Enabling Web Service with Common Information Model

Explore the Mapping from WS-Management capability to CIM operations

Sun Hao

May 08, 2009

In this article we will introduce the concept of WS-Management and Common Information Model (CIM). By exploring the SOAP message with multiple examples, we will learn how to transfer CIM operations through WS-Management SOAP messages.

To meet the increasing needs of interoperability between multiple computer resources, Common Information Model (CIM) is widely used. It provides a common definition of management information for systems, networks, applications and services, and allows for vendor extensions. WS-Management is a SOAP-based Web Service protocol allowing systems to interact with management information, independently from system brand or platform. WS-Management enables systems to send and receive event and performance information and execute tasks on managed systems.

Currently, CIM operations have been mapped to web service by WS-Management and WS-CIM related specifications defined by DMTF. The standard specifications have just been finalized and some implementation projects are still in the process, e.g. Openwsman. So there is little complete technical document investigating the concrete soap message which illustrates the CIM operation from web to CIMOM. In this article we will introduce CIM and WS-Management to readers and then explore the complete soap message for mapping WS-Management capability to CIM operations. This will benefit readers who are working on this area and will be valuable for those having interest in CIM and Web service.

WBEM and CIM

Web-Based Enterprise Management (WBEM) is a set of management and Internet standard technologies developed by Distributed Management Task Force (DMTF) to unify the management of distributed computing environments. WBEM provides the ability for the industry to deliver a well-integrated set of standard-based management tools, facilitating the exchange of data across otherwise disparate technologies and platforms.
CIM provides a common definition of management information for systems, networks, applications and services, and allows for vendor extensions. CIM's common definitions enable vendors to exchange semantically rich management information between systems throughout the network.

**WS-Management in Web Services**

WS-Management is a SOAP-based protocol allowing systems to interact with management information, independently from system brand or platform. WS-Management enables systems to send and receive event and performance information and execute tasks on managed systems.

WS-Management addresses the cost and complexity of IT management by providing a common way for systems to access and exchange management information across the entire IT infrastructure. By using Web services to manage IT systems, deployments that support WS-Management will enable IT managers to remotely access devices on their networks - everything from silicon components and handheld devices to PCs, servers and large-scale data centers.

**Figure 1. WS-Management specification**

The current version of WS-Management specification, released by DMTF includes several sub-specifications:

- **WS-Addressing**: WS-Management relies on WS-Addressing to define references to other Web service endpoints and to define some of the headers used in SOAP messages.
- **WS-Transfer**: This is used as a basis for simple unary resource access: Get, Put, Delete, and Create.
- **WS-Enumeration**: This is used as a basis for iteration through the members of a collection.
- **WS-Eventing**: WS-Management provides eventing functionality through the WS-Eventing specification.
- **WS-Security**: The security mechanisms that sit in front of any SOAP message processing, limiting the attack surface.

**Relationship between WBEM/CIM and WS-Management**

In brief, WBEM/CIM describes and manages the hardware and software resources as defined in the object model. While on the other hand, WS-Management determines the transport for accessing the resources.
Figure 2. CIM and WS-Management hierarchy

Figure 2 shows the hierarchy of CIM in WS-Management. CIM objects are represented by WS-CIM and described by WS-Management WSDL. Based on HTTP, SOAP messaging transport, WS-Management application can deliver CIM objects through multiple modes, e.g. transfer, enumeration and eventing. Currently, there are already several implementation projects for WS-Management, for example, Openwsman and Wiseman.

Mapping from WS-Management to WBEM/CIM

DMTF specification DSP0227 describes the CIM binding for WS-Management, which describes how transformed CIM resources, as specified by the WS-CIM specification, are bound to WS-Management operations and WSDL definitions.

Figure 3. Mapping from WS-Management to CIM

In the next sections, we will introduce this mapping through examples. During that, more detailed rules and syntax will be illustrated.

CIM operation message flow

In this experiment, the whole system is constructed as Figure 4. To be able to use Web service with Common Information Model, four necessary components are required: CIMOM, WS-
Enabling Web Service with Common Information Model

Management implementation, a Web server and Web pages. CIM user talks with Web pages to access CIM objects in the managed system. The process is illustrated below:

**Figure 4. CIM operation message flow**

![CIM operation message flow diagram](image)

The entire operation process includes these steps:

1. CIM user sends CIM request and gets CIM response on the web pages.
2. Web pages sends request to the Web server through a SOAP message which contains transformed CIM operation and CIM objects.
3. Web server relies on WS-Management implementation to handle CIM related SOAP messages.
4. WS-Management implementation translates the SOAP message to CIM-XML and communicates with CIM object manager (CIMOM), for example, SFCB or OpenPegasus.
5. CIMOM relies on CIM provider to query and access CIM objects in the managed system.

In this article, we are focusing on the SOAP message processed in step 2. We will research the SOAP messages for enumerate-instances, get-instance, association, invoke-method and indication CIM operations including both request and response process. We will use Openwsman as the WS-Management implementation.

**Enumerate Instance operation**

This operation allows the client to enumerate all CIM instances of a specified CIM class or its subclasses.

**Listing 1. Request SOAP message for enumerating instance**

```xml
<?xml version="1.0"?>
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
    xmlns:wsman="http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd"
    xmlns:wsmb="http://schemas.dmtf.org/wbem/wsman/1/cimbinding.xsd">
  <s:Header>
    <wsa:Action s:mustUnderstand="true">
    </wsa:Action>
  </s:Header>
  <s:Body/>
</s:Envelope>
```
Explanation:

- **Wsa:Action** --- identifies which operation is to be carried out against the resource.
- **Wsa:To** --- the transport address of the service
- **Wsman:ResourceURI** --- the URI of the resource class or instance representation.
- **Wsa:MessageID** --- identifies this specific message uniquely for tracking and correlation purpose. The format defined in RFC 4122 is often used
- **Wsa:ReplyTo** --- identifies the response address. In this example, the string represents the response is to be delivered over the same connection with request.
- **Wsen:enumerate** --- identifies the operation is enumeration
- **Wsmb:PolymorphismMode** --- is used as an option to identifies whether the subclasses should be returned.

**Listing 2. Response SOAP message for enumeration instance**

```xml
<?xml version="1.0" encoding="UTF-8"?>
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
  <s:Header>
    <wsa:To>
      http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
    </wsa:To>
    <wsa:Action>
    </wsa:Action>
    <wsa:RelatesTo>
      uuid:c9a366aa-59f5-19f5-8002-7ec0ed251100
    </wsa:RelatesTo>
    <wsa:MessageID>
      uuid:c0b353c8-59f5-19f5-8003-7ec0ed251100
    </wsa:MessageID>
  </s:Header>
  <s:Body>
    <wsen:EnumerateResponse>
      <wsen:EnumerationContext>c9b2ca21-59f5-19f5-8002-7ec0ed251100
    </wsen:EnumerateResponse>
  </s:Body>
</s:Envelope>
```
Explanation:

Enabling Web Service with Common Information Model
• This response includes several soap messages that construct a whole response, the sequence is as follows: enumeration, pull and pull response. Enumeration returns enumeration context for Pull operation and Pull response returns the CIM objects
• Wsa:RelatesTo --- identifies its requested correlated operation message ID
• Wsen:EnumerateResponse --- returns a context object to represent referenced enumeration. This is used for following Pull operation
• Wsen:Pull --- continues an enumeration to retrieve batches of results from the initial enumeration message
• Wsen:PullResponse --- is followed by wsen:Items to represent retrieved CIM objects in the form of class name and property name-value pairs. In this example, Linux_OperatingSystem is the CIM instance's class and its properties are listed as well

Get Instance operation
This operation allows the client to receive a specified CIM instance by supplying its key identities.

Listing 3. Request SOAP message for get instance

```xml
<?xml version="1.0"?>
<wsa:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
 xmlns:wsman="http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd">
  <s:Header>
    <wsa:Action s:mustUnderstand="true">
    </wsa:Action>
    <wsa:To s:mustUnderstand="true">http://localhost:8889/wsman</wsa:To>
    <wsman:ResourceURI s:mustUnderstand="true">
      http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_OperatingSystem
    </wsman:ResourceURI>
    <wsa:MessageID s:mustUnderstand="true">
      uuid:099a84d8-59f6-19f6-8002-7ec0ed251100
    </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="CSName">sles101-hao.site</wsman:Selector>
      <wsman:Selector Name="Name">sles101-hao.site</wsman:Selector>
      <wsman:Selector Name="CSCreationClassName">Linux_ComputerSystem</wsman:Selector>
      <wsman:Selector Name="CreationClassName">Linux_OperatingSystem</wsman:Selector>
    </wsman:SelectorSet>
  </s:Header>
  <s:Body/>
</wsa:Envelope>
```

Explanation:

• Wsman:SelectorSet --- identifies or selects the resource instance to be accessed with certain properties name and value. Only the matched instance will be returned.

Listing 4. Response SOAP message for get instance

```xml
<?xml version="1.0" encoding="UTF-8"?>
<wsa:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
 xmlns:n1="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_OperatingSystem"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <s:Header>
    <wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:To>
  </s:Header>
</wsa:Envelope>
```
<wsa:Action>
</wsa:Action>
<wsa:RelatesTo>uuid:099a84d8-59f6-19f6-8002-7ec0ed251100</wsa:RelatesTo>
<wsa:MessageID>uuid:09aab9b8-59f6-19f6-8006-7ec0ed251100</wsa:MessageID>
</s:Header>
<s:Body>
<n1:CIM_OperatingSystem>
  <n1:CSCreationClassName>Linux_ComputerSystem</n1:CSCreationClassName>
  <n1:CSName>sles101-hao.site</n1:CSName>
  <n1:Caption>Operating System</n1:Caption>
  <n1:CodeSet>ANSI_X3.4-1968</n1:CodeSet>
  ...  
  <n1:CurrentTimeZone>480</n1:CurrentTimeZone>
  <n1:RequestedState>2</n1:RequestedState>
  <n1:SizeStoredInPagingFiles>2104444</n1:SizeStoredInPagingFiles>
  <n1:Status)NULL</n1:Status>
</n1:CIM_OperatingSystem>
</s:Body>
</s:Envelope>

Explanation:

- Wsa:Action --- just notice that the action section is changed according to different CIM operation.

Association operation

This operation allows the client to enumerate all CIM instances that are associated with the source object through some certain CIM association classes.

Listing 5. Request SOAP message for associate instances

```xml
<?xml version="1.0"?>
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsman="http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd"
  xmlns:wsmb="http://schemas.dmtf.org/wbem/wsman/1/cimbinding.xsd">
  <s:Header>
    <wsa:Action s:mustUnderstand="true">
    </wsa:Action>
    <wsa:To s:mustUnderstand="true">http://9.123.253.83:8889/wsman</wsa:To>
    <wsman:ResourceURI s:mustUnderstand="true">
      http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/KVM_HostSystem
    </wsman:ResourceURI>
    <wsa:MessageID s:mustUnderstand="true">
      uuid:4fa307cb-5b8b-5b5-8002-402497251100</wsa:MessageID>
    <wsa:ReplyTo>
      <wsa:Address>
        http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
      </wsa:Address>
    </wsa:ReplyTo>
  </s:Header>
  <s:Body>
    <wsen:Enumerate>
      <wsman:Filter>
        Dialect="http://schemas.dmtf.org/wbem/wsman/1/cimbinding/associationFilter">
          <wsmb:AssociatedInstances>
            <wsmb:Object>
              <wsa:Address>
                http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
              </wsa:Address>
            </wsmb:Object>
          </wsmb:AssociatedInstances>
        </wsman:Filter>
      </wsen:Enumerate>
    </s:Body>
</s:Envelope>
```
Explanation:

- **Wsman:Filter Dialect**--- allows the use of enumeration using existing query languages such as SQL and CQL, which combine predicate and projection information in the same syntax.
- **Wsmb:AssociatedInstances**--- defined in WS-CIM to represent the instances related to source object through an association.
- **Wsa:ReferenceParameters**--- is same as wsman:SelectorSet. These values are used to select an instance if the resource URI identifies a multiple-instantiated target.
- **Wsmb:AssociationClassName**--- defined in WS-CIM to represent the association's class name.
- **Wsmb:ResultClassName**--- identifies the CIM class name of target object.

**Listing 6. Response SOAP message for associate instances**

```xml
<?xml version="1.0" encoding="UTF-8"?>
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
    <s:Header>
        <wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:To>
        <wsa:Action>
        </wsa:Action>
        <wsa:RelatesTo>uuid:4fa307cb-58b5-18b5-8002-402497251100</wsa:RelatesTo>
        <wsa:MessageID>uuid:4fc2b63b-58b5-18b5-8008-402497251100</wsa:MessageID>
    </s:Header>
    <s:Body>
        <wsen:EnumerateResponse>
            <wsen:EnumerationContext>
                4fc1ef2e-58b5-18b5-8007-402497251100
            </wsen:EnumerationContext>
        </wsen:EnumerateResponse>
    </s:Body>
</s:Envelope>
```
<wsa:To s:mustUnderstand="true">http://9.123.253.83:8889/wsman</wsa:To>
<wsman:ResourceURI s:mustUnderstand="true">http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/KVM_HostSystem</wsman:ResourceURI>
<wsa:MessageID s:mustUnderstand="true">uuid:4fc2bb41-58b5-18b5-8003-402497251100</wsa:MessageID>
<wsa:ReplyTo>
</wsa:ReplyTo>
</s:Header>
<s:Body>
<wsen:Pull>
  <wsen:EnumerationContext>4fc1ef2e-58b5-18b5-8007-402497251100</wsen:EnumerationContext>
</wsen:Pull>
</s:Body>
</s:Envelope>

<?xml version="1.0" encoding="UTF-8"?>
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:n1=http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/KVM_ProcessorPool
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <s:Header>
    <wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:To>
    <wsa:Action>
    </wsa:Action>
    <wsa:RelatesTo>uuid:4fc2bb41-58b5-18b5-8003-402497251100</wsa:RelatesTo>
    <wsa:MessageID>uuid:4fc2c806-58b5-18b5-8009-402497251100</wsa:MessageID>
  </s:Header>
  <s:Body>
    <wsen:PullResponse>
      <wsen:Items>
        <n1:KVM_ProcessorPool>
          <n1:AllocationUnits>Processors</n1:AllocationUnits>
          <n1:Capacity>2</n1:Capacity>
          <n1:ResourceSubType xsi:nil="true"/>
          <n1:ResourceType>3</n1:ResourceType>
          <n1:Status xsi:nil="true"/>
        </n1:KVM_ProcessorPool>
      </wsen:Items>
      <wsen:EnumerationContext>4fc1ef2e-58b5-18b5-8007-402497251100</wsen:EnumerationContext>
    </wsen:PullResponse>
  </s:Body>
</s:Envelope>

Explanation:

- Wsen:PullResponse --- the associated instances are returned through enumeration and pull operation.

**Invoke Method operation**

This operation allows the client to invoke the extrinsic method of a CIM instance.
Listing 7. Request SOAP message for invoke method

```xml
<?xml version="1.0"?>
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:wsman="http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd"
xmlns:n1="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_ComputerSystem">
  <s:Header>
    <wsa:Action s:mustUnderstand="true">
      http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_ComputerSystem/mymethod
    </wsa:Action>
    <wsa:To s:mustUnderstand="true">http://9.123.251.94:8889/wsman</wsa:To>
    <wsman:ResourceURI s:mustUnderstand="true">
      http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_ComputerSystem
    </wsman:ResourceURI>
    <wsa:MessageID s:mustUnderstand="true">
      uuid:a9d78e51-5943-1943-8002-7ec0ed251100
    </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="Name">sunhaodemethod</wsman:Selector>
      <wsman:Selector Name="CreationClassName">SUN_method</wsman:Selector>
      <wsman:Selector Name="__cimnamespace">root/cimv2</wsman:Selector>
    </wsman:SelectorSet>
  </s:Header>
  <s:Body>
    <n1:mymethod_INPUT>
      <n1:name>myName</n1:name>
    </n1:mymethod_INPUT>
  </s:Body>
</s:Envelope>
```

Explanation:

- **WsA:Action** --- is changed according to customized method name. The format is that method name appends CIM class name to form a specific action. In this example, we will invoke the method of “mymethod” on an instance of CIM_ComputerSystem.
- **Wsman:SelectorSet** --- identifies the specific instance that will take the responsibility of invoking extrinsic method.
- **Mymethod_INPUT** --- includes all input parameters for the “mymethod” method. The parameter name and value are listed after this tag such as this example for parameter “name”.

---

**Enabling Web Service with Common Information Model**

Page 11 of 14
Listing 8. Response SOAP message for invoke method

```xml
<?xml version="1.0" encoding="UTF-8"?>
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:n1="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_ComputerSystem">
  <s:Header>
    <wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:To>
    <wsa:Action>
      http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_ComputerSystem/mymethodResponse
    </wsa:Action>
    <wsa:RelatesTo>uuid:a9d78e51-5943-1943-8002-7ec0ed251100</wsa:RelatesTo>
    <wsa:MessageID>uuid:a9f66c80-5943-1943-8002-7ec0ed251100</wsa:MessageID>
  </s:Header>
  <s:Body>
    <n1:mymethod_OUTPUT>
      <n1:ReturnValue>called method</n1:ReturnValue>
      <n1:out>this is out parameter</n1:out>
    </n1:mymethod_OUTPUT>
  </s:Body>
</s:Envelope>
```

Explanation:

- **Mymethod_OUTPUT** --- this tag includes return value and output parameters from the response method invocation. Return value is wrapped in the tag of “ReturnValue” and every output parameter is wrapped separately as name-value pairs.

Subscribe Indication operation

This operation allows the client to subscribe to certain event notifications in the managed system. To be more detailed, CIM users register event notifications by creating necessary CIM instances in the managed system and waits the for expected CIM indication instances.

Listing 9. Request SOAP message for subscribe indications

```xml
<?xml version="1.0"?>
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:wsman="http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd"
  <s:Header>
    <wsa:Action s:mustUnderstand="true">
    </wsa:Action>
    <wsa:To s:mustUnderstand="true">http://localhost:8889/wsman</wsa:To>
    <wsman:ResourceURI s:mustUnderstand="true">
      http://schemas.dmtf.org/wbem/wscim/1/
    </wsman:ResourceURI>
    <wsa:MessageID s:mustUnderstand="true">
      uuid:e48c7154-59d2-19d2-8002-7ec0ed251100
    </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/interop</wsman:Selector>
    </wsman:SelectorSet>
  </s:Header>
</s:Envelope>
```
Explanation:

• Wsa:Action --- for subscribing indication operation, the action tag should be “/eventing/Subscribe”.
• Wsman:ResourceURI --- this is different from other SOAP messages that it does give a specified CIM class name but a “*” which means all available CIM classes under that domain.
• Name= “_cimnamespace” --- represents the CIM namespace from where the event notification comes.
• Wse:Subscribe --- includes the information about the delivery mode, notification receiver and filters, etc.
• Wse:Delivery Mode --- contains Push, Pull and PushWithAck, represent different way for delivering notification to subscriber.
• Wse:NotifyTo --- represents the subscriber.
• Wsman:Filter --- this example uses WQL as its dialect to select all indications of the CIM_ProcessIndication class.

Listing 10. Response SOAP message for subscribe indications

```xml
<?xml version="1.0" encoding="UTF-8"?>
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
    <s:Header>
        <wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:To>
        <wsa:Action>
        </wsa:Action>
        <wsa:RelatesTo>uuid:e48c7154-59d2-19d2-8002-7ec0ed251100</wsa:RelatesTo>
        <wsa:MessageID>uuid:e49d431d-59d2-19d2-8004-7ec0ed251100</wsa:MessageID>
    </s:Header>
    <s:Body>
        <wse:SubscribeResponse>
            <wse:SubscriptionManager>
                <wsa:Address>http://localhost:8889/wsman</wsa:Address>
                <wse:Identifier>uuid:e49bdf13-59d2-19d2-8003-7ec0ed251100</wse:Identifier>
            </wse:SubscriptionManager>
            <wse:Expires>PT120.000000S</wse:Expires>
            <wsen:EnumerationContext>uuid:e49bdf13-59d2-19d2-8003-7ec0ed251100</wsen:EnumerationContext>
        </wse:SubscribeResponse>
    </s:Body>
</s:Envelope>
```
Explanation:

- Wse:SubscriptionManager --- returns any service-specific reference parameters for CIM client to use when issuing other event requests.
- The enumeration context can be used to retrieve the indication instance through Pull operation.

**Conclusion**

In this article we learned the concept of WS-Management and CIM. Through binding WS-Management and CIM, we can enable the Web service with the capability of Common Information Model which will bring great benefits in many aspects. The SOAP messages explained in this article show the details for successfully mapping CIM operations and WS-Management operations.

© Copyright IBM Corporation 2009

**Trademarks**
(www.ibm.com/developerworks/ibm/trademarks/)