
5. <xs:schema ... targetNamespace="http://schemas.dmtf.org/wbem/wscim/1/cim-
   schema/2/CIM_AuthorizationService">
6. ...
7. <xs:element name="RequestStateChange_INPUT">
8. <xs:complexType>
9. <xs:sequence>
10. <xs:element name="RequestedState" nillable="true">
11. <xs:complexType>
12. <xs:simpleContent>
13. <xs:restriction base="cim:cimAnySimpleType">
14. <xs:simpleType>
15. <xs:union>
16. <xs:simpleType>
17. <xs:restriction base="xs:unsignedShort">
18. <xs:enumeration value="2"/>
19. ...
20. <xs:enumeration value="11"/>
21. </xs:restriction>
22. </xs:simpleType>
23. ...
24. <xs:anyAttribute namespace="##any" processContents="lax"/>
25. </xs:restriction>
26. </xs:simpleContent>
27. </xs:complexType>
28. </xs:element>
29. <xs:element name="TimeoutPeriod" nillable="true" type="cim:cimDateTime"/>
30. </xs:sequence>
31. </xs:complexType>
32. </xs:element>
33. </xs:schema>
34. </wsdl:types>
35. ...
36. <wsdl:message name="RequestStateChange_InputMessage">
37. <wsdl:part element="cimClass:RequestStateChange_INPUT" name="body"/>
38. </wsdl:message>
39. ...
40. <wsdl:portType name="CIM_AuthorizationService">
41. ...
42. <wsdl:operation name="RequestStateChange">
43. <wsdl:input message="thisWSDL:RequestStateChange_InputMessage" ... />
44. <wsdl:output message="thisWSDL:RequestStateChange_OutputMessage" ... />
45. </wsdl:operation>
46. </wsdl:portType>
47. </wsdl:definitions>
```



## 4.3 Parsing a WSDL to Use in Scripting

This section describes the procedure to parse a WSDL to collect the necessary information with which SOAP XMLs can be created. The example for WSDL in section 4.2 will be addressed as “example WSDL” throughout this section. The complete example WSDL can be found in Appendix-B. The following subsection provides the method to collect the required data.

### 4.3.1 CIM Namespace

Every CIM class has an associated WSDL file. In the example WSDL in the <wsdl:definitions> tag (line 2.), there is a tag attribute ‘targetnamespace’ whose value is in the format “CIMNamespace/ClassName”. Therefore, example WSDL is of the class **CIM\_AuthorizationService** that belongs to the **root/dcim** namespace.

### 4.3.2 Resource URI

The <xs:schema> tag (line 5.) in the example WSDL, , an attribute ‘targetNamespace’ whose value is the Resource URI for the class is provided. Therefore, from example WSDL, [http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM\\_AuthorizationService](http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_AuthorizationService) is provided as the Resource URI for the class.

### 4.3.3 Methods and Method Parameters

A few CIM Classes support methods that can be Invoked. A list of all these methods is available under the <wsdl:portType> tag (lines 40-46) in example WSDL. Under the <wsdl:portType> tag, there is a number of <wsdl:operation> tags, with each tag defining a method. In each <wsdl:operation> tag, the tag attribute ‘name’ reflects the name of the method. Under the <wsdl:operation> tag, the two tags, <wsdl:input> and <wsdl:output>, define the input(s) accepted by the method and the output given by the method. In the <wsdl:input> tag, the attribute message will provide the corresponding message name for the method’s input. A trace on example WSDL example will provide clarity on how the method signature is determined. In example WSDL, under the <wsdl:portType> tag, there is a <wsdl:operation> tag which informs that one of the methods supported is **RequestStateChange**. Under this tag, the <wsdl:input> tag’s attribute ‘message’ informs that the method’s message name is **thisWSDL:RequestStateChange\_InputMessage**. Then the corresponding <wsdl:message> tag (lines 36-43) whose attribute ‘name’ equals **RequestStateChange\_InputMessage** should be searched under this tag. A <wsdl:part> tag, whose attribute ‘element’ will provide the element name, it is **cimClass:RequestStateChange\_INPUT**.

<wsdl:types> tag (lines 4-34), which includes the <xs:schema> tag should be looked in next. In the <xs:schema> tag, <xs:element> tag whose attribute ‘name’ is **RequestStateChange\_INPUT** should be searched. This tag will provide all the details about all parameters the method accepts as input, their data types, restriction, that is the possible values that the parameter may assume or the maximum and minimum value of a parameter and so on. In this <xs:element> tag there may be (there can be methods which do not accept any input parameters) multiple <xs:element> tags and each of these will define an input parameter, their attribute ‘name’ specifying the parameter name. In example WSDL, the two parameters for the method **RequestStateChange**, ‘RequestedState’ and ‘TimeoutPeriod’ (lines 10 and 29) can be found. Additionally, each of these <xs:element> tags representing a parameter may have additional tags under them. As seen



in example WSDL, the tag for 'RequestedState' has additional tags (lines 10-28) and the tag for 'TimeoutPeriod' does not have any subtags (line 29). The <xs:element> tag for 'TimeoutPeriod' informs from its attribute 'type' that the datatype is 'datetime'. Similarly, the datatype of 'RequestedState', the tag informs that it is unsignedShort (line 17) along with the restrictions that is a list of possible values, 2 ... 11.. For information about the complete WSDL, see Appendix-B.

#### 4.3.4 Purpose of the WSDL

The WSDL functions as an .XSD file for any data the class returns, which means that the data returned by any operation on the class MUST match the Schema Definition as defined by its WSDL.

### 4.4 Example of a Python Code for Parsing WSDL

The python script in Appendix-A provides the process to parse any WSDL to collect necessary information which is used later to create SOAP XMLs for WS-MAN API. The script is written in Python language version 2.7.5. As shown in Appendix-A, when the program is run with main() called with a path to the required WSDL file, here. **CIM\_AuthorizationService.wsdl**, it parses the WSDL and generates the following data,

```
Class: CIM_AuthorizationService
CIMNamespace: root/dcim
ResourceURI: http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_AuthorizationService
Methods:
RequestStateChange (
    RequestedState # Possible Values ['2', '3', '4', '6', '7', '8', '9', '10', '11', '5']
    TimeoutPeriod )
StartService ( )
StopService ( )
ChangeAffectedElementsAssignedSequence (
    AssignedSequence
    ManagedElements )
```

As seen above, the WSDL was parsed and the output includes the ClassName, CIMNamespace, ResourceURI, and the list of methods supported along with their input parameters and possible values where applicable.

## 5 WSDL to SOAP XML

A SOAP message is an XML document that is used to transmit data. WSDL is an XML document that describes the method to connect and send requests to a web service. The WSDL document is not required for the soap communication, but it is a required element according to the protocol with regard to the formation and validation of SOAP request/response. This section describes the method to use the information collected by parsing the WSDLs. A brief overview with an example Enumerate command and Invoke command is provided here. For detailed information about the method to create SOAP XMLs, see the DMTF WS-Man specifications in the following link,

[http://www.dmtf.org/sites/default/files/standards/documents/DSP0226\\_1.0.0.pdf](http://www.dmtf.org/sites/default/files/standards/documents/DSP0226_1.0.0.pdf)



## 5.1 Example of Enumerate with Filter

The example provided in this section gets a list of Virtual MAC Addresses of the NIC Cards from the DCIM\_NICAttribute\* Class. The WSDL for DCIM\_NICAttribute is used to figure out the ResourceUri and the SelectorSet. This information is used to create the following SOAP XML:

\*Note: DCIM classes are Dell derivation of CIM schema classes.

```
<?xml version="1.0" encoding="utf-8"?>
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:wsen="http://schemas.xmlsoap.org/ws/2004/09/enumeration" xmlns:WSMan
="http://schemas.dmtf.org/wbem/WSMan /1/WSMan .xsd">
  <s:Header>
    <wsa:Action
s:mustUnderstand="true">http://schemas.xmlsoap.org/ws/2004/09/enumeration/Enumerate</wsa:
Action>
    <wsa:To s:mustUnderstand="true">https://IPADDRESS/WSMan </wsa:To>
    <WSMan :ResourceURI s:mustUnderstand="true">http://schemas.dell.com/wbem/wscim/1/cim-
schema/2/DCIM_NICAttribute</WSMan :ResourceURI>
    <wsa:MessageID s:mustUnderstand="true">urn:uuid:4a2768a1-e7f4-11e3-a7c4-
2cd05a84c9be</wsa:MessageID>
    <WSMan :OperationTimeout>PT300.0S</WSMan :OperationTimeout>
    <WSMan :MaxEnvelopeSize>153600</WSMan :MaxEnvelopeSize>
    <wsa:ReplyTo>
<wsa:Address>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address
>
    </wsa:ReplyTo>
    <WSMan :SelectorSet>
    <WSMan :Selector Name="__ cimnamespace">root/dcim</WSMan :Selector>
    </WSMan :SelectorSet>
    <WSMan :FragmentTransfer s:mustUnderstand="true">CurrentValue</WSMan
:FragmentTransfer>
  </s:Header>
  <s:Body>
    <wsen:Enumerate>
    <WSMan :OptimizeEnumeration/>
    <WSMan :MaxElements>32000</WSMan :MaxElements>
    <WSMan :Filter Dialect="http://schemas.dmtf.org/wbem/cql/1/dsp0202.pdf" xmlns:WSMan
="http://schemas.dmtf.org/wbem/WSMan /1/WSMan .xsd">select * from DCIM_NICAttribute where
AttributeName = "VirtMacAddr"</WSMan :Filter>
    </wsen:Enumerate>
  </s:Body>
</s:Envelope>
```





## 5.2 Example of Invoke

The example in section provides information about invoking the method SetAttribute to set a particular attribute in DCIM\_NICAttribute. The WSDL of DCIM\_NICService is used to figure out ResourceURI and the method input parameters. Using that information and user input, the following SOAP XML is created.

```
<?xml version="1.0" encoding="utf-8"?>
<s:Envelope xmlns:n1="http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_NICService"
xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:WSMan
="http://schemas.dmtf.org/wbem/WSMan /1/WSMan .xsd">
  <s:Header>
    <wsa:Action s:mustUnderstand="true">http://schemas.dell.com/wbem/wscim/1/cim-
schema/2/DCIM_NICService/SetAttribute</wsa:Action>
    <wsa:To s:mustUnderstand="true">https://IPADDRESS/WSMan </wsa:To>
    <WSMan :ResourceURI s:mustUnderstand="true">http://schemas.dell.com/wbem/wscim/1/cim-
schema/2/DCIM_NICService</WSMan :ResourceURI>
    <wsa:MessageID s:mustUnderstand="true">urn:uuid:f5ffc540-ea20-11e3-8173-
2cd05a84c9be</wsa:MessageID>
    <WSMan :OperationTimeout>PT300.0S</WSMan :OperationTimeout>
    <wsa:ReplyTo>
<wsa:Address>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address
>
    </wsa:ReplyTo>
    <WSMan :SelectorSet>
      <WSMan :Selector Name="__cimnamespace">root/dcim</WSMan :Selector>
      <WSMan :Selector Name="SystemCreationClassName">DCIM_ComputerSystem</WSMan
:Selector>
      <WSMan :Selector Name="SystemName">DCIM:ComputerSystem</WSMan :Selector>
      <WSMan :Selector Name="CreationClassName">DCIM_NICService</WSMan :Selector>
      <WSMan :Selector Name="Name">DCIM:NICService</WSMan :Selector>
    </WSMan :SelectorSet>
  </s:Header>
  <s:Body>
    <n1:SetAttribute_INPUT>
      <n1:AttributeName>UseIndTgtPortal</n1:AttributeName>
      <n1:AttributeValue>Disabled</n1:AttributeValue>
      <n1:Target>NIC.Integrated.1-2-1</n1:Target>
    </n1:SetAttribute_INPUT>
  </s:Body>
</s:Envelope>
```



## 6 Conclusion

WS-Man as a programmatic interface is a powerful interface providing a comprehensive API suite to create a systems management application. WS-Man APIs not only provide capabilities of device inventory, monitoring and management but also the methods for event subscription and incident management. Using WS-MAN API, any management application developer can provide complete standard based systems management which works across a heterogeneous environment. This whitepaper aims to help a developer to get started with the WS-MAN API and provide direction of parsing the WSDL files and creating a basic systems management application.



## 7 Appendix-A

This appendix provides a sample Python code that parses a WSDL document.

### 7.1 Sample Python Code to Parse a WSDL Document

```
import xml.etree.ElementTree as ET

def getClassname(xml):
    targetNamespace = xml.attrib['targetNamespace']
    className=targetNamespace[targetNamespace.rfind('/')+1:]
    return className

def getCIMNamespace(xml):
    targetNamespace = xml.attrib['targetNamespace']
    cimNamespace=targetNamespace[:targetNamespace.rfind('/')]
    return cimNamespace

def getResourceURI(xml):
    namespaces={'wsdl':"http://schemas.xmlsoap.org/wsdl/", "xs":"http://www.w3.org/2001/XMLSchema"}

    resourceUri=xml.find('wsdl:types', namespaces=namespaces).find('xs:schema', namespaces=namespaces).attrib[
'targetNamespace']
    return resourceUri

def getMethods(xml):
    methods={}
    namespaces={'wsdl':"http://schemas.xmlsoap.org/wsdl/", "xs":"http://www.w3.org/2001/XMLSchema"}
    try:

operationTags=xml.find('wsdl:portType', namespaces=namespaces).findall('wsdl:operation', namespaces=namesp
aces)
        for operationTag in operationTags:
            methodName=operationTag.attrib['name']
            parameters={}
            messageName=operationTag.find('wsdl:input', namespaces=namespaces).attrib['name']
            for messageTag in xml.findall('wsdl:message', namespaces=namespaces):
                if messageTag.attrib['name'] == messageName:
                    elementName =
messageTag.find('wsdl:part', namespaces=namespaces).attrib['element'].split(':')[1]
                    found=False
                    for elementTag in
xml.find('wsdl:types', namespaces=namespaces).find('xs:schema', namespaces=namespaces).findall('xs:element
', namespaces=namespaces):
                        try:
                            if elementTag.attrib['name']==elementName:
                                found=True
                                break
                        except Exception as e:
                            pass
                    if not found:
                        return {}
                    for tag in elementTag.iter():
                        if tag.tag.split(':')[1]=='element':
```



```

        if tag.attrib['name'] == elementName:
            continue
        paramName=tag.attrib['name']
        possibleValues=[]
        for subtag in tag.iter():
            if subtag.tag.split('.')[1]=='enumeration':
                try:
                    possibleValues.append(subtag.attrib['value'])
                except Exception as e:
                    pass
            try:
                if tag.attrib['nillable'] == 'true':
                    nillable=True
            except Exception as e:
                nillable=False
        parameters[paramName]=(possibleValues,nillable)
    methods[methodname]=parameters
    return methods
except Exception as e:
    print e
    return {}

def main(filename):
    wsdl = ET.parse(filename)
    xml=wsdl.getroot()
    print "Class:", getClassName(xml)
    print "CIMNamespace:", getCIMNamespace(xml)
    print "ResourceURI:", getResourceURI(xml)
    try:
        print "Methods:"
        methods=getMethods(xml)
        for method in methods.keys():
            print "%s(" %(method),

            for parameter in methods[method].keys():
                print ' '
                print "\t%s " %parameter,
                if methods[method][parameter][0]:
                    print "# Possible Values %s" %(methods[method][parameter][0]),
            print ")"
    except Exception as e:
        pass

if __name__=="__main__":
    main('CIM_AuthorizationService.wsdl')

```



## 8 Appendix-B

This appendix provides the full WSDL which was shown in section 4.2.

### 8.1 CIM\_AuthorizationService.wsdl

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!--Generated by Dell, Inc. WS-Browser--><wsdl:definitions
xmlns:cimClass="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_AuthorizationService"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:thisWSDL="root/dcim/CIM_AuthorizationService"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" targetNamespace="root/dcim/CIM_AuthorizationService">
  <wsdl:import location="http://www.w3.org/2005/08/addressing/ws-addr.xsd"
namespace="http://www.w3.org/2005/08/addressing"/>
  <wsdl:types>
    <xs:schema xmlns:cim="http://schemas.dmtf.org/wbem/wscim/1/common"
xmlns:class="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_AuthorizationService"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
elementFormDefault="qualified" targetNamespace="http://schemas.dmtf.org/wbem/wscim/1/cim-
schema/2/CIM_AuthorizationService">
      <xs:import namespace="http://schemas.dmtf.org/wbem/wscim/1/common"
schemaLocation="http://schemas.dmtf.org/wbem/wscim/1/common.xsd"/>
      <xs:element name="SystemCreationClassName">
        <xs:complexType>
          <xs:simpleContent>
            <xs:restriction base="cim:cimString">
              <xs:maxLength value="256"/>
              <xs:anyAttribute namespace="##any" processContents="lax"/>
            </xs:restriction>
          </xs:simpleContent>
        </xs:complexType>
      </xs:element>
      <xs:element name="SystemName">
        <xs:complexType>
          <xs:simpleContent>
            <xs:restriction base="cim:cimString">
              <xs:maxLength value="256"/>
              <xs:anyAttribute namespace="##any" processContents="lax"/>
            </xs:restriction>
          </xs:simpleContent>
        </xs:complexType>
      </xs:element>
      <xs:element name="CreationClassName">
        <xs:complexType>
          <xs:simpleContent>
            <xs:restriction base="cim:cimString">
              <xs:maxLength value="256"/>
              <xs:anyAttribute namespace="##any" processContents="lax"/>
            </xs:restriction>
          </xs:simpleContent>
        </xs:complexType>
      </xs:element>
      <xs:element name="Name">
        <xs:complexType>
          <xs:simpleContent>
            <xs:restriction base="cim:cimString">
              <xs:maxLength value="256"/>
              <xs:anyAttribute namespace="##any" processContents="lax"/>
            </xs:restriction>
          </xs:simpleContent>
        </xs:complexType>
      </xs:element>
    </xs:schema>
  </wsdl:types>

```



```

        </xs:restriction>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="PrimaryOwnerName" nillable="true">
    <xs:complexType>
      <xs:simpleContent>
        <xs:restriction base="cim:cimString">
          <xs:maxLength value="64"/>
          <xs:anyAttribute namespace="##any" processContents="lax"/>
        </xs:restriction>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="PrimaryOwnerContact" nillable="true">
    <xs:complexType>
      <xs:simpleContent>
        <xs:restriction base="cim:cimString">
          <xs:maxLength value="256"/>
          <xs:anyAttribute namespace="##any" processContents="lax"/>
        </xs:restriction>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="StartMode" nillable="true">
    <xs:complexType>
      <xs:simpleContent>
        <xs:restriction base="cim:cimString">
          <xs:maxLength value="10"/>
          <xs:enumeration value="Automatic"/>
          <xs:enumeration value="Manual"/>
          <xs:anyAttribute namespace="##any" processContents="lax"/>
        </xs:restriction>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="Started" nillable="true" type="cim:cimBoolean"/>
  <xs:element name="EnabledState" nillable="true">
    <xs:complexType>
      <xs:simpleContent>
        <xs:restriction base="cim:cimAnySimpleType">
          <xs:simpleType>
            <xs:union>
              <xs:simpleType>
                <xs:restriction base="xs:unsignedShort">
                  <xs:enumeration value="0"/>
                  <xs:enumeration value="1"/>
                  <xs:enumeration value="2"/>
                  <xs:enumeration value="3"/>
                  <xs:enumeration value="4"/>
                  <xs:enumeration value="5"/>
                  <xs:enumeration value="6"/>
                  <xs:enumeration value="7"/>
                  <xs:enumeration value="8"/>
                  <xs:enumeration value="9"/>
                  <xs:enumeration value="10"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:union>
          </xs:simpleType>
        </xs:restriction>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>

```



```

        <xs:simpleType>
            <xs:restriction base="xs:unsignedShort">
                <xs:minInclusive value="11"/>
                <xs:maxInclusive value="32767"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:union>
    <xs:simpleType>
        <xs:restriction base="xs:unsignedShort">
            <xs:minInclusive value="32768"/>
            <xs:maxInclusive value="65535"/>
        </xs:restriction>
    </xs:simpleType>
</xs:union>
</xs:simpleType>
<xs:anyAttribute namespace="##any" processContents="lax"/>
</xs:restriction>
</xs:simpleContent>
</xs:complexType>
</xs:element>
<xs:element name="OtherEnabledState" nillable="true" type="cim:cimString"/>
<xs:element name="RequestedState" nillable="true">
    <xs:complexType>
        <xs:simpleContent>
            <xs:restriction base="cim:cimAnySimpleType">
                <xs:simpleType>
                    <xs:union>
                        <xs:simpleType>
                            <xs:restriction base="xs:unsignedShort">
                                <xs:enumeration value="0"/>
                                <xs:enumeration value="2"/>
                                <xs:enumeration value="3"/>
                                <xs:enumeration value="4"/>
                                <xs:enumeration value="5"/>
                                <xs:enumeration value="6"/>
                                <xs:enumeration value="7"/>
                                <xs:enumeration value="8"/>
                                <xs:enumeration value="9"/>
                                <xs:enumeration value="10"/>
                                <xs:enumeration value="11"/>
                                <xs:enumeration value="12"/>
                            </xs:restriction>
                        </xs:simpleType>
                        <xs:simpleType>
                            <xs:union>
                                <xs:simpleType>
                                    <xs:restriction base="xs:unsignedShort">
                                        <xs:enumeration value="1"/>
                                    </xs:restriction>
                                </xs:simpleType>
                                <xs:simpleType>
                                    <xs:restriction base="xs:unsignedShort">
                                        <xs:minInclusive value="13"/>
                                        <xs:maxInclusive value="32767"/>
                                    </xs:restriction>
                                </xs:simpleType>
                            </xs:union>
                        </xs:simpleType>
                    </xs:union>
                </xs:simpleType>
            </xs:restriction>
        </xs:simpleContent>
    </xs:complexType>

```



```

        <xs:restriction base="xs:unsignedShort">
            <xs:minInclusive value="32768"/>
            <xs:maxInclusive value="65535"/>
        </xs:restriction>
    </xs:simpleType>
</xs:union>
</xs:simpleType>
    <xs:anyAttribute namespace="##any" processContents="lax"/>
</xs:restriction>
</xs:simpleContent>
</xs:complexType>
</xs:element>
<xs:element name="EnabledDefault" nillable="true">
    <xs:complexType>
        <xs:simpleContent>
            <xs:restriction base="cim:cimAnySimpleType">
                <xs:simpleType>
                    <xs:union>
                        <xs:simpleType>
                            <xs:restriction base="xs:unsignedShort">
                                <xs:enumeration value="2"/>
                                <xs:enumeration value="3"/>
                                <xs:enumeration value="5"/>
                                <xs:enumeration value="6"/>
                                <xs:enumeration value="7"/>
                                <xs:enumeration value="9"/>
                            </xs:restriction>
                        </xs:simpleType>
                        <xs:simpleType>
                            <xs:union>
                                <xs:simpleType>
                                    <xs:restriction base="xs:unsignedShort">
                                        <xs:maxInclusive value="1"/>
                                    </xs:restriction>
                                </xs:simpleType>
                                <xs:simpleType>
                                    <xs:restriction base="xs:unsignedShort">
                                        <xs:enumeration value="4"/>
                                    </xs:restriction>
                                </xs:simpleType>
                                <xs:simpleType>
                                    <xs:restriction base="xs:unsignedShort">
                                        <xs:enumeration value="8"/>
                                    </xs:restriction>
                                </xs:simpleType>
                                <xs:simpleType>
                                    <xs:restriction base="xs:unsignedShort">
                                        <xs:minInclusive value="10"/>
                                        <xs:maxInclusive value="32767"/>
                                    </xs:restriction>
                                </xs:simpleType>
                            </xs:union>
                        </xs:simpleType>
                    </xs:union>
                </xs:simpleType>
            </xs:restriction>
        </xs:simpleType>
    </xs:complexType>

```





```

        </xs:simpleType>
    </xs:union>
</xs:simpleType>
<xs:anyAttribute namespace="##any" processContents="lax"/>
</xs:restriction>
</xs:simpleContent>
</xs:complexType>
</xs:element>
<xs:element name="TimeOfLastStateChange" nillable="true" type="cim:cimDateTime"/>
<xs:element name="AvailableRequestedStates" nillable="true">
    <xs:complexType>
        <xs:simpleContent>
            <xs:restriction base="cim:cimAnySimpleType">
                <xs:simpleType>
                    <xs:union>
                        <xs:simpleType>
                            <xs:restriction base="xs:unsignedShort">
                                <xs:enumeration value="2"/>
                                <xs:enumeration value="3"/>
                                <xs:enumeration value="4"/>
                                <xs:enumeration value="6"/>
                                <xs:enumeration value="7"/>
                                <xs:enumeration value="8"/>
                                <xs:enumeration value="9"/>
                                <xs:enumeration value="10"/>
                                <xs:enumeration value="11"/>
                            </xs:restriction>
                        </xs:simpleType>
                        <xs:simpleType>
                            <xs:union>
                                <xs:simpleType>
                                    <xs:restriction base="xs:unsignedShort">
                                        <xs:maxInclusive value="1"/>
                                    </xs:restriction>
                                </xs:simpleType>
                                <xs:simpleType>
                                    <xs:restriction base="xs:unsignedShort">
                                        <xs:enumeration value="5"/>
                                    </xs:restriction>
                                </xs:simpleType>
                                <xs:simpleType>
                                    <xs:restriction base="xs:unsignedShort">
                                        <xs:minInclusive value="12"/>
                                    </xs:restriction>
                                </xs:simpleType>
                            </xs:union>
                        </xs:simpleType>
                    </xs:union>
                </xs:simpleType>
            </xs:restriction>
        </xs:simpleContent>
    </xs:complexType>
</xs:element>
<xs:element name="TransitioningToState" nillable="true">
    <xs:complexType>
        <xs:simpleContent>
            <xs:restriction base="cim:cimAnySimpleType">

```



```

<xs:simpleType>
  <xs:union>
    <xs:simpleType>
      <xs:restriction base="xs:unsignedShort">
        <xs:enumeration value="0"/>
        <xs:enumeration value="2"/>
        <xs:enumeration value="3"/>
        <xs:enumeration value="4"/>
        <xs:enumeration value="5"/>
        <xs:enumeration value="6"/>
        <xs:enumeration value="7"/>
        <xs:enumeration value="8"/>
        <xs:enumeration value="9"/>
        <xs:enumeration value="10"/>
        <xs:enumeration value="11"/>
        <xs:enumeration value="12"/>
      </xs:restriction>
    </xs:simpleType>
    <xs:simpleType>
      <xs:union>
        <xs:simpleType>
          <xs:restriction base="xs:unsignedShort">
            <xs:enumeration value="1"/>
          </xs:restriction>
        </xs:simpleType>
        <xs:simpleType>
          <xs:restriction base="xs:unsignedShort">
            <xs:minInclusive value="13"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:union>
    </xs:simpleType>
  </xs:union>
</xs:simpleType>
<xs:anyAttribute namespace="##any" processContents="lax"/>
</xs:restriction>
</xs:simpleContent>
</xs:complexType>
</xs:element>
<xs:element name="InstallDate" nillable="true" type="cim:cimDateTime"/>
<xs:element name="OperationalStatus" nillable="true">
  <xs:complexType>
    <xs:simpleContent>
      <xs:restriction base="cim:cimAnySimpleType">
        <xs:simpleType>
          <xs:union>
            <xs:simpleType>
              <xs:restriction base="xs:unsignedShort">
                <xs:enumeration value="0"/>
                <xs:enumeration value="1"/>
                <xs:enumeration value="2"/>
                <xs:enumeration value="3"/>
                <xs:enumeration value="4"/>
                <xs:enumeration value="5"/>
                <xs:enumeration value="6"/>
                <xs:enumeration value="7"/>
                <xs:enumeration value="8"/>
                <xs:enumeration value="9"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:union>
        </xs:simpleType>
      </xs:restriction>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>

```



```

        <xs:enumeration value="10"/>
        <xs:enumeration value="11"/>
        <xs:enumeration value="12"/>
        <xs:enumeration value="13"/>
        <xs:enumeration value="14"/>
        <xs:enumeration value="15"/>
        <xs:enumeration value="16"/>
        <xs:enumeration value="17"/>
        <xs:enumeration value="18"/>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType>
    <xs:union>
        <xs:simpleType>
            <xs:restriction base="xs:unsignedShort">
                <xs:minInclusive value="19"/>
                <xs:maxInclusive value="32767"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:union>
</xs:simpleType>
<xs:simpleType>
    <xs:restriction base="xs:unsignedShort">
        <xs:minInclusive value="32768"/>
    </xs:restriction>
</xs:simpleType>
</xs:union>
</xs:simpleType>
    <xs:anyAttribute namespace="##any" processContents="lax"/>
</xs:restriction>
</xs:simpleContent>
</xs:complexType>
</xs:element>
<xs:element name="StatusDescriptions" nillable="true" type="cim:cimString"/>
<xs:element name="Status" nillable="true">
    <xs:complexType>
        <xs:simpleContent>
            <xs:restriction base="cim:cimString">
                <xs:maxLength value="10"/>
                <xs:enumeration value="OK"/>
                <xs:enumeration value="Error"/>
                <xs:enumeration value="Degraded"/>
                <xs:enumeration value="Unknown"/>
                <xs:enumeration value="Pred Fail"/>
                <xs:enumeration value="Starting"/>
                <xs:enumeration value="Stopping"/>
                <xs:enumeration value="Service"/>
                <xs:enumeration value="Stressed"/>
                <xs:enumeration value="NonRecover"/>
                <xs:enumeration value="No Contact"/>
                <xs:enumeration value="Lost Comm"/>
                <xs:enumeration value="Stopped"/>
                <xs:anyAttribute namespace="##any" processContents="lax"/>
            </xs:restriction>
        </xs:simpleContent>
    </xs:complexType>
</xs:element>
<xs:element name="HealthState" nillable="true">

```



```

<xs:complexType>
  <xs:simpleContent>
    <xs:restriction base="cim:cimAnySimpleType">
      <xs:simpleType>
        <xs:union>
          <xs:simpleType>
            <xs:restriction base="xs:unsignedShort">
              <xs:enumeration value="0"/>
              <xs:enumeration value="5"/>
              <xs:enumeration value="10"/>
              <xs:enumeration value="15"/>
              <xs:enumeration value="20"/>
              <xs:enumeration value="25"/>
              <xs:enumeration value="30"/>
            </xs:restriction>
          </xs:simpleType>
          <xs:simpleType>
            <xs:union>
              <xs:simpleType>
                <xs:restriction base="xs:unsignedShort">
                  <xs:minInclusive value="1"/>
                  <xs:maxInclusive value="4"/>
                </xs:restriction>
              </xs:simpleType>
              <xs:simpleType>
                <xs:restriction base="xs:unsignedShort">
                  <xs:minInclusive value="6"/>
                  <xs:maxInclusive value="9"/>
                </xs:restriction>
              </xs:simpleType>
              <xs:simpleType>
                <xs:restriction base="xs:unsignedShort">
                  <xs:minInclusive value="11"/>
                  <xs:maxInclusive value="14"/>
                </xs:restriction>
              </xs:simpleType>
              <xs:simpleType>
                <xs:restriction base="xs:unsignedShort">
                  <xs:minInclusive value="16"/>
                  <xs:maxInclusive value="19"/>
                </xs:restriction>
              </xs:simpleType>
              <xs:simpleType>
                <xs:restriction base="xs:unsignedShort">
                  <xs:minInclusive value="21"/>
                  <xs:maxInclusive value="24"/>
                </xs:restriction>
              </xs:simpleType>
              <xs:simpleType>
                <xs:restriction base="xs:unsignedShort">
                  <xs:minInclusive value="26"/>
                  <xs:maxInclusive value="29"/>
                </xs:restriction>
              </xs:simpleType>
              <xs:simpleType>
                <xs:restriction base="xs:unsignedShort">
                  <xs:minInclusive value="31"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:union>
          </xs:simpleType>
        </xs:union>
      </xs:simpleType>
    </xs:restriction>
  </xs:simpleContent>
</xs:complexType>

```



```

        </xs:simpleType>
    </xs:union>
</xs:simpleType>
</xs:union>
</xs:simpleType>
    <xs:anyAttribute namespace="##any" processContents="lax"/>
</xs:restriction>
</xs:simpleContent>
</xs:complexType>
</xs:element>
<xs:element name="CommunicationStatus" nillable="true">
    <xs:complexType>
        <xs:simpleContent>
            <xs:restriction base="cim:cimAnySimpleType">
                <xs:simpleType>
                    <xs:union>
                        <xs:simpleType>
                            <xs:restriction base="xs:unsignedShort">
                                <xs:enumeration value="0"/>
                                <xs:enumeration value="1"/>
                                <xs:enumeration value="2"/>
                                <xs:enumeration value="3"/>
                                <xs:enumeration value="4"/>
                            </xs:restriction>
                        </xs:simpleType>
                        <xs:simpleType>
                            <xs:union>
                                <xs:simpleType>
                                    <xs:restriction base="xs:unsignedShort">
                                        <xs:minInclusive value="5"/>
                                        <xs:maxInclusive value="32767"/>
                                    </xs:restriction>
                                </xs:simpleType>
                            </xs:union>
                        </xs:simpleType>
                        <xs:simpleType>
                            <xs:restriction base="xs:unsignedShort">
                                <xs:minInclusive value="32768"/>
                            </xs:restriction>
                        </xs:simpleType>
                    </xs:union>
                </xs:simpleType>
            <xs:anyAttribute namespace="##any" processContents="lax"/>
        </xs:restriction>
    </xs:simpleContent>
</xs:complexType>
</xs:element>
<xs:element name="DetailedStatus" nillable="true">
    <xs:complexType>
        <xs:simpleContent>
            <xs:restriction base="cim:cimAnySimpleType">
                <xs:simpleType>
                    <xs:union>
                        <xs:simpleType>
                            <xs:restriction base="xs:unsignedShort">
                                <xs:enumeration value="0"/>
                                <xs:enumeration value="1"/>
                                <xs:enumeration value="2"/>
                            </xs:restriction>
                        </xs:simpleType>
                    </xs:union>
                </xs:simpleType>
            </xs:restriction>
        </xs:simpleContent>
    </xs:complexType>
</xs:element>

```



```

        <xs:enumeration value="3"/>
        <xs:enumeration value="4"/>
        <xs:enumeration value="5"/>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType>
    <xs:union>
        <xs:simpleType>
            <xs:restriction base="xs:unsignedShort">
                <xs:minInclusive value="6"/>
                <xs:maxInclusive value="32767"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:union>
</xs:simpleType>
<xs:simpleType>
    <xs:restriction base="xs:unsignedShort">
        <xs:minInclusive value="32768"/>
    </xs:restriction>
</xs:simpleType>
</xs:union>
</xs:simpleType>
    <xs:anyAttribute namespace="##any" processContents="lax"/>
</xs:restriction>
</xs:simpleContent>
</xs:complexType>
</xs:element>
<xs:element name="OperatingStatus" nillable="true">
    <xs:complexType>
        <xs:simpleContent>
            <xs:restriction base="cim:cimAnySimpleType">
                <xs:simpleType>
                    <xs:union>
                        <xs:simpleType>
                            <xs:restriction base="xs:unsignedShort">
                                <xs:enumeration value="0"/>
                                <xs:enumeration value="1"/>
                                <xs:enumeration value="2"/>
                                <xs:enumeration value="3"/>
                                <xs:enumeration value="4"/>
                                <xs:enumeration value="5"/>
                                <xs:enumeration value="6"/>
                                <xs:enumeration value="7"/>
                                <xs:enumeration value="8"/>
                                <xs:enumeration value="9"/>
                                <xs:enumeration value="10"/>
                                <xs:enumeration value="11"/>
                                <xs:enumeration value="12"/>
                                <xs:enumeration value="13"/>
                                <xs:enumeration value="14"/>
                                <xs:enumeration value="15"/>
                                <xs:enumeration value="16"/>
                            </xs:restriction>
                        </xs:simpleType>
                    </xs:union>
                </xs:simpleType>
            </xs:restriction>
        </xs:simpleContent>
    </xs:complexType>
</xs:element>

```



```

        <xs:minInclusive value="17"/>
        <xs:maxInclusive value="32767"/>
    </xs:restriction>
</xs:simpleType>
</xs:union>
</xs:simpleType>
<xs:simpleType>
    <xs:restriction base="xs:unsignedShort">
        <xs:minInclusive value="32768"/>
    </xs:restriction>
</xs:simpleType>
</xs:union>
</xs:simpleType>
<xs:anyAttribute namespace="##any" processContents="lax"/>
</xs:restriction>
</xs:simpleContent>
</xs:complexType>
</xs:element>
<xs:element name="PrimaryStatus" nillable="true">
    <xs:complexType>
        <xs:simpleContent>
            <xs:restriction base="cim:cimAnySimpleType">
                <xs:simpleType>
                    <xs:union>
                        <xs:simpleType>
                            <xs:restriction base="xs:unsignedShort">
                                <xs:enumeration value="0"/>
                                <xs:enumeration value="1"/>
                                <xs:enumeration value="2"/>
                                <xs:enumeration value="3"/>
                            </xs:restriction>
                        </xs:simpleType>
                        <xs:simpleType>
                            <xs:union>
                                <xs:simpleType>
                                    <xs:restriction base="xs:unsignedShort">
                                        <xs:minInclusive value="4"/>
                                        <xs:maxInclusive value="32767"/>
                                    </xs:restriction>
                                </xs:simpleType>
                            </xs:union>
                        </xs:simpleType>
                        <xs:simpleType>
                            <xs:restriction base="xs:unsignedShort">
                                <xs:minInclusive value="32768"/>
                            </xs:restriction>
                        </xs:simpleType>
                    </xs:union>
                </xs:simpleType>
                <xs:anyAttribute namespace="##any" processContents="lax"/>
            </xs:restriction>
        </xs:simpleContent>
    </xs:complexType>
</xs:element>
<xs:element name="InstanceID" type="cim:cimString"/>
<xs:element name="Caption" nillable="true">
    <xs:complexType>
        <xs:simpleContent>

```



```

        <xs:restriction base="cim:cimString">
            <xs:maxLength value="64"/>
            <xs:anyAttribute namespace="##any" processContents="lax"/>
        </xs:restriction>
    </xs:simpleContent>
</xs:complexType>
</xs:element>
<xs:element name="Description" nillable="true" type="cim:cimString"/>
<xs:element name="ElementName" nillable="true" type="cim:cimString"/>
<xs:element name="Generation" nillable="true" type="cim:cimUnsignedLong"/>
<xs:element name="StartService_INPUT">
    <xs:complexType/>
</xs:element>
<xs:element name="StartService_OUTPUT">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="ReturnValue" nillable="true" type="cim:cimUnsignedInt"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="StopService_INPUT">
    <xs:complexType/>
</xs:element>
<xs:element name="StopService_OUTPUT">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="ReturnValue" nillable="true" type="cim:cimUnsignedInt"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="ChangeAffectedElementsAssignedSequence_INPUT">
    <xs:complexType>
        <xs:sequence>
            <xs:element maxOccurs="unbounded" name="ManagedElements" type="cim:cimReference"/>
            <xs:element maxOccurs="unbounded" name="AssignedSequence" type="cim:cimUnsignedShort"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="ChangeAffectedElementsAssignedSequence_OUTPUT">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Job" nillable="true" type="cim:cimReference"/>
            <xs:element name="ReturnValue" nillable="true">
                <xs:complexType>
                    <xs:simpleContent>
                        <xs:restriction base="cim:cimAnySimpleType">
                            <xs:simpleType>
                                <xs:union>
                                    <xs:simpleType>
                                        <xs:restriction base="xs:unsignedInt">
                                            <xs:enumeration value="0"/>
                                            <xs:enumeration value="1"/>
                                            <xs:enumeration value="2"/>
                                            <xs:enumeration value="3"/>
                                            <xs:enumeration value="4"/>
                                            <xs:enumeration value="5"/>
                                            <xs:enumeration value="6"/>
                                        </xs:restriction>
                                    </xs:simpleType>
                                </xs:union>
                            </xs:simpleType>
                        </xs:restriction>
                    </xs:simpleContent>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>

```





```

        </xs:simpleType>
        <xs:simpleType>
            <xs:restriction base="xs:unsignedInt">
                <xs:minInclusive value="7"/>
                <xs:maxInclusive value="32767"/>
            </xs:restriction>
        </xs:simpleType>
        <xs:simpleType>
            <xs:restriction base="xs:unsignedInt">
                <xs:minInclusive value="32768"/>
                <xs:maxInclusive value="65535"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:union>
</xs:simpleType>
    <xs:anyAttribute namespace="##any" processContents="lax"/>
</xs:restriction>
</xs:simpleContent>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="RequestStateChange_INPUT">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="RequestedState" nillable="true">
                <xs:complexType>
                    <xs:simpleContent>
                        <xs:restriction base="cim:cimAnySimpleType">
                            <xs:simpleType>
                                <xs:union>
                                    <xs:simpleType>
                                        <xs:restriction base="xs:unsignedShort">
                                            <xs:enumeration value="2"/>
                                            <xs:enumeration value="3"/>
                                            <xs:enumeration value="4"/>
                                            <xs:enumeration value="6"/>
                                            <xs:enumeration value="7"/>
                                            <xs:enumeration value="8"/>
                                            <xs:enumeration value="9"/>
                                            <xs:enumeration value="10"/>
                                            <xs:enumeration value="11"/>
                                        </xs:restriction>
                                    </xs:simpleType>
                                    <xs:simpleType>
                                        <xs:union>
                                            <xs:simpleType>
                                                <xs:restriction base="xs:unsignedShort">
                                                    <xs:maxInclusive value="1"/>
                                                </xs:restriction>
                                            </xs:simpleType>
                                            <xs:simpleType>
                                                <xs:restriction base="xs:unsignedShort">
                                                    <xs:enumeration value="5"/>
                                                </xs:restriction>
                                            </xs:simpleType>
                                        </xs:union>
                                    </xs:simpleType>
                                </xs:union>
                            </xs:simpleType>
                        </xs:restriction>
                    </xs:simpleContent>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>

```



```

        <xs:restriction base="xs:unsignedShort">
            <xs:minInclusive value="12"/>
            <xs:maxInclusive value="32767"/>
        </xs:restriction>
    </xs:simpleType>
</xs:union>
</xs:simpleType>
<xs:simpleType>
    <xs:restriction base="xs:unsignedShort">
        <xs:minInclusive value="32768"/>
        <xs:maxInclusive value="65535"/>
    </xs:restriction>
</xs:simpleType>
</xs:union>
</xs:simpleType>
<xs:anyAttribute namespace="##any" processContents="lax"/>
</xs:restriction>
</xs:simpleContent>
</xs:complexType>
</xs:element>
<xs:element name="TimeoutPeriod" nillable="true" type="cim:cimDateTime"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="RequestStateChange_OUTPUT">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Job" nillable="true" type="cim:cimReference"/>
            <xs:element name="ReturnValue" nillable="true">
                <xs:complexType>
                    <xs:simpleContent>
                        <xs:restriction base="cim:cimAnySimpleType">
                            <xs:simpleType>
                                <xs:union>
                                    <xs:simpleType>
                                        <xs:restriction base="xs:unsignedInt">
                                            <xs:enumeration value="0"/>
                                            <xs:enumeration value="1"/>
                                            <xs:enumeration value="2"/>
                                            <xs:enumeration value="3"/>
                                            <xs:enumeration value="4"/>
                                            <xs:enumeration value="5"/>
                                            <xs:enumeration value="6"/>
                                        </xs:restriction>
                                    </xs:simpleType>
                                    <xs:simpleType>
                                        <xs:union>
                                            <xs:simpleType>
                                                <xs:restriction base="xs:unsignedInt">
                                                    <xs:minInclusive value="7"/>
                                                    <xs:maxInclusive value="4095"/>
                                                </xs:restriction>
                                            </xs:simpleType>
                                            <xs:simpleType>
                                                <xs:restriction base="xs:unsignedInt">
                                                    <xs:minInclusive value="32769"/>
                                                </xs:restriction>
                                            </xs:simpleType>
                                        </xs:union>
                                    </xs:simpleType>
                                </xs:union>
                            </xs:simpleType>
                        </xs:restriction>
                    </xs:simpleContent>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>

```



```

        </xs:union>
      </xs:simpleType>
    <xs:simpleType>
      <xs:restriction base="xs:unsignedInt">
        <xs:enumeration value="4096"/>
        <xs:enumeration value="4097"/>
        <xs:enumeration value="4098"/>
        <xs:enumeration value="4099"/>
      </xs:restriction>
    </xs:simpleType>
    <xs:simpleType>
      <xs:restriction base="xs:unsignedInt">
        <xs:minInclusive value="4100"/>
        <xs:maxInclusive value="32767"/>
      </xs:restriction>
    </xs:simpleType>
    <xs:simpleType>
      <xs:restriction base="xs:unsignedInt">
        <xs:minInclusive value="32768"/>
        <xs:maxInclusive value="65535"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:union>
</xs:simpleType>
  <xs:anyAttribute namespace="##any" processContents="lax"/>
</xs:restriction>
</xs:simpleContent>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="CIM_AuthorizationService" type="class:CIM_AuthorizationService_Type"/>
<xs:complexType name="CIM_AuthorizationService_Type">
  <xs:sequence>
    <xs:element maxOccurs="unbounded" minOccurs="0" ref="class:AvailableRequestedStates"/>
    <xs:element minOccurs="0" ref="class:Caption"/>
    <xs:element minOccurs="0" ref="class:CommunicationStatus"/>
    <xs:element ref="class:CreationClassName"/>
    <xs:element minOccurs="0" ref="class:Description"/>
    <xs:element minOccurs="0" ref="class:DetailedStatus"/>
    <xs:element minOccurs="0" ref="class:ElementName"/>
    <xs:element minOccurs="0" ref="class:EnabledDefault"/>
    <xs:element minOccurs="0" ref="class:EnabledState"/>
    <xs:element minOccurs="0" ref="class:Generation"/>
    <xs:element minOccurs="0" ref="class:HealthState"/>
    <xs:element minOccurs="0" ref="class:InstallDate"/>
    <xs:element ref="class:InstanceID"/>
    <xs:element ref="class:Name"/>
    <xs:element minOccurs="0" ref="class:OperatingStatus"/>
    <xs:element maxOccurs="unbounded" minOccurs="0" ref="class:OperationalStatus"/>
    <xs:element minOccurs="0" ref="class:OtherEnabledState"/>
    <xs:element minOccurs="0" ref="class:PrimaryOwnerContact"/>
    <xs:element minOccurs="0" ref="class:PrimaryOwnerName"/>
    <xs:element minOccurs="0" ref="class:PrimaryStatus"/>
    <xs:element minOccurs="0" ref="class:RequestedState"/>
    <xs:element minOccurs="0" ref="class:StartMode"/>
    <xs:element minOccurs="0" ref="class:Started"/>
  </xs:sequence>

```



```

        <xs:element minOccurs="0" ref="class:Status"/>
        <xs:element maxOccurs="unbounded" minOccurs="0" ref="class:StatusDescriptions"/>
        <xs:element ref="class:SystemCreationClassName"/>
        <xs:element ref="class:SystemName"/>
        <xs:element minOccurs="0" ref="class:TimeOfLastStateChange"/>
        <xs:element minOccurs="0" ref="class:TransitioningToState"/>
        <xs:any maxOccurs="unbounded" minOccurs="0" namespace="##other" processContents="lax"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##any" processContents="lax"/>
</xs:complexType>
</xs:schema>
</wsdl:types>
<wsdl:message name="StartService_InputMessage">
    <wsdl:part element="cimClass:StartService_INPUT" name="body"/>
</wsdl:message>
<wsdl:message name="StartService_OutputMessage">
    <wsdl:part element="cimClass:StartService_OUTPUT" name="body"/>
</wsdl:message>
<wsdl:message name="StopService_InputMessage">
    <wsdl:part element="cimClass:StopService_INPUT" name="body"/>
</wsdl:message>
<wsdl:message name="StopService_OutputMessage">
    <wsdl:part element="cimClass:StopService_OUTPUT" name="body"/>
</wsdl:message>
<wsdl:message name="ChangeAffectedElementsAssignedSequence_InputMessage">
    <wsdl:part element="cimClass:ChangeAffectedElementsAssignedSequence_INPUT" name="body"/>
</wsdl:message>
<wsdl:message name="ChangeAffectedElementsAssignedSequence_OutputMessage">
    <wsdl:part element="cimClass:ChangeAffectedElementsAssignedSequence_OUTPUT" name="body"/>
</wsdl:message>
<wsdl:message name="RequestStateChange_InputMessage">
    <wsdl:part element="cimClass:RequestStateChange_INPUT" name="body"/>
</wsdl:message>
<wsdl:message name="RequestStateChange_OutputMessage">
    <wsdl:part element="cimClass:RequestStateChange_OUTPUT" name="body"/>
</wsdl:message>
<wsdl:portType name="CIM_AuthorizationService">
    <wsdl:operation name="StartService">
        <wsdl:input message="thisWSDL:StartService_InputMessage" name="StartService_InputMessage"/>
        <wsdl:output message="thisWSDL:StartService_OutputMessage" name="StartService_OutputMessage"/>
    </wsdl:operation>
    <wsdl:operation name="StopService">
        <wsdl:input message="thisWSDL:StopService_InputMessage" name="StopService_InputMessage"/>
        <wsdl:output message="thisWSDL:StopService_OutputMessage" name="StopService_OutputMessage"/>
    </wsdl:operation>
    <wsdl:operation name="ChangeAffectedElementsAssignedSequence">
        <wsdl:input message="thisWSDL:ChangeAffectedElementsAssignedSequence_InputMessage"
name="ChangeAffectedElementsAssignedSequence_InputMessage"/>
        <wsdl:output message="thisWSDL:ChangeAffectedElementsAssignedSequence_OutputMessage"
name="ChangeAffectedElementsAssignedSequence_OutputMessage"/>
    </wsdl:operation>
    <wsdl:operation name="RequestStateChange">
        <wsdl:input message="thisWSDL:RequestStateChange_InputMessage"
name="RequestStateChange_InputMessage"/>
        <wsdl:output message="thisWSDL:RequestStateChange_OutputMessage"
name="RequestStateChange_OutputMessage"/>
    </wsdl:operation>
</wsdl:portType>

```



```
</wsdl:definitions><!--Generated by Dell, Inc. WS-Browser -->
```



## 9 Appendix C

List of abbreviations:

Acronym	Definition
DMTF	Distributed Management Task Force
CIM	Common Information Model
CIM-OM	CIM-Object Manager
WS-Man	Web Services for Management
SOAP	Simple Object Access Protocol
WSDL	Web Services Description Language
SMASH	System Management Architecture for Server Hardware
MOF	Managed object format

