Filter Dialects

This Dell Technical White Paper provides information about use of filters in WS-Man enumeration and subscription

Dell Engineering
April 2014

Abhirup Seal
Chitrak Gupta

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.

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.

Contents

Executive Summary 4

1 Introduction 5

2 WS-Man Filters 5

2.1 Usage of Filter 5

2.2 Selector Filter 6

2.2.1 Combining Selectors 7

2.2.2 SOAP Request Format 7

2.3 Query Languages (CQL and WQL) 8

2.3.1 Enumeration and Query Languages 8

2.4 Dialects and Associations 9

2.4.1 What are Associations? 9

2.4.2 Associators 9

2.4.3 References 12

3 Filtering on Subscriptions 14

3.1 Subscription Management 14

3.2 Usage from Different Consoles 15

3.2.1 Windows PowerShell 15

3.2.2 Linux WS-Man CLI 15

4 Fragment Transfer 16

5 Conclusion 17

6 Appendix A 18

Executive Summary

This technical white paper is aimed at the customers of WS-Man API’s to understand the usage of filter dialects in enumeration and subscription requests. This document assumes that the reader has knowledge of the basic WS-Man specification and the usage. For detailed information about WS-Man protocol specification, refer to the WS-Man Specification (DSP-0226) at www.dmtf.org .

1 Introduction

PowerEdge servers and chassis for modular servers, include management controllers, iDRAC and CMC (Chassis Management Controller). These controllers assist the system administrators to manage and monitor the chassis and servers. The management controllers provide functionality of management through GUI, CLI, and programmatic (WS-Man) interfaces.

The WS-Man interface is a DMTF (Distributed Management Task Force) 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.

The size of WS-Man enumeration data file can be large. If spanning across large structures of managed entities, filter dialects are provided to find out certain instances or data that are necessary to the consumer. This whitepaper describes the techniques to filter data from the WS-Man enumeration. The use of filters varies between GET commands. GET command returns a single instance defined by key-binding values (unique identifiers). However, using a filter can return multiple instances of each, matching the entered condition.

2 WS-Man Filters

WS-Man enumeration on a class returns all the instances of that particular class. If a class has children (derived) classes, enumeration on the parent, returns instances of all the children. The return data can be of large size and hence difficult to manage. The filters enable the client to select the necessary instances.

The following filters are supported in the WS-Man stack provided by Dell:

a) Selectors
b) CIM Query Language (CQL) and Windows Management Instrumentation Query Language (WQL)
c) Association Filters

Note: XPATH filters are not supported in Dell WS-Man stack

2.1 Usage of Filter

A basic example of a filter from the Microsoft-provided WinRM application is given here.

```
winrm e "http://schemas.dmtf.org/wbem/wscim/1/cim-
schema/2/DCIM_NICView?__cimnamespace=root/dcim" -u:<user> -p:<password> -
r:https://<ip>/wsman -encoding:utf-8 -a:basic -SkipCNcheck -SkipCAcheck  -
filter:"{SlotLength = "0004" }" -
dialect:http://schemas.dmtf.org/wbem/wsman/1/wsman/SelectorFilter
```

This example is a simple usage of a Selector Filter. The command indicates the required instances which match the specified filter, such as `SlotLength = "0004"`.

2.2 Selector Filter

The selector filter is one of the simplest filter methods for consoles to filter the enumeration data. The selector filter is supported for both Enumeration and Subscribe methods.

A sample SOAP request format is given here.

```
      <s:Header>
         <wsa:To> Service transport address </wsa:To>
         <wsman:ResourceURI> Resource URI </wsman:ResourceURI>
               ...
      </s:Header>
      <s:Body>
         <wsen:Enumerate>
   <wsman:Filter
   Dialect="http://schemas.dmtf.org/wbem/wsman/1/wsman/SelectorFilter">
         <wsman:SelectorSet>
            <wsman:Selector Name="selector-name">
                  selector-value
            </wsman:Selector>
         </wsman:SelectorSet>
      </wsman:Filter>
      ...
      </wsen:Enumerate>
      </s:Body>
```

The highlighted section indicates the SOAP tag to describe a SelectorFilter in the request. Any console aiming to filter their output should present this Filter tag to the WS-Man implementation.

The example included in section 2.1 is a simple usage of the filter. The command without the filter returns all the instances of NICView class that represents the network devices in the server. If the server has four network devices with multiple ports (N) 4 x N instances are returned. If the ports are further partitioned, many instances are returned. To filter a particular instance from a large number of output instances, the

Selector Filter can be used. The selector filter will enable the console to quickly query an instance based on a "Property=Value" expression.

2.2.1 Combining Selectors

The example in section 2.1, indicates the use of one Selector Filter to filter the instances. However, multiple selectors can be provided in one Selector Query string for further filtration. The example here indicates the use of multiple selectors in an enumeration command,

```
winrm e "http://schemas.dmtf.org/wbem/wscim/1/cim-
schema/2/DCIM_NICView?__cimnamespace=root/dcim" -u:<user> -p:<password> -
r:https://<ip>/wsman -encoding:utf-8 -a:basic -SkipCNcheck -SkipCAcheck  -
filter:"{SlotLength = "0004";FunctionNumber = "7" }" -
dialect:http://schemas.dmtf.org/wbem/wsman/1/wsman/SelectorFilter
```

The above example indicates the use of a string selector along with an integer selector. The instances that matches SlotLength="0004" AND FunctionNumber="7" will be returned.

2.2.2 SOAP Request Format

Following is an example of the SOAP request format for using the selector filter as indicated in the example in section 2.2.1 .

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
   xmlns:a="http://schemas.xmlsoap.org/ws/2004/08/addressing"
   xmlns:n="http://schemas.xmlsoap.org/ws/2004/09/enumeration"
   xmlns:w="http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd"
   xmlns:p="http://schemas.microsoft.com/wbem/wsman/1/wsman.xsd"
   xmlns:b="http://schemas.dmtf.org/wbem/wsman/1/cimbinding.xsd">
  <s:Header>
    <a:To>https://<ip>:<port>/wsman</a:To>
    <w:ResourceURI
   s:mustUnderstand="true">http://schemas.dmtf.org/wbem/wscim/1/cim-
   schema/2/DCIM_NICView</w:ResourceURI>
    ...
  </s:Header>
  <s:Body>
    <n:Enumerate>
        ...
        <w:Filter
   Dialect="http://schemas.dmtf.org/wbem/wsman/1/wsman/SelectorFilter">
        <wsman:SelectorSet
   xmlns:wsman='http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd'>
          <wsman:Selector Name='SlotLength'>0004</wsman:Selector>
          <wsman:Selector Name='FunctionNumber'>7</wsman:Selector>
        </wsman:SelectorSet>
      </w:Filter>
    </n:Enumerate>
  </s:Body>
</s:Envelope>
```

2.3 Query Languages (CQL and WQL)

CQL (CIM Query Language) and WQL (Windows Management Instrumentation Query Language) are languages derived from SQL with minor semantic changes. In WS-Man, both Enumeration and subscription can use the query languages to select a particular instances of a class. CQL is a DMTF standard and WQL is a Microsoft's implementation of CQL. Familiarity with the syntax of SQL will ease the use of CQL or WQL.

The query languages can be used as part of any WS-Man enumeration or subscription call.

2.3.1 Enumeration and Query Languages

You can use the CQL in an enumeration as follows.

```
winrm e cimv2/DCIM_NICView -u:<userid> -p:<password> -r:https://<ip>/wsman -
a:basic -encoding:utf-8 -skipCACheck -skipCNCheck -
dialect:http://schemas.dmtf.org/wbem/cql/1/dsp0202.pdf -filter:"select * from
DCIM_NICView WHERE SlotLength='0004”"
```

You can use the WQL in an enumeration as follows,

```
winrm e cimv2/DCIM_MemoryView -u:<userid> -p:<password> -r:https://ip/wsman -
SkipCNcheck -SkipCAcheck -encoding:utf-8 -a:basic -
dialect:http://schemas.microsoft.com/wbem/wsman/1/WQL -filter:"select * from
DCIM_MemoryView WHERE BankLabel='A'"
```

The commands are similar except for the specified dialect (highlighted). The CQL specifies the DMTF specification as the dialect, whereas the WQL uses a different dialect.

The above commands return the same output as the SelectorFilter (described in Section 2.1). However, the query language provides more options to set the conditions. The CQL and WQL can be used interchangeably. Users familiar with WMI will prefer to use WQL whereas CQL is traditional CIM query infrastructure defined as a DMTF specification.

To find out about the possible query strings, refer to the CQL and WQL specification.

For details on CQL, go to

http://www.dmtf.org/sites/default/files/standards/documents/DSP0202_1.0.0.pdf

For details on WQL, go to

http://msdn.microsoft.com/en-us/library/aa394606(v=vs.85).aspx

2.4 Dialects and Associations

2.4.1 What are Associations?

Association classes describe the relation between two CIM classes. An association class has two major properties that link the classes with each other.

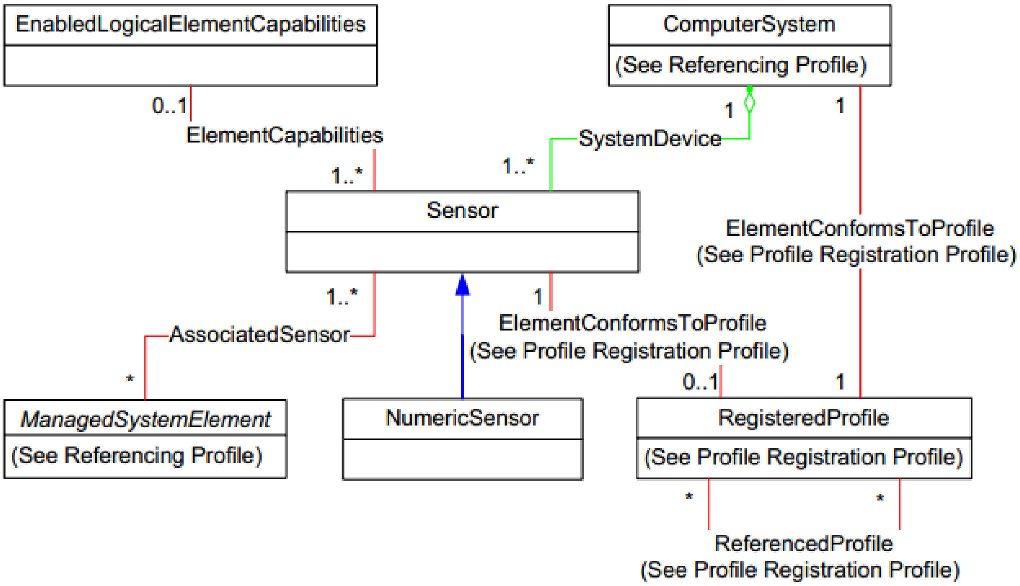
EnabledLogicalElementCapabilities
ComputerSystem
(See Referencing Profile)
0..1
ElementCapabilities
1..*
1
1
SystemDevice
1..*
Sensor
ElementConformsToProfile
(See Profile Registration Profile)
1..*
1
AssociatedSensor
ElementConformsToProfile
(See Profile Registration Profile)
*
0..1
1
ManagedSystemElement
(See Referencing Profile)
NumericSensor
RegisteredProfile
(See Profile Registration Profile)
*
*
ReferencedProfile
(See Profile Registration Profile)

Figure 1: A UML diagram to show Associations

The figure here (DMTF Sensor Profile) indicates the various classes and the association between the classes. The ComputerSystem class is associated with the Sensor class as a SystemDevice. Similarly, fans, power supply units (PSUs), processors, and so on are associated to the ComputerSystem as a SystemDevice.

2.4.2 Associators

Associators operation is used to enumerate CIM Objects (Classes or Instances) that are associated to a particular source CIM Object. The following example indicates how the association classes of a source CIM Object, DCIM_ComputerSystem can be derived using a WinRM command.

```
winrm e http://schemas.dell.com/wbem/wscim/1/cim-schema/2/* -dialect:association
-associations -
filter:{object=DCIM_ComputerSystem?__cimnamespace=root/dcim+CreationClassName=DC
IM_ComputerSystem+Name=srv:system} -u:<user> -p:<pwd> -r:https://<ip>/wsman -
encoding:utf-8 -a:basic -SkipCNcheck –SkipCAcheck
```

This command here provides all the associations (an association is a pair of related or associated EPRs of instance) which have one of the instances matching the filter condition. The value of the filter, is the selector set for an instance of DCIM_ComputerSystem and may be obtained by enumerating the End Point Reference (EPR) of DCIM_ComputerSystem.

The above example will return the EPRs of all the instances which are associated with the ComputerSystem class instance. Output of few instances from multiple instances is given here.

```
DCIM_LCUseOfLog
  Antecedent
    EndpointReference
      Address = http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
      ReferenceParameters
        ResourceURI = http://schemas.dell.com/wbem/wscim/1/cim-
schema/2/DCIM_LCRecordLog
        SelectorSet
          Selector: InstanceID = DCIM:LifeCycleLog
          Selector: __cimnamespace = root/dcim
  Dependent
    EndpointReference
      Address = http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
      ReferenceParameters
        ResourceURI = http://schemas.dell.com/wbem/wscim/1/cim-
schema/2/DCIM_ComputerSystem
        SelectorSet
          Selector: Name = srv:system
          Selector: CreationClassName = DCIM_ComputerSystem
          Selector: __cimnamespace = root/dcim

DCIM_SELUseOfLog
  Antecedent
    EndpointReference
      Address = http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
      ReferenceParameters
        ResourceURI = http://schemas.dell.com/wbem/wscim/1/cim-
schema/2/DCIM_SELRecordLog
        SelectorSet
```

```
          Selector: InstanceID = DCIM:SEL:1
          Selector: __cimnamespace = root/dcim
  Dependent
    EndpointReference
      Address = http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
      ReferenceParameters
        ResourceURI = http://schemas.dell.com/wbem/wscim/1/cim-
schema/2/DCIM_ComputerSystem
        SelectorSet
          Selector: Name = srv:system
          Selector: CreationClassName = DCIM_ComputerSystem
          Selector: __cimnamespace = root/dcim
...
...
```

The corresponding SOAP request packet is given here,

```
   <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>
     ...
     </s:Header>
     <s:Body>
       <wsen:Enumerate>
         ...
         <wsman:Filter
   Dialect="http://schemas.dmtf.org/wbem/wsman/1/cimbinding/associationFilter"
   xmlns:b="http://schemas.dmtf.org/wbem/wsman/1/cimbinding.xsd">
           <b:AssociationInstances>
             <b:Object>
<wsa:Address>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</ws
a:Address>
               <wsa:ReferenceParameters>
                 <wsman:ResourceURI>http://schemas.dell.com/wbem/wscim/1/cim-
   schema/2/DCIM_ComputerSystem</wsman:ResourceURI>
                 <wsman:SelectorSet>
                   <wsman:Selector
   Name="__cimnamespace">root/dcim</wsman:Selector>
                   <wsman:Selector
   Name="CreationClassName">DCIM_ComputerSystem</wsman:Selector>
                   <wsman:Selector Name="Name">srv:system</wsman:Selector>
                 </wsman:SelectorSet>
               </wsa:ReferenceParameters>
             </b:Object>
           </b:AssociationInstances>
```

```
      </wsman:Filter>
    </wsen:Enumerate>
  </s:Body>
</s:Envelope>
```

2.4.3 References

References operation is used to enumerate the association objects that refer to a particular target CIM Object (Class or Instance). It is quite similar to the Associators except for the fact that the associations are not listed, but all of the instances associated to the target CIM class instance is listed.

The WinRM command to find the references is very similar to the command run to find the associations, The '–associations' option must be removed. The following WinRM command indicates the method to enumerate the references of target DCIM_ComputerSystem.

```
winrm e http://schemas.dell.com/wbem/wscim/1/cim-schema/2/* -dialect:association
-
filter:{object=DCIM_ComputerSystem?__cimnamespace=root/dcim+CreationClassName=DC
IM_ComputerSystem+Name=srv:system} -u:<user> -p:<pwd> -r:https://<ip>/wsman -
encoding:utf-8 -a:basic -SkipCNcheck -SkipCAcheck
```

The above example will return all the device instances that are associated with the ComputerSystem class instance. Output of an instance from the multiple instances is given here.

```
DCIM_LCRecordLog
  CurrentNumberOfRecords = 4980
  ElementName = LifeCycle Log
  EnabledState = 2
  HealthState = 5
  InstanceID = DCIM:LifeCycleLog
  LogState = 2
  MaxNumberOfRecords = 0
  OperationalStatus = 2
  OverwritePolicy = 8
  RequestedState = 12

DCIM_SELRecordLog
  CurrentNumberOfRecords = 34
  ElementName = System Event Log
  EnabledState = 2
  HealthState = 5
  InstanceID = DCIM:SEL:1
  LogState = 2
  MaxNumberOfRecords = 512
  OperationalStatus = 2
  OverwritePolicy = 7
  RequestedState = 12
...
...
```

The corresponding SOAP request packet is given here.

```
<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>
  ...
  </s:Header>
  <s:Body>
    <wsen:Enumerate>
    ...
      <wsman:Filter
Dialect="http://schemas.dmtf.org/wbem/wsman/1/cimbinding/associationFilter"
xmlns:b="http://schemas.dmtf.org/wbem/wsman/1/cimbinding.xsd">
        <b:AssociatedInstances>
          <b:Object>
            <wsa:Address>http://schemas.xmlsoap.org/ws/2004/08/addressing/rol
      e/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="__cimnamespace">root/dcim</wsman:Selector>
                <wsman:Selector
Name="CreationClassName">DCIM_ComputerSystem</wsman:Selector>
                <wsman:Selector Name="Name">srv:system</wsman:Selector>
              </wsman:SelectorSet>
            </wsa:ReferenceParameters>
          </b:Object>
        </b:AssociatedInstances>
      </wsman:Filter>
    </wsen:Enumerate>
  </s:Body>
</s:Envelope>
```

3 Filtering on Subscriptions

The Subscribe message allows a client to express interest in receiving events. Filters are used to identify the event or events a client wants to receive. The Subscription can be generic on a class of events or be specific to particular property in the class.

3.1 Subscription Management

WS-Man specification defines a subscription-driven methodology for CIM indications. For any necessary event in iDRAC, the console can subscribe to an event class and redirect the generated events to an event sink address provided in the subscription request. For any subscription request, a similar SOAP request packet is returned.

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:wse="http://schemas.xmlsoap.org/ws/2004/08/eventing"
xmlns:wsman="http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd">
  <s:Header>
    <wsa:Action
s:mustUnderstand="true">http://schemas.xmlsoap.org/ws/2004/08/eventing/Subscribe</w
sa:Action>
    <wsa:To s:mustUnderstand="true"> Service Transport Address </wsa:To>
    <wsman:ResourceURI s:mustUnderstand="true"> Resource URI </wsman:ResourceURI>
    <wsa:MessageID s:mustUnderstand="true">urn:uuid:4a8eeae1-b32c-11e3-b73a-
2cd05a84c9be</wsa:MessageID>
    <wsa:ReplyTo>
      <wsa:Address/>
    </wsa:ReplyTo>
    <wsman:SelectorSet>
      <wsman:Selector Name="__cimnamespace">root/dcim</wsman:Selector>
    </wsman:SelectorSet>
  </s:Header>
  <s:Body>
    <wse:Subscribe>
      <wse:Delivery
Mode="http://schemas.xmlsoap.org/ws/2004/08/eventing/DeliveryModes/Push">
        <wse:NotifyTo>
          <wsa:Address> eventsink url </wsa:Address>
        </wse:NotifyTo>
        <wsman:Heartbeats> heart beat </wsman:Heartbeats>
      </wse:Delivery>
      <wsman:Filter Dialect="http://schemas.microsoft.com/wbem/wsman/1/WQL">SELECT
* FROM DCIM_LifecycleJobAlertIndication</wsman:Filter>
    </wse:Subscribe>
  </s:Body>
</s:Envelope>
```

The above SOAP packet includes the Filter tag which specifies the instances of the indication provider required when an event occurs. The WQL query can be modified to create a subscription for a particular job by adding the clause, `WHERE JOB_ID=<job id>`. The job id is typically a response received from the server after the user creates a job.

3.2 Usage from Different Consoles

3.2.1 Windows PowerShell

WinRM as a tool does not provide support for Ws-events. However, we can subscribe to events from powershell.

A powershell example of subscription to events is given here..

```
Register-CimIndicationEvent -Query "SELECT * FROM
DCIM_LifecycleJobAlertIndication" -Namespace root/dcim  -MessageData "New Event
Has Come" -Action {$Global:myevent = $event } -CimSession $Session -
SourceIdentifier "Test2"
```

The usage of query is the same for powershell.

3.2.2 Linux WS-Man CLI

The Linux WS-Man CLI provides support for WS-Events. The below command provides a subscription request to iDRAC and creates a filter for the instances of the indication class where PerceivedSecurity property is equal to 2. This filtration is possible by using dialects.

```
wsman subscribe 'http://schemas.dmtf.org/wbem/wscim/1/*' -x "SELECT * FROM
DCIM_LifecycleJobAlertIndication WHERE PerceivedSeverity='2'" -D
'http://schemas.dmtf.org/wbem/cql/1/dsp0202.pdf' -Z
'http://10.94.44.69:80/eventsink' -h 10.94.195.40 -P 443 -u root -p calvin -v -j
utf-8 -y basic -R -o -m 256 -N root/dcim -c dummy -V -r 600
```

4 Fragment Transfer

Fragment Transfer may also be an effective way to assist in filtering unnecessary data. In other transfer methods, transfer is done at instance level. However, Fragment Transfer enables to perform fragment or property level transfers while doing GET or ENUMERATION operation. In winrm, ***-fragment:<Property Name>*** should transfer only the property specified in ***<Property Name>*** option.

The following SOAP packet indicates the process to get only the InstanceIDs from an enumeration of LCEnumeration.

```
<?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"> Service Transport Address </wsa:To>
    <wsman:ResourceURI
s:mustUnderstand="true">http://schemas.dell.com/wbem/wscim/1/cim-schema/2/DCIM_LCEnumeration</wsman:ResourceURI>
    <wsa:MessageID s:mustUnderstand="true">urn:uuid:21a0e230-b644-11e3-8ef4-2cd05a84c9be</wsa:MessageID>
    <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">InstanceID</wsman:FragmentTransfer>
  </s:Header>
  <s:Body>
    <wsen:Enumerate>
      ...
    </wsen:Enumerate>
  </s:Body>
</s:Envelope>
```

5 Conclusion

The use of dialects in WS-Man SOAP requests enable the consoles to filter data based on the requirements. WS-Man responses may become large and difficult to manage. If the console requires particular instances matching certain conditions then filter dialect is a useful tool. This can be further improved by using Fragment Transfers which transfer only a particular property of a selected instance.

6 Appendix A

List of abbreviations:

Acronym	Definition
EPR	End Point Reference, a selector set that uniquely identifies an object/instance.
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
LC	Lifecycle Controller