Document Number: DSP0227

Date: 2009-06-19

Version: 1.0.0

WS-Management CIM Binding Specification

Document Type: Specification

Document Status: DMTF Standard

Document Language: E

Contents

Tables

Foreword

The *WS-Management CIM Binding Specification* (DSP0227) was prepared by the DMTF WS-Management Working Group.

Acknowledgments

The authors wish to acknowledge the following people:

Editors:

- Richard Landau – Dell Inc.
- Hemal Shah – Broadcom Corporation
- Steve Hand – Symantec Corp.

Contributors:

- Josh Cohen – Microsoft Corporation (Chair)
- Jim Davis – WBEM Solutions
- David Hines – Intel
- Bryan Murray – Hewlett-Packard
- Brian Reistad – Microsoft Corporation

Introduction

This document describes the CIM binding for WS-Management. It describes how transformed CIM resources, as specified by the *WS-CIM Mapping Specification*, are bound to WS-Management operations and WSDL definitions.

WS-Management CIM Binding Specification

1 Scope

This clause describes the scope of this specification, including some items that are specifically out of scope.

1.1 In-Scope

This specification describes how to use the Web Services for Management (WS-Management) protocol to communicate with resources modeled with CIM and exposed through the XML schema mapping described by WS-CIM.

1.2 Out of Scope

This specification does not describe how to expose the WBEM intrinsic methods that perform schema manipulation of CIM classes (for example, CreateClass) using the WS-Management protocol.

This specification does not describe how to generate the XML schema for a CIM class.

1.3 Conformance

This specification supplements the *WS-Management Specification*. When this specification is supported, requests using a particular version of WS-Management are assumed to use the same version of this specification; both specifications will be updated concurrently. (The version of this specification cannot generally be directly determined from a SOAP message because most requests do not contain any elements from this specification or the XML namespace of this specification.)

An implementation is not conformant with this specification if it fails to satisfy one or more of the requirements defined in the conformance rules for each clause, as indicated by the following format:

R*nnnn*: Rule text

2 Normative References

The following reference documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

2.1 Approved References

DMTF DSP0004, *CIM Infrastructure Specification 2.3,*
http://www.dmtf.org/standards/published_documents/DSP0004_2.3.pdf

DMTF DSP0200, *Specification for CIM Operations over HTTP 1.3,*
http://www.dmtf.org/standards/published_documents/DSP0200_1.3.pdf

DMTF DSP0226, *WS-Management Specification, 1.0*,
http://www.dmtf.org/standards/published_documents/DSP0226_1.0.pdf

DMTF DSP0230, *WS-CIM Mapping Specification, 1.0*,
http://www.dmtf.org/standards/published_documents/DSP0230_1.0.pdf

DMTF DSP8016, *WBEM Operations Message Registry 1.0*, http://schemas.dmtf.org/wbem/messageregistry/1/DSP8016_1.0.xml

IETF RFC3986, *Uniform Resource Identifier (URI) Generic Syntax, January 2005,* http://www.ietf.org/rfc/rfc3986.txt

IETF RFC4646, *Tags for Identifying Languages, September 2006,* http://www.ietf.org/rfc/rfc4646.txt

WC3, *Namespaces in XML, W3C Recommendations, 14 January 1999,* http://www.w3.org/TR/1999/REC-xml-names-19990114

W3C, *SOAP Version 1.2 Part 1: Messaging Framework (Second Edition) SOAP*, *1.2 , W3C Recommendation, 27 April 2007,* http://www.w3.org/TR/soap12-part1/

WC3, *Web Services Description Language (WSDL)*, *1.1, W3C Note, 15 March 2001,* http://www.w3.org/TR/wsdl

W3C, *Web Services Addressing (WS-Addressing), W3C Member Submission, 10 August 2004,* http://www.w3.org/Submission/ws-addressing/

W3C, *Web Services Enumeration (WS-Enumeration), W3C Member Submission, 15 March 2006,* http://www.w3.org/Submission/WS-Enumeration/

W3C, *Web Services Eventing (WS-Eventing), W3C Member Submission 15 March 2006,* http://www.w3.org/Submission/WS-Eventing/

W3C, *Web Services Transfer (WS-Transfer), W3C Member Submission, 27 September 2006,* http://www.w3.org/Submission/WS-Transfer/

WC3, *XML Path Language (XPath) Version 1.0, W3C Recommendation, 16 November 1999,* http://www.w3.org/TR/1999/REC-xpath-19991116

WC3, *XML Schema Part 1: Structures Second Edition, W3C Recommendation, 28 October 2004,* http://www.w3.org/TR/xmlschema-1/

WC3, *XML Schema Part 2: Datatypes Second Edition, W3C Recommendation, 28 October 2004,* http://www.w3.org/TR/xmlschema-2/

3 Terms and Definitions

The terms used in DSP0226 and DSP0230 also apply to this specification.

3.1
can
used for statements of possibility and capability, whether material, physical, or causal

3.2
cannot
used for statements of possibility and capability, whether material, physical or causal

3.3
conditional
indicates requirements to be followed strictly in order to conform to the document when the specified conditions are met

3.4
mandatory
indicates requirements to be followed strictly in order to conform to the document and from which no deviation is permitted

3.5
may
indicates a course of action permissible within the limits of the document

3.6
need not
indicates a course of action permissible within the limits of the document

3.7
optional
indicates a course of action permissible within the limits of the document

3.8
shall
indicates requirements to be followed strictly in order to conform to the document and from which no deviation is permitted

3.9
shall not
indicates requirements to be followed strictly in order to conform to the document and from which no deviation is permitted

3.10
should
indicates that among several possibilities, one is recommended as particularly suitable, without mentioning or excluding others, or that a certain course of action is preferred but not necessarily required

3.11
should not
indicates that a certain possibility or course of action is deprecated but not prohibited

3.12
unspecified
indicates that this profile does not define any constraints for the referenced CIM element or operation

3.13
base class
A class that is defined in a CIM schema and from which other classes are derived which may contain other properties or other CIM named elements. These additional named elements are extensions to the base class.

4 Symbols and Abbreviated Terms

4.1
CQL
CIM Query Language

4.2
EPR
Endpoint Reference

4.3
GED
Global Element Definition

4.4
URI
Uniform Resource Identifier

4.5
WBEM
Web-Based Enterprise Management

4.6
WSDL
Web Services Description Language

4.7
XSD
XML Schema Definition

5 Prefixes and XML Namespaces

Table 1 lists namespaces that are used in this specification. The choice of any namespace prefix is arbitrary and not semantically significant.

Table 1 – Prefixes and XML Namespaces

Prefix	XML Namespace	Reference
wsmb	http://schemas.dmtf.org/wbem/wsman/1/cimbinding.xsd	This specification
wsman	http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd	WS-Management
cim	http://schemas.dmtf.org/wbem/wscim/1/common	WS-CIM
s	http://www.w3.org/2003/05/soap-envelope	SOAP 1.2
xs	http://www.w3.org/2001/XMLSchema	XML Schema
wsdl	http://schemas.xmlsoap.org/wsdl	WSDL 1.1
wsa	http://schemas.xmlsoap.org/ws/2004/08/addressing	WS-Addressing
wsen	http://schemas.xmlsoap.org/ws/2004/09/enumeration	WS-Enumeration
wxf	http://schemas.xmlsoap.org/ws/2004/09/transfer	WS-Transfer
wse	http://schemas.xmlsoap.org/ws/2004/08/eventing	WS-Eventing

6 WS-Management Default Addressing Model

WS-Management defines a default addressing model based on WS-Addressing. This clause describes how CIM objects are addressed when they are accessed with the protocol.

WS-Management makes use of WS-Addressing to identify and access resources. WS-Management defines a reference format using the WS-Addressing EndpointReference element, making use of the ReferenceParameter field to contain specific elements (ResourceURI and SelectorSet) to aid in identifying the desired object or objects.

R6-1: Services that support the default addressing model defined by WS-Management are required to conform to this clause and its subclauses.

6.1 Class-Specific ResourceURI

For standard CIM classes, the ResourceURI is identical to the XML namespace URI of the schema for the class. This ResourceURI targets the named class and any derived classes depending on the role of polymorphism.

R6.1-1: Instances of a specific class shall be addressed using a ResourceURI that identifies a specific class.

EXAMPLE: The following ResourceURI is used to reference the CIM_SoftwareElement class in version 2 of the CIM schema.

```
http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_SoftwareElement
```

Note that the XML schema namespace for the instances never changes to reflect CIM namespace usage; only the ResourceURI changes. Class definitions are pure schema; they are independent of their scope or CIM namespace residence. See 6.3 for a description of classes that reside in explicit namespaces.

R6.1-2: It is recommended that vendor-defined classes use the same value for ResourceURI that is used for the XML namespace of the class. The vendor-defined XML namespace should include some form of version field in the namespace URI that can be changed when backward-incompatible changes are made to the XML schema.

Resources without keys are referenced by a class-specific ResourceURI within the SOAP binding, as follows:

```
<s:Envelope ...>
  <s:Header>
    <wsa:To> network address </wsa:To>
    <wsman:ResourceURI> URI of the item </wsman:ResourceURI>
  </s:Header>
```

R6.1-3: If keys are required to discriminate among instances, the WS-Management SelectorSet SOAP header shall be used, as follows:

```
<s:Envelope ...>
  <s:Header>
    <wsa:To> network address </wsa:To>
    <wsman:ResourceURI> URI of the item </wsman:ResourceURI>
      <wsman:SelectorSet>
        <wsman:Selector Name="KeyName"> Key Value </wsman:Selector>
      </wsman:SelectorSet>
      ...
    </s:Header>
```

In this case, the key values required by CIM become individual Selector values. The name of the key is repeated in the Name attribute, and the key value becomes the value of the Selector element. Note that all CIM instances except indications have keys.

EXAMPLE: Example class definition:

```
class CIM_SoftwareElement : CIM_LogicalElement
{
  [key] string Name;
  [key] string Version;
  [key] uint16 SoftwareElementState;
  [key] string SoftwareElementID;
  [key] uint16 TargetOperatingSystem;
  string OtherTargetOS;
  string Manufacturer;
  string BuildNumber;
  string SerialNumber;
  string CodeSet;
  string IdentificationCode;
  string LanguageEdition;
};
```

R6.1-4: The ResourceURI shall be the XML namespace for the class, and the zero or more Selectors shall contain keys defined by this class. A service may process a request with a subset of the keys if the subset uniquely identifies the instance. Clients are guaranteed correct behavior if they supply all keys in the request. Clients might encounter different behavior at different resources if they do not supply all keys.

EXAMPLE: The following example illustrates how to form a EPR using the class definition above:

```
<s:Header>
  <wsa:To> network address </wsa:To>
  <wsman:ResourceURI>
    http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_SoftwareElement
  </wsman:ResourceURI>
  <wsman:SelectorSet>
    <wsman:Selector Name="Name"> AcmeCAD </wsman:Selector>
    <wsman:Selector Name="Version"> 1.2 </wsman:Selector>
    <wsman:Selector Name="SoftwareElementState"> 1 </wsman:Selector>
      <wsman:Selector Name="SoftwareElementID"> 123F00 </ wsman:Selector>
      <wsman:Selector Name="TargetOperatingSystem"> 12 </wsman:Selector>
    </wsman:SelectorSet>
    ...
  </s:Header>
```

R6.1-5: A service shall accept a properly-formed endpoint reference that specifies a class-specific ResourceURI and keys, if necessary, as defined in this clause.

6.2 "All Classes" ResourceURI

Because certain types of queries may cross class boundaries, the class-specific ResourceURI defined in 6.1 is not always applicable.

R6.2-1: Services supporting cross-class queries shall accept an "all classes" ResourceURI.

This ResourceURI effectively targets the query processor in the CIM Server itself and can be used to return both CIM and vendor classes.

The “all classes” ResourceURI is of the same form as the class-specific ResourceURI in which the schema version and class name are replaced with the star character. The presence of the WS-CIM version in this ResourceURI allows the client to indicate which version of the *WS-CIM Mapping Specification* should be used in the translation of the CIM instances to XML.

For example, the following ResourceURI refers to all classes in the CIM namespace represented using version 1 of WS-CIM:

```
http://schemas.dmtf.org/wbem/wscim/1/*
```

When using the class-specific ResourceURI, the results of the enumeration may contain instances of the class identified in the ResourceURI or any derived class. However, the class name is typically repeated in both the ResourceURI and the filter expression.

The advantage to the “all classes” construct is that a single URI may be used for all resource queries and the class information appears in only one place: the filter expression. When the "all classes" construct is used in an Enumerate request, the results returned contain instances from a single CIM namespace, with one important exception: a query using an associationFilter filter dialect such as AssociatedInstances may return instances from more than one CIM namespace.

6.3 Accounting for Different CIM Namespaces

The following special Selector Name is defined to indicate the CIM namespace of the resource or resources for which the message is intended:

```
<wsman:Selector Name="__cimnamespace">xs:string</wsman:Selector>
```

This selector is in addition to any other selectors for CIM keys and is unlikely to collide with others because most CIM keys do not start with two underscore (__) characters.

The absence of this Selector Name in a message indicates that the intended resources are in the default CIM namespace for that service. This specification does not define what the default CIM namespace should be.

R6.3-1: A service offering more than one CIM namespace should include the __cimnamespace Selector Name in an EPR returned in a response message to identify the CIM namespace of an instance in the response. New implementations are strongly encouraged to include the __cimnamespace selector; alternate methods of conveying the CIM Namespace may be deprecated in the future.

R6.3-2: A service shall not fault if the __cimnamespace Selector Name is absent and instead shall utilize the default CIM namespace.

R6.3-3: A service offering more than one CIM namespace should indicate in metadata which CIM namespace is the default. This specification does not define the location or format of such metadata.

R6.3-4: A service supporting more than one CIM namespace shall fault a request that specifies a namespace whose name is not one of the names of the CIM namespaces supported.

R6.3-5: If a service supports exactly one namespace, then

a) the service shall fault a request that includes a __cimnamespace selector that does not match the name of the single namespace; and

b) the service should include the __cimnamespace selector in an EPR returned in a response message to identify the CIM namespace of an instance in the response.

In all cases, R6.3-2 applies: a request with no __cimnamespace selector utilizes the default namespace. If a service supports only one namespace, then that namespace is the default.

7 Accessing Instances

When retrieving and updating an instance of a class, the WS-Transfer Get, Put, and Delete operations from WS-Transfer are used. When creating an instance of a class, the WS-Transfer create operation is used. The fragment access SOAP header defined by WS-Management may be applied to these operations.

Class inheritance also affects how WS-Transfer operations are specified. In many cases vendors have derived a vendor-specific class from the CIM class that allows multiple vendors to implement the same class in the same CIM namespace even if they have not added any additional properties. For example, an implementation may choose to instantiate Vendor_ComputerSystem, which is derived from CIM_ComputerSystem. In many cases, a client must access instances of the derived class, but has only the name of the base class. To access an instance of such a derived class, or obtain an EPR for such an instance that can be used in WS-Transfer operations, a client generally will enumerate instances using the base class. The returned instances or EPRs can optionally contain the correct derived classname. See section 9.3 (“Polymorphism”) for details.

The XML Schema representation of CIM instances permits the omission of non-key and non-required properties in their corresponding XML instance documents. The *WS-CIM Mapping Specification* (DSP0230) defines runtime rules for the Get, Delete, and Create operations.

R7-1: A service should return a wsa:ActionNotSupported fault if the “all classes” ResourceURI is used with any of the WS-Transfer operations, even if this ResourceURI is supported for enumerations or eventing.

7.1 Get

The following clause defines requirements and presents examples related to getting instances.

R7.1-1: A service supporting the Get operation and using the WS-Management Default Addressing Model shall support access using the class-specific ResourceURI that corresponds to the creation class and the selectors of the given instance.

R7.1-2: The response representation shall use the XML Schema identified by the class in the ResourceURI.

7.2 Put

The following clause defines requirements and presents examples related to putting or modifying instances.

R7.2-1: A service supporting the Put operation and using the WS-Management Default Addressing Model shall support access using the class-specific ResourceURI that corresponds to the creation class and the selectors of the given instance.

R7.2-2: A service supporting the Put operation shall accept instance representations that have omitted schema-optional. Any elements not included in the transfer operation shall be left unchanged. A service supporting fragment-level put operations shall also observe this behavior.

R7.2-3: The request and response representations shall use the XML Schema identified by the class in the ResourceURI.

7.3 Delete

The following clause defines requirements and presents examples related to deleting instances.

R7.3-1: A service supporting the Delete operation and using the WS-Management Default Addressing Model shall support access using the class-specific ResourceURI that corresponds to the creation class and the selectors of the given instance.

7.4 Create

The Create operation is different from the other WS-Transfer operations because it is sent to a resource factory rather than to a resource. For CIM, the class-specific ResourceURI is the factory resource that can be used to create instances of the class.

R7.4-1: A service supporting the Create operation and using the WS-Management Default Addressing Model shall support access using the class-specific ResourceURI corresponding to the creation class and, if warranted, the __cimnamespace Selector Name.

However, the fragment-level Create operation operates on the resource itself, so it behaves in the same fashion as the Put operation:

R7.4-2: A service may support the fragment-level Create operation using the class-specific ResourceURI that corresponds to the creation class and the selectors of the given instance.

R7.4-3: A service supporting the Create operation shall accept instance representations that have omitted schema-optional properties and shall interpret such omissions as a request to create the object with the corresponding omitted properties set to a value of NULL. A service supporting the fragment-level Create operation shall also observe this behavior.

8 Filter Dialects

Both WS-Enumeration and WS-Eventing define XPath Version 1.0 as the default filter language (called a "dialect" in those specifications), though other filter languages are accommodated. This specification defines two additional dialects for use with resources modeled using CIM. Services may support these and other query languages by accepting messages with appropriate dialect URIs.

The filter dialects defined in this clause are intended for use with WS-Enumeration and WS-Eventing and not with Fragment-level WS-Transfer.

8.1 CQL

CQL is a SQL-based query language that includes the class name as part of the query. The dialect filter URI for this language is as follows:

http://schemas.dmtf.org/wbem/cql/1/dsp0202.pdf

R8.1-1: Services that accept CQL statements of the form “select * from …” shall return each instance representation using the GED defined for the object’s class within the wsen:Items element.

R8.1-2: Services that accept CQL statements of the form “select a,b,c from …” (a query with projection) shall return each instance representation using the wsman:XmlFragment element. Within the wsman:XmlFragment element, the service shall return property values named in the select statement using either an element with the given label if the AS keyword is used or the property’s GED defined in the *WS-CIM Mapping Specification* if the select-list entry is a property (ignoring any chain or property-scope). Expressions and literals without AS keywords are not valid CQL expressions.

Clients should use wsman:Filter, as opposed to wsen:Filter or wse:Filter, when using CQL statements of the form “select a,b,c from …” because these queries contain projections and are not Boolean predicates.

R8.1-3: Services supporting CQL statements of the form “select a,b,c from …” may return results in any order. To provide clients a mechanism to correlate results with the CQL expression, services should include the attribute wsmb:Expression for all selected-entry elements, and shall include the attribute wsmb:Expression for any selected-entry that would have a duplicate name with another selected-entry. The value of the wsmb:Expression attribute on the element shall be the selected-entry in the select-list from which the element resulted.

EXAMPLE 1: If the select-list of a CQL statement is “ID, Foo.Name, Bar::Host, A AS B, X * Y AS Z”, the query returns the associated elements in the following fragment:

```
(1)  <wsen:Items xmlns:ex='...'>
(2)    <wsman:XmlFragment>
(3)      <ex:ID>...</ex:ID>
(4)      <ex:Name>...</ex:Name>
(5)      <ex:Host>...</ex:Host>
(6)      <B>…</B>
(7)      <Z>…</Z>
(8)    </wsman:XmlFragment>
(9)  </wsen:Items>
```

NOTE 1: The elements that result from the AS keyword do not have an XML namespace.

NOTE 2: Because the response elements are wrapped in the XmlFragment element, which is defined to turn off validation for the entire content of the XmlFragment, it is permissible for the service not to include namespace prefixes for the enclosed elements.

If a join were used with the same named property included from both classes, then the wsmb:Expression would be used to differentiate between them.

EXAMPLE 2: Given a select-list of “CIM_Foo,ID, CIM_Foo.Name, CIM_Bar.Name” the associated elements would be as follows:

```
(1)  <wsen:Items xmlns:bar='...' xmlns:foo='...'>
(2)    <wsman:XmlFragment>
(3)      <foo:ID>...</foo:ID>
(4)      <bar:Name wsmb:Expression='CIM_Bar.Name'> ...</bar:Name>
(5)      <foo:Name wsmb:Expression='CIM_Foo.Name'> ...</foo:Name>
(6)    </wsman:XmlFragment>
(7)  </wsen:Items>
```

R8.1-4: If a service supports wsman:EnumerationMode=EnumerateObjectAndEPR for enumerating instances and endpoint references, then it shall compose the instance representation of the results of the CQL query (as specified in the previous two rules) with the EPR. The CQL query selects the instances and properties of the instance to be returned but has no effect on the EPR that refers to objects that match the where clause of the CQL query.

R8.1-5: If a service supports wsman:EnumerationMode=EnumerateEPR for enumerating endpoint references, then it shall return the EPRs for instances that match the where clause of the CQL query and ignore any properties specified in the select portion of the CQL query.

R8.1-6: If a service uses the WS-Management Default Addressing model, then it should support this filter dialect for Enumerate operations. If the CQL dialect is not supported by the addressed endpoint service, the service shall respond with a wsen:FilterDialectRequestedUnavailable fault.

R8.1-7: If a service uses the WS-Management Default Addressing model and supports the CQL dialect for Enumerate operations it shall support addressing the CIM Server (through the “all classes” ResourceURI) and it should support addressing instances of a class (through the class-specific ResourceURI). If the CQL query references in the FROM clause more than one CIM class, then the Enumerate operation shall be addressed to the "all classes" ResourceURI. If the addressed endpoint and the query contradict each other (for example, the CIM classname in the class-specific ResourceURI does not match the CIM classname in the CQL FROM clause), the service shall respond with a wsen:CannotProcessFilter fault.

R8.1-8: If a service uses the WS-Management Default Addressing model it should support this filter dialect for Subscribe operations. If the CQL dialect is not supported by the addressed endpoint service, the service shall respond with a wsen:FilterDialectRequestedUnavailable fault.

R8.1-9: If a service uses the WS-Management Default Addressing model and supports the CQL dialect for Subscribe operations it shall support addressing the CIM Server (through the "all classes" ResourceURI) and it should support addressing instances of a class (through the class-specific ResourceURI). If the addressed endpoint and the query contradict each other (for example, the CIM classname in the class-specific ResourceURI does not match the CIM classname in the CQL FROM clause), the service shall respond with a wse:EventSourceUnableToProcess fault.

R8.1-10: Services that accept CQL queries should return instances of the most-derived class rather than a requested class, even though the query names a specific class.

EXAMPLE 3: The following request issues a CQL query in which the returned results include properties from the selected instances. This example uses the WS-Management Default Addressing Model but applies to any EPR model used by the service.

```
<s:Envelope>
  <s:Header>
    <wsman:ResourceURI>
      http://schemas.dmtf.org/wbem/wscim/1/*
    </wsman:ResourceURI>
  </s:Header>
  <s:Body>
    <wsen:Enumerate>
      <wsman:Filter Dialect="http://schemas.dmtf.org/wbem/cql/1/dsp0202.pdf">
        SELECT Name, PrimaryOwnerName
        FROM CIM_ComputerSystem
        WHERE EnabledState = 3
      </wsman:Filter>
    </wsen:Enumerate>
  </s:Body>
</s:Envelope>
```

The results include the two requested properties for instances that are "Disabled":

```
<s:Body>
  <wsen:PullResponse>
    <wsen:EnumerationContext> ... </wsen:EnumerationContext>
    <wsen:Items>
      <wsman:XmlFragment>
        <Name>system1</Name>
        <PrimaryOwnerName>Joe</PrimaryOwnerName>
      </wsman:XmlFragment>
      <wsman:XmlFragment>
        <Name>system2</Name>
        <PrimaryOwnerName>Mary</PrimaryOwnerName>
      </wsman:XmlFragment>
      ... etc.
    </wsen:Items>
  </wsen:PullResponse>
</s:Body>
```

8.2 Association Queries

CIM uses associations to relate instances of different classes and defines intrinsic operations to find related classes. Association queries start with one instance that participates in the association (called the source object) and finds all related instances (called the result objects) linked through associations in which a reference to the source object appears as the value of a specific property (called the role) in the association. The query can be further constrained by limiting the roles that are used for the source or result objects as well as limiting the type of the association and result classes. Alternatively, it is possible to issue a query for instances of the associations themselves using a similar set of constraining parameters.

This specification defines the following dialect filter URI for association queries:

http://schemas.dmtf.org/wbem/wsman/1/cimbinding/associationFilter

The following rules apply only to services that support association queries:

R8.2-1: If a service uses the WS-Management Default Addressing model it should support the association filter dialect for Enumerate operations that are addressed to the “all classes” ResourceURI. If such a service receives an Enumerate request addressed to a class-specific Resource URI specifying this filter dialect, the service shall respond with a wsen:FilterDialectRequestedUnavailable fault.

R8.2-2: If a service supports wsman:EnumerationMode=EnumerateObjectAndEPR for enumerating endpoint references, then it shall compose the instance representation of the results of the association query with the EPR as directed. The association query selects the instances and properties of the instance to be returned but has no effect on the presence or absence of the EPR.

R8.2-3: The service should return a wse:FilteringRequestedUnavailable fault in response to Subscribe requests using the association filter dialect.

R8.2-4: If the result of an association query includes no instances, the service shall not return a fault.

8.2.1 Associated Instances

For queries that return associated instances, the Enumerate message has the following form:

```
<wsen:Enumerate>
  <wsman:Filter
    Dialect="http://schemas.dmtf.org/wbem/wsman/1/cimbinding/associationFilter">
      <wsmb:AssociatedInstances>
        <wsmb:Object> xs:any </wsmb:Object>
        <wsmb:AssociationClassName> xs:NCName </wsmb:AssociationClassName> ?
        <wsmb:Role> xs:NCName </wsmb:Role> ?
        <wsmb:ResultClassName> xs:NCName </wsmb:ResultClassName> ?
        <wsmb:ResultRole> xs:NCName </wsmb:ResultRole> ?
          <wsmb:IncludeResultProperty> xs:NCName </wsmb:IncludeResultProperty> *
        </wsmb:AssociatedInstances>
    </wsman:Filter>
  </wsen:Enumerate>
```

The following definitions provide additional, normative constraints on the preceding outline:

- wsen:Enumerate/wsman:Filter/wsmb:AssociatedInstances

  The results include instances related to the source object through an association.

  R8.2.1-1: The results of the enumeration shall be instances associated with the object through an association instance subject to the additional constraints listed in this clause.

- wsen:Enumerate/wsman:Filter/wsmb:AssociatedInstances/wsmb:Object

  Identifies the source object for the association query and is required.

R8.2.1-2: The results shall be associated with the object identified by the endpoint reference in wsmb:Object.

R8.2.1-3: If the EPR to which the Enumerate message is sent and the EPR of the source object reference two different CIM namespaces, the service may respond with a wsen:CannotProcessFilter fault.

R8.2.1-4: If the EPR of the source object does not reference exactly one valid CIM instance, the service shall respond with a wsen:CannotProcessFilter fault. Services should include a textual description of the problem.

- wsen:Enumerate/wsman:Filter/wsmb:AssociatedInstances/wsmb:AssociationClassName

  Represents the name of a CIM association class. This element or parameter is optional.

R8.2.1-5: If the AssociationClassName is present, the results shall include only the instances related to the source object through associations that are instances of only the named class or derived classes. If the AssociationClassName is absent, results shall include instances that are related to the source object through associations of any type.

- wsen:Enumerate/wsman:Filter/wsmb:AssociatedInstances/wsmb:Role

  Represents the name of a reference property of a CIM association class. This element or parameter is optional.

R8.2.1-6: If the Role name is present, the results shall include only instances related to the source object through an association in which the source object plays the specified role (that is, the name of the property in the association class that refers to the source object shall match the value of this parameter). If the Role name is absent, the results shall include instances associated to the source regardless of the role of the source object in the association.

- wsen:Enumerate/wsman:Filter/wsmb:AssociatedInstances/wsmb:ResultClassName

  Represents the name of a CIM class. This element or parameters is optional.

R8.2.1-7: If the ResultClassName is present, the results shall include only objects that are instances of the named class or any of its derived classes. If the ResultClassName is absent, the results shall include all objects regardless of type.

- wsen:Enumerate/wsman:Filter/wsmb:AssociatedInstances/wsmb:ResultRole

  Represents the name of a reference property of a CIM association class. This element or parameter is optional.

R8.2.1-8: If ResultRole name is present, the results shall only include instances related to the source object via an association in which the returned object plays the specified role. In other words, the name of the property in the association class that refers to the returned object shall match the value of this parameter.

- wsen:Enumerate/wsman:Filter/wsmb:AssociatedInstances/wsmb:IncludeResultProperty

  Represents the name of one or more properties of a CIM class. This element or parameter is optional.

R8.2.1-9: If the query does not include an IncludeResultProperty element, the service shall return each instance representation using the GED defined for the object's class within the wsen:Items element.

R8.2.1-10: If the query includes one or more IncludeResultProperty elements, the service shall return each instance representation using the wsman:XmlFragment element. Within the wsman:XmlFragment element, the service shall return property values using the property GEDs defined in the *WS-CIM Mapping Specification*. If the query includes one or more IncludeResultProperty elements, the service shall not return any IncludeResultProperty elements not specified. The service shall ignore any IncludeResultProperty elements that describe properties not defined by the target class. If the service does not support fragment-level access, it shall return a wsman:UnsupportedFeature fault with the following detail code:

http://schemas.dmtf.org/wbem/wsman/1/wsman/faultDetail/FragmentLevelAccess

R8.2.1-11: A service may omit returned properties, even when explicitly requested, if and only if such properties have not been set (that is, the properties have a NULL value). The requestor is to interpret the absence of these properties as the properties having a NULL value.

R8.2.1-12: A service shall not return a fault if the association query contains a value for the AssociationClassName, Role, ResultClassName, or ResultRole method parameters that names a CIM element that is not defined in the target CIM namespace or relevant CIM class.

The association query uses these parameters to filter the results and not to define the results.

Clients should use wsman:Filter when using IncludeResultProperty elements because these queries contain projections and are not Boolean predicates.

EXAMPLE: The following request issues an association query in which the returned results include properties from the associated instances as well as the EPRs of the associated instances. This example uses the WS-Management Default Addressing Model but applies to any EPR model used by the service.

```
<s:Envelope>
  <s:Header>
    <wsman:ResourceURI>
      http://schemas.dmtf.org/wbem/wscim/1/*
    </wsman:ResourceURI>
  </s:Header>
  <s:Body>
  <wsen:Enumerate>
    <wsman:EnumerationMode>EnumerateObjectAndEPR</wsman:EnumerationMode>
        <wsman:Filter
          Dialect="http://schemas.dmtf.org/wsman/cimbinding/associationFilter">
            <wsmb:AssociatedInstances>
              <wsmb:Object>
                <wsa:Address> ... </wsa:Address>
                <wsa:ReferenceParameters>
                  <wsman:ResourceURI>
              http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_PhysicalElement
                  </wsman:ResourceURI>
                  <wsman:SelectorSet>
                    <wsman:Selector Name="Tag">81190b2</wsman:Selector>
                    <wsman:Selector Name="CreationClassName">
                      Vendor_PhysicalElement
                    </wsman:Selector>
                  </wsman:SelectorSet>
                </wsa:ReferenceParameters>
              </wsmb:Object>
              <wsmb:AssociationClassName>
                CIM_SystemPackaging
              </wsmb:AssociationClassName>
              <wsmb:ResultClassName>CIM_System</wsmb:ResultClassName>
              <wsmb:IncludeResultProperty>Name</wsmb:IncludeResultProperty>
              <wsmb:IncludeResultProperty>
                PrimaryOwnerName
            </wsmb:IncludeResultProperty>
          </wsmb:AssociatedInstances>
        </wsman:Filter>
      </wsen:Enumerate>
    </s:Body>
  </s:Envelope>
```

The results include the two requested properties as well as the EPR of the associated instances:

```
<s:Body>
    <wsen:PullResponse>
        <wsen:EnumerationContext> ... </wsen:EnumerationContext>
        <wsen:Items>
            <wsman:Item>
                <wsman:XmlFragment>
                    <Name>system1</Name>
                    <PrimaryOwnerName>Joe</PrimaryOwnerName>
                </wsman:XmlFragment>
                <wsa:EndpointReference>
                    <wsa:Address> ... </wsa:Address>
                    <wsa:ReferenceParameters>
                        <wsman:ResourceURI>
    http://schemas.dmtf.org/cim/wscim/1/cim-schema/2/CIM_ComputerSystem
                        </wsman:ResourceURI>
                        ...
                    </wsa:ReferenceParameters>
                </wsa:EndpointReference>
            </wsman:Item>
            <wsman:Item>
                <wsman:XmlFragment>
                    <Name>system2</Name>
                    <PrimaryOwnerName>Mary</PrimaryOwnerName>
                </wsman:XmlFragment>
                <wsa:EndpointReference>
                    <wsa:Address> ... </wsa:Address>
                    <wsa:ReferenceParameters>
                        <wsman:ResourceURI>
                            http://schemas.vendor.com/.../Vendor_System
                        </wsman:ResourceURI>
                        ...
                    </wsa:ReferenceParameters>
                </wsa:EndpointReference>
            </wsman:Item>
            ...etc.
        </wsen:Items>
    </wsen:PullResponse>
</s:Body>
```

8.2.2 Association Instances

For queries that return instances of the association class used in a relationship, the Enumerate message has the following form:

```
(1)  <wsen:Enumerate>
(2)    <wsman:Filter
(3)      Dialect="http://schemas.dmtf.org/wbem/wsman/1/cimbinding/associationFilter">
(4)        <wsmb:AssociationInstances>
(5)          <wsmb:Object> xs:any </wsmb:Object>
(6)          <wsmb:ResultClassName> xs:NCName </wsmb:ResultClassName> ?
(7)          <wsmb:Role> xs:NCName </wsmb:Role> ?
(8)          <wsmb:IncludeResultProperty> xs:NCName </wsmb:IncludeResultProperty> *
(9)        </wsmb:AssociationInstances>
(10)     </wsman:Filter>
(11)   </wsen:Enumerate>
```

The following definitions provide additional, normative constraints on the preceding outline:

- wsen:Enumerate/wsman:Filter/wsmb:AssociationInstances

  The results include association instances related to the source object.

  R8.2.2-1: The results of the enumeration shall be instances of an association class subject to the additional constraints listed in this clause.

- wsen:Enumerate/wsman:Filter/wsmb:AssociationInstances/wsmb:Object

  Identifies the source object for the association query and is required.

  R8.2.2-2: The results shall be instances of association classes for which one of the references is the object identified by this endpoint reference.

  R8.2.2-3: If the EPR to which the Enumerate message is sent and the EPR of the source object represent two different CIM namespaces, the service may return a wsen:CannotProcessFilter fault.

  R8.2.2-4: If the EPR of the source object does not reference exactly one valid CIM instance, the service shall respond with a wsen:CannotProcessFilter fault. Services should include a textual description of the problem.

- wsen:Enumerate/wsman:Filter/wsmb:AssociationInstances/wsmb:ResultClassName

  Represents the name of a CIM association class. This element or parameter is optional.

  R8.2.2-5: If the ResultClassName is present, the results shall contain only instances of the named class or a derived class.

- wsen:Enumerate/wsman:Filter/wsmb:AssociationInstances/wsmb:Role

  Represents the name of a reference property of a CIM association class. This element or parameter is optional.

  R8.2.2-6: If the Role element is present, the results shall include only instances of association classes that refer to the source object through a property whose name matches the value of this parameter.

- wsen:Enumerate/wsman:Filter/wsmb:AssociationInstances/wsmb:IncludeResultProperty

  Represents the name of one or more properties of a CIM class. This element or parameter is optional.

  R8.2.2-7: If the query does not include an IncludeResultProperty element, the service shall return each instance representation using the GED defined for the object's class within the wsen:Items element.

R8.2.2-8: If the query includes one or more IncludeResultProperty elements, the service shall return each instance representation using the wsman:XmlFragment element. Within the wsman:XmlFragment element, the service shall return property values using the property GEDs defined in the *WS-CIM Mapping Specification*. If the query includes one or more IncludeResultProperty elements, the service shall not return any IncludeResultProperty elements not specified. The service shall ignore any IncludeResultProperty elements that describe properties not defined by the target class. If the service does not support fragment-level access, it shall return a wsman:UnsupportedFeature fault with the following detail code:

http://schemas.dmtf.org/wbem/wsman/1/wsman/faultDetail/FragmentLevelAccess

R8.2.2-9: A service may omit returned properties, even if explicitly requested, if and only if such properties have not been set (that is, the properties have a NULL value). The requestor is to interpret the absence of these properties as the properties having a value of NULL.

R8.2.2-10: A service shall not return a fault if the association query contains a value for the Role or ResultClassName method parameters that name a CIM element that is not defined in the target CIM namespace or relevant CIM class.

Clients should use wsman:Filter when using IncludeResultProperty elements as these queries contain projections and are not Boolean predicates.

9 Enumeration

The *WS-Enumeration* specification is used as a basis for iteration through the members of a collection. When enumerating instances of classes, the WS-Management Enumerate operation is used.

9.1 EnumerationMode

Supporting wsman:EnumerationMode enables clients to use enumeration as a method to discover instances. Clients can incorporate one of the EnumerationMode values to obtain the endpoint reference to such instances.

R9.1-1: To maximize interoperation, it is recommended that services that support enumeration also support wsman:EnumerationMode as defined in WS-Management.

EXAMPLE 1: The following example shows an unfiltered enumeration of a class. The class-specific ResourceURI is used when performing a simple unfiltered enumeration:

```
...
<s:Header>
  <wsa:Action>
    http://schemas.xmlsoap.org/ws/2004/09/enumeration/Enumerate
  </wsa:Action>

  <wsman:ResourceURI>
    http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_ComputerSystem
  </wsman:ResourceURI>
  </s:Header>
  <s:Body>
    <wsen:Enumerate/>
  </s:Body>
```

Enumerating this ResourceURI returns all instances of the named class and any derived classes:

```
<CIM_ComputerSystem> <Name>Red-202</Name> ... </CIM_ComputerSystem>
<CIM_ComputerSystem> <Name>Blue-03</Name> ... </CIM_ComputerSystem>
<CIM_ComputerSystem> <Name>Blue-04</Name> </CIM_ComputerSystem>
<Vendor_ComputerSystem> <Name>Green-1</Name> ... </Vendor_ComputerSystem>
```

Each XML instance retrieved by the preceding enumeration contains all the properties of the specific class. For example, the third XML instance is actually of type CIM_UnitaryComputerSystem and might look as follows:

```
<CIM_UnitaryComputerSystem
xmlns= "http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_UnitaryComputerSystem">

  <Name> Blue-04 </Name>
  <PowerManagementSupported> true </PowerManagementSupported>
  <PrimaryOwnerName> Dave </PrimaryOwnerName>
  ...

</CIM_UnitaryComputerSystem>
```

9.2 XmlFragment

XPath allows fragments of the instance to be returned.

R9.2-1: Some filter expressions allow fragments of the instance to be returned. When these ad-hoc queries are performed, the results should be wrapped using wsman:XmlFragment as per R6.6-1 of the *WS-Management Specification*.

EXAMPLE 1: The following filter expression finds the name of all CIM_ComputerSystems owned by Dave and returns just the Name element of the instance provided that the owner is "Dave":

```
XPath: ../CIM_ComputerSystem[PrimaryOwnerName="Dave"]/Name
```

The filter expression results in a PullResponse of the following form:

```
<wsen:PullResponse>
  <wsman:XmlFragment>
    <Name> Red-202 </Name>
  </wsman:XmlFragment>
  <wsman:XmlFragment>
    <Name> Blue-04 </Name>
  </wsman:XmlFragment>
  ...
</wsen:PullResponse>
```

EXAMPLE 2: As a further refinement, just the value alone may be returned:

```
XPath: ../CIM_ComputerSystem[PrimaryOwnerName="Dave"]/Name/text()
```

This modification of the filter expression results in a PullResponse of the following form:

```
<wsen:PullResponse>
  <wsman:XmlFragment> Red-202 </wsman:XmlFragment>
  <wsman:XmlFragment> Blue-04 </wsman:XmlFragment>
  ...
</wsen:PullResponse>
```

9.3 Polymorphism

Many CIM implementations allow polymorphism.

A common way to extend CIM classes is to define derivatives of the CIM class. When a client requests objects of the type for CIM_Process, it is possible to return instances that are actually of a derived type such as Vendor_Process.

The result set may contain instances in accord with one of these three scenarios:

- Results should contain instances from the base class and all derived classes, and each instance should be represented in its actual type including any derived properties.
- Results should contain instances from the base class and all derived classes, but the XML document should be of the base class type and contain only elements corresponding to the properties of the base class.
- Results should contain only instances of the base class and no instances of derived classes.

The default behavior is to return all instances in their native representation.

R9.3-1: A service supporting enumeration shall include instances from the requested class and derived classes in the enumeration result unless otherwise directed by the client.

The client can request other behavior by adding the optional wsmb:PolymorphismMode element as a child element of the wsen:Enumeration element in the Enumeration request, as follows:

```
(
   ...
(
   <s:Body>
(
      <wsen:Enumerate>
(
         ...
(
         <wsmb:PolymorphismMode> ... </wsmb:PolymorphismMode> ?
(
      </wsen:Enumerate>
(
   </s:Body>
```

R9.3-2: A service may optionally support the wsmb:PolymorphismMode modifier element with a value of ExcludeSubClassProperties. The ExcludeSubClassProperties PolymorphismMode shall return instances of the requested class and derived classes represented using the base class's GED and XSD type. Properties defined in the derived class are not returned.

R9.3-3: A service may optionally support the wsmb:PolymorphismMode modifier element with a value of None. The None Polymorphism mode shall return instances of the requested class only.

R9.3-4: A service may optionally support the wsmb:PolymorphismMode modifier element with a value of IncludeSubClassProperties. The IncludeSubClassProperties shall return instances of the requested class and derived classes using the actual class's GED and XSD type. This is the same as not specifying the polymorphism mode.

R9.3-5: If the service does not support the requested polymorphism mode, it should return a wsmb:PolymorphismModeNotSupported fault.

R9.3-6: The service should return a wsmb:PolymorphismModeNotSupported fault for requests using the "all classes" ResourceURI if the PolymorphismMode element is present and does not have a value of IncludeSubClassProperties.

R9.3-7: If both wsman:EnumerationMode and wsmb:PolymorphismMode are supported and wsman:EnumerationMode is present in the request, the service shall always use the Resource URI of the actual class in the returned EPR regardless of the value of wsmb:PolymorphismMode. This allows the client to retrieve and update the actual instance.

EXAMPLE 1: The following example shows an unfiltered enumeration using just base class properties. Using the PolymorphismMode element along with the class-specific ResourceURI yields the same results as the example in 9.1, but the derived type is “cast away” or dropped.

```
...
<s:Header>
  <wsa:Action>
    http://schemas.xmlsoap.org/ws/2004/09/enumeration/Enumerate
  </wsa:Action>

  <wsman:ResourceURI>
    http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_ComputerSystem
  </wsman:ResourceURI>
  </s:Header>
  <s:Body>
    <wsen:Enumerate>
      <wsmb:PolymorphismMode> ExcludeSubClassProperties <wsmb:PolymorphismMode>
    </wsen:Enumerate>
  </s:Body>
```

The same four instances are returned but “cast” as CIM_ComputerSystem:

```
<CIM_ComputerSystem> <Name>Red-202</Name> ... </CIM_ComputerSystem>
<CIM_ComputerSystem> <Name>Blue-03</Name> ... </CIM_ComputerSystem>
<CIM_ComputerSystem> <Name>Blue-04</Name> ... </CIM_ComputerSystem>
<CIM_ComputerSystem> <Name>Green-1</Name> ... </CIM_ComputerSystem>
```

Note that the third instance no longer contains the PowerManagementSupported property added by CIM_UnitaryComputerSystem:

```
<CIM_ComputerSystem
  xmlns="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_ComputerSystem">

  <Name> Blue-04 </Name>
  <PrimaryOwnerName> Dave </PrimaryOwnerName>
  ...

</CIM_ComputerSystem>
```

R9.3-8: If an Enumerate request specifies wsmb:PolymorphismMode=ExcludeSubClassProperties and wsman:EnumerationMode=EnumerateObjectAndEPR or EnumerateEPR, then the service shall return EPRs that reference instances of the most-derived classes of the requested class in the ResourceURI.

The body of the request message appears as follows:

```
<wsen:Enumerate>
  <wsman:EnumerationMode> EnumerateObjectAndEPR </wsman:EnumerationMode>
  <wsmb:PolymorphismMode> ExcludeSubClassProperties </wsmb:PolymorphismMode>
</wsen:Enumerate>
```

The corresponding response message contains the following fragment. Note that the EPR for Blue-04 can be used to access the property PrimaryOwnerName that is not present in the value returned.

```
<wsen:Items>
  <wsman:Item>
    <CIM_ComputerSystem> <Name>Red-202</Name> ... </CIM_ComputerSystem>
    <wsa:EndpointReference>
      <wsa:Address> ... </wsa:Address>
      <wsa:ReferenceParameters>
        <wsman:ResourceURI>
          http://schemas.dmtf.org/.../CIM_ComputerSystem
        </wsman:ResourceURI>
          <wsman:SelectorSet> ... </wsman:SelectorSet>
        </wsa:ReferenceParameters>
      </wsa:EndpointReference>
    </wsman:Item>
  <wsman:Item>
    <CIM_ComputerSystem> <Name>Blue-04</Name> ... </CIM_ComputerSystem>
    <wsa:EndpointReference>
      <wsa:Address> ... </wsa:Address>
      <wsa:ReferenceParameters>
          <wsman:ResourceURI>
            http://schemas.dmtf.org/.../CIM_UnitaryComputerSystem
          </wsman:ResourceURI>
          <wsman:SelectorSet> ... </wsman:SelectorSet>
        </wsa:ReferenceParameters>
      </wsa:EndpointReference>
    </wsman:Item>
    ...
  </wsen:Items>
```

9.4 XPath Enumeration Using the Class-Specific ResourceURI

The ResourceURI contains the class name, as for unfiltered enumeration:

```
<wsman:ResourceURI>
  http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_ComputerSystem
</wsman:ResourceURI>
```

The XPath is anchored at an abstract array of CIM_ComputerSystem XML nodes, which represent all available instances:

```
<CIM_ComputerSystem> ... </CIM_ComputerSystem>
<CIM_ComputerSystem> ... </CIM_ComputerSystem>
<CIM_ComputerSystem> ... </CIM_ComputerSystem>
<CIM_ComputerSystem> ... </CIM_ComputerSystem>
```

The XPath filter expression is evaluated against each possible instance of the specified class, and the instance is either selected as part of the result set or is discarded. PolymorphismMode=ExcludeSubClassProperties is used to ensure that all instances have the same type.

The following XPath expressions all select every instance of CIM_ComputerSystem and are identical:

```
XPath: .
XPath: ../CIM_ComputerSystem
```

To filter, the [ ] filter expressions from XPath may be used. The following selects only instances that have a PrimaryOwnerName property set to "Dave":

```
XPath: ../CIM_ComputerSystem[PrimaryOwnerName="Dave"]
```

If PolymorphismMode=IncludeSubClassProperties were used, the following two XPath filters would have different results:

```
(1)  XPath: .[Owner="Dave"]
(2)  XPath: ../CIM_ComputerSystem[Owner="Dave"]
```

The first XPath would match all instances regardless of type, while the second XPath would select only those instances whose actual type was CIM_ComputerSystem.

9.5 XPath Enumerate Using the "All Classes" ResourceURI

As an alternative to a class-specific ResourceURI, the URI meaning "all classes" may be specified:

```
http://schemas.dmtf.org/wbem/wscim/1/*
```

This URI is a resource that refers to all instances of all classes. In this case, the abstract array of instances is mixed and includes elements of classes other than CIM_ComputerSystem.

```
(1)  <CIM_ComputerSystem> ... </CIM_ComputerSystem>
(2)  <CIM_ComputerSystem> ... </CIM_ComputerSystem>
(3)  <CIM_SoftwareElement> ... </CIM_SoftwareElement>
(4)  <CIM_SoftwareElement> ... </CIM_SoftwareElement>
(5)  <CIM_LogicalDisk> ... </CIM_LogicalDisk>
(6)  <CIM_LogicalDisk> ... </CIM_LogicalDisk>
(7)  <CIM_LogicalDisk> ... </CIM_LogicalDisk>
(8)  ...etc.
```

In the following example, the first query contains no class-specific information. Therefore, the query specifies "all instances of all classes". The second query refers to a specific class:

```
(1)  XPath: .
(2)  XPath: ../CIM_ComputerSystem
```

Services do not typically support the first query if the "all classes" ResourceURI is used, but they may do so.

NOTE: The XPath queries are identical to those provided in 9.4. The ResourceURI simply changes the implied pool of instances over which the query is executed.

10 Subscriptions

The WS-Management Subscribe operation (from WS-Eventing) is used to subscribe to CIM indications. WS-Eventing uses the term "event" for the SOAP message sent to the receiver, while CIM uses the term "indication" for the observation of an event.

The CIM Schema defines a set of special classes to support the delivery of indications to interested receivers. In the CIM Schema, indications are represented by the CIM_Indication class or a subclass of CIM_Indication. Subscriptions can express interest in a set of CIM_Indications by providing a query expression or by referring to an already existing query. This clause outlines the relationship between the WS-Eventing messages and these CIM classes.

A typical scenario for use of CIM indications would be a management client interested in receiving "sensor state change" indications from a device that it is managing. To receive these indications, the client would take the following steps:

1) Construct or identify the indication filter.

2) Create the WS-Eventing Subscribe request.

3) Receive indications.

A management service might need the ability to report on all subscriptions on a server.

In the CIM Schema, subscriptions are represented by a trio of classes:

- CIM_IndicationFilter (or CIM_FilterCollection) captures the query or filter identifying the subset of indications of interest.
- CIM_ListenerDestination captures information about where or how the indications are to be delivered.
- CIM_IndicationSubscription (or CIM_FilterCollectionSubscription) associates an instance of CIM_IndicationFilter (or CIM_FilterCollection) with CIM_ListenerDestination.

These classes are used in different parts of the subscription life cycle, as indicated in the remainder of this clause.

R10-1: A service that supports subscriptions shall do so using the WS-Eventing operations as defined in WS-Management. It is recommended that a service internally create the requisite CIM indication-related instances when the service accepts a subscription using the Subscribe message from a Web services client.

R10-2: A service may deliver indications based on the creation of instances of the CIM indication-related classes in addition to supporting WS-Eventing.

R10-3: A service that does not support the WS-Management Default Addressing Model is not required to conform to the rules for the ResourceURI described in the text and examples in the following subclauses (clause 10 and its subclauses). All examples about WS-Eventing filter dialects apply to services independent of their addressing model.

10.1 Indication Filters

When subscribing to indications, the same XPath and CQL filter usage is observed as for enumerations. However, association queries are not applicable to subscriptions.

When CQL is used, the subscription filter includes the name of the class being selected for the subscription:

```
select * from CIM_AlertIndication where MessageID="394"
```

CQL statements with projections can also be used, in which case the selected properties of the indications are wrapped using wsman:XmlFragment as described in 8.1.

The same filter can be expressed in XPath:

```
../CIM_AlertIndication[MessageID="394"]
```

XPath filters can also be written without identifying the class. The same filter could be expressed using the following XPath filter if it were applied to instances of CIM_AlertIndication:

```
./[MessageID="394"]
```

These filter expressions can be formulated by the client, or they might already exist on the server (as an instance of CIM_IndicationFilter).

10.2 Subscribe Request

The client constructs the subscribe request to express interest in a subset of the indications on the service. The client can filter the indications by specifying a filter directly in the subscribe request or by referring to an existing filter stored on the service.

10.2.1 Subscribing Using a Filter

When subscribing using a filter expression, the client can target the subscribe request to either the CIM Server or a specific indication class.

10.2.1.1 Subscribing to the CIM Server

When subscribing to the CIM Server, a filter dialect such as CQL can be used. In this case, the query alone contains the necessary information as to which class is being filtered and the "all classes" ResourceURI can be used for addressing.

R10.2.1.1-1: If a service supports client-supplied CQL expressions and the WS-Management Default Addressing Model, it should accept wse:Subscribe messages addressed to the "all-classes" ResourceURI.

EXAMPLE: The following example shows a Subscribe message to set up a subscription for changes in sensor state. It is addressed to the "all classes" ResourceURI and uses a CQL filter to detect instance indications in which the CurrentState property has changed:

```
<s:Envelope 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"
    xmlns:wse="http://schemas.xmlsoap.org/ws/2004/09/eventing">
  <s:Header>
    <wsa:Action>
      http://schemas.xmlsoap.org/ws/2004/08/eventing/Subscribe
    </wsa:Action>
    <wsa:To> http://127.0.0.1:9999/wsman </wsa:To>
    <wsa:MessageID> . . . </wsa:MessageID>
    <wsa:ReplyTo>
      http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
    </wsa:ReplyTo>
    <wsman:ResourceURI>
      http://schemas.dmtf.org/wbem/wscim/1/*
    </wsman:ResourceURI>
  </s:Header>
  <s:Body>
    <wse:Subscribe>
      <wse:Delivery
          Mode="http://schemas.dmtf.org/wbem/wsman/1/wsman/PushWithAck">
        <wse:NotifyTo>
          <wsa:Address> . . . </wsa:Address>
          . . .
        </wse:NotifyTo>
      </wse:Delivery>
      <wsman:Filter dialect="http://schemas.dmtf.org/wbem/cql/1/dsp0202.pdf">
```

```
          <!-- whenever the state of any sensor changes -->
          SELECT *
          FROM CIM_InstIndication
          WHERE SourceInstance ISA CIM_Sensor
            AND PreviousInstance ISA CIM_Sensor
            AND PreviousInstance.CIM_Sensor::CurrentState &lt;&gt;
                SourceInstance.CIM_Sensor::CurrentState
        </wsman:Filter>
      </wse:Subscribe>
    </s:Body>
  </s:Envelope>
```

When subscribing to the CIM Server, instances of all classes are implicitly addressed; therefore, separate polymorphism modes are not relevant.

R10.2.1.1-2: A service supporting wse:Subscribe messages addressed to the “all classes” ResourceURI shall return a wsmb:PolymorphismModeNotSupported fault if the wsmb:PolymorphismMode modifier is present and does not equal IncludeSubClassProperties.

10.2.1.2 Subscribing to an Indication Class

A subset of all indications can also be expressed by subscribing to an indication class. In this case, the EPR contains the necessary information as to which class is being filtered. An additional filter might or might not be present, but it would apply only to the instances of class indicated by the EPR.

R10.2.1.2-1: If a service supports client filtering over a particular class of indications and the WS-Management Default Addressing Model, it should accept wse:Subscribe messages addressed to the class-specific ResourceURI for CIM_Indication or a subclass of CIM_Indication.

EXAMPLE: The following example shows a Subscribe message to set up a subscription for changes in temperature sensors. It is addressed to the resource URI for the CIM_AlertIndication class and uses XPath to select instances of the class in which one of the desired messages is present:

```
<s:Envelope 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"
    xmlns:wse="http://schemas.xmlsoap.org/ws/2004/09/eventing" >
  <s:Header>
    <wsa:Action>
      http://schemas.xmlsoap.org/ws/2004/08/eventing/Subscribe
    </wsa:Action>
    <wsa:To> http://127.0.0.1:9999/wsman </wsa:To>
      <wsa:MessageID> . . . </wsa:MessageID>
      <wsa:ReplyTo>
        http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
      </wsa:ReplyTo>
      <wsman:ResourceURI>
        http://schemas.dmtf.org/wbem/wscim/1/CIM_AlertIndication
      </wsman:ResourceURI>
    </s:Header>
    <s:Body>
      <wse:Subscribe>
        <wse:Delivery
            Mode="http://schemas.dmtf.org/wbem/wsman/1/wsman/PushWithAck">
          <wse:NotifyTo>
```

```
(23)           <wsa:Address> . . . </wsa:Address>
(24)           . . .
(25)         </wse:NotifyTo>
(26)       </wse:Delivery>
(27)       <wsman:Filter
(28)           xmlns:c="http://schemas.dmtf.org/wbem/wscim/1/CIM_AlertIndication">
(29)         .[c:OwningEntity="DMTF" and (c:MessageID="394" or c:MessageID="396"
(30)         or c:MessageID="398" or c:MessageID="400" or c:MessageID="413")]
(31)       </wsman:Filter>
(32)     </wse:Subscribe>
(33)   </s:Body>
(34) </s:Envelope>
```

Additional filtering, such as XPath filters, on the instances of CIM_AlertIndication that are identified by the EPR can be allowed. However, this practice is discouraged because using CQL expressions in this context creates the possibility for contradictions between the class identified by the EPR and the class identified in the CQL expression.

R10.2.1.2-2: A service that supports a class-specific ResourceURI as a target of the wse:Subscribe message should return the wse:InvalidMessage fault if such messages specify a filter that includes class information as part of the filter expression.

When the wse:Subscribe message is addressed to an indication class, the wsmb:PolymorphismMode element described in 9.3 can be used to control how polymorphism is handled for indications on event delivery. The wsmb:PolymorphismMode element becomes a child element of the Subscribe element.

R10.2.1.2-3: A service supporting wse:Subscribe messages addressed to a CIM indication class through a class-specific ResourceURI shall provide indication instances from the requested class and its subclasses in event delivery unless otherwise directed by the client.

R10.2.1.2-4: A service supporting wse:Subscribe messages addressed to a CIM indication class through a class-specific ResourceURI may support the use of the wsmb:PolymorphismMode modifier as a child of the wse:Subscribe element, with the resulting event instances typed according to rules **R9.3-2**, **R9.3-3**, and **R9.3-4**.

10.2.2 Subscribing to an Existing Filter

The service may have existing filters because of profile provisions implemented or filters previously created by a client. The client needs a way to express interest in one of these filters. These filters are represented by instances of either the CIM_IndicationFilter or CIM_FilterCollection classes; hereafter these instances are referred to as existing filters.

R10.2.2-1: If a service supports filtering using an existing filter expression and the WS-Management Default Addressing Model, it should accept wse:Subscribe messages addressed to the class-specific ResourceURI for an instance of the existing filter class.

EXAMPLE: The following example shows a Subscribe message to set up a subscription to an existing filter named by "example.org::temperatureSensors::stateChanges":

```
(1) <s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
(2)     xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
(3)     xmlns:wsman="http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd"
(4)     xmlns:wse="http://schemas.xmlsoap.org/ws/2004/09/eventing" >
(5)   <s:Header>
(6)     <wsa:Action>
(7)       http://schemas.xmlsoap.org/ws/2004/08/eventing/Subscribe
(8)     </wsa:Action>
```

```
    <wsa:To> http://127.0.0.1:9999/wsman </wsa:To>
    <wsa:MessageID> . . . </wsa:MessageID>
    <wsa:ReplyTo>
      http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
    </wsa:ReplyTo>
    <wsman:ResourceURI>
      http://schemas.dmtf.org/wbem/wscim/1/CIM_IndicationFilter
    </wsman:ResourceURI>
    <wsman:SelectorSet>
      <wsman:Selector name="Name">
        example.org::temperatureSensors::stateChanges
      </wsman:Selector>
      <wsman:Selector name="SystemCreationClassName">
        CIM_ComputerSystem
      </wsman:Selector>
      <wsman:Selector name="__cimnamespace">interop</wsman:Selector>
    </wsman:SelectorSet>
  </s:Header>
  <s:Body>
    <wse:Subscribe>
      <wse:Delivery
          Mode="http://schemas.dmtf.org/wbem/wsman/1/wsman/PushWithAck">
        <wse:NotifyTo>
          <wsa:Address> . . . </wsa:Address>
          . . .
        </wse:NotifyTo>
      </wse:Delivery>
      <!-- wse:Filter and wsman:Filter not permitted in this case. -->
    </wse:Subscribe>
  </s:Body>
</s:Envelope>
```

R10.2.2-2: If a service supports filtering using an existing filter expression (as indicated by the EPR), the service message shall return the wsman:InvalidParameter fault if the wse:Subscribe request includes a filter expression (such as in the wse:Filter or wsman:Filter elements).

R10.2.2-3: A service supporting Subscribe to an existing filter using the WS-Management Default Addressing Model should support access using a class-specific ResourceURI corresponding to a filter with selector values that identify the instance of the actual class of the desired filter. The referenced base class shall be one for which CIM keys have been defined; otherwise, the service should respond with a wsman:InvalidSelectors fault with the following detail code:

http://schemas.dmtf.org/wbem/wsman/1/wsman/faultDetail/UnexpectedSelectors

When subscribing to an existing filter, the classes of interest are indicated by the filter expression and separate polymorphism modes are not relevant.

R10.2.2-4: A service supporting wse:Subscribe messages addressed to an instance of CIM_IndicationFilter or CIM_FilterCollection through a class-specific ResourceURI shall return a wsmb:PolymorphismModeNotSupported fault if the wsmb:PolymorphismMode modifier is present and does not equal IncludeSubClassProperties.

Subscribing to an instance of CIM_IndicationFilter (or CIM_FilterCollection) works regardless of whether or not the service created the filter or if a client constructed the instance prior to sending the Subscribe message. The client can construct instances of these filter classes using mechanisms such as WS-Transfer Create. In this case, the service is accepting a client-defined filter expression, so the service must also accept the same filter expression in a Subscribe message.

R10.2.2-5: If a service supports creating an instance of CIM_IndicationFilter (using WS-Transfer Create or another mechanism), the service shall also support a wse:Subscribe message in which the filter expression is specified in the wsman:Filter element in body of the Subscribe message.

10.3 Subscription Response

A successful SubscribeResponse message includes a SubscriptionManager element containing an EPR to be used to Unsubscribe from or Renew this subscription.

R10.3-1: The SubscriptionManager EPR in a successful SubscribeResponse shall be unique to the subscription created by the Subscribe request.

That is, the SubscriptionManager EPR returned by the service shall contain some elements that correlate `one-to-one with the single subscription that was just created.

R10.3-2: A service shall accept an Unsubscribe or Renew request whose EPR matches a SubscriptionManager EPR that was previously returned to a client, provided that the subscription is still active.

That is, if a service accepts a subscription and returns a SubscriptionManager EPR to a client, the service shall accept that EPR as the target of an Unsubscribe or Renew message.

Because both the client and the service depend on this EPR, the SubscriptionManager EPR shall be valid for the duration of the subscription.

10.4 Event Delivery

When instances of CIM_Indication or a subclass are indicated by the eventing infrastructure, they are delivered as event SOAP messages according to the delivery mode in the wse:Subscribe request. The following rules describe the XML representation of the indication:

R10.4-1: When delivering the event XML for an indication, the wsa:Action URI of the event should be set to the same value as the XML namespace for the actual class of the indication instance.

R10.4-2: When delivering the event XML for an indication, the event body shall be the XML representation of the indication instance as per the *WS-CIM Mapping Specification*, subject to any additional client requests such as projection or polymorphism.

EXAMPLE: The following example shows an instance of CIM_InstModification delivered as a single event using the Push delivery mode:

```
(1) <s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
(2)     xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
(3)     xmlns:wsman="http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd"
(4)     xmlns:class=
(5)         "http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_InstModification"
(6)     xmlns:common="http://schemas.dmtf.org/wbem/wscim/1/common"
(7)     xmlns:wse="http://schemas.xmlsoap.org/ws/2004/09/eventing">
(8)   <s:Header>
(9)     <wsa:Action>
(10)        http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_InstModification
(11)      </wsa:Action>
```

```
(12)     <wsa:To> . . . </wsa:To>
(13)     <wsa:MessageID> . . . </wsa:MessageID>
(14)   </s:Header>
(15)   <s:Body>
(16)     <class:CIM_InstModification>
(17)       <class:IndicationIdentifier>
(18)         CIM:12345678-abcd-0000-fedc-0123456789ab
(19)       </class:IndicationIdentifier>
(20)       <class:IndicationTime>
(21)         <common:dateTime>2007-04-01T11:22:33.123Z</common:dateTime>
(22)       </class:IndicationTime>
(23)       <class:PerceivedSeverity>5</class:PerceivedSeverity>
(24)       <class:PreviousInstance> . . . </class:PreviousInstance>
(25)       <class:SourceInstance> . . . </class:SourceInstance>
(26)       <class:SourceInstanceHost>10.57.217.39</class:SourceInstanceHost>
(27)       <class:SourceInstanceModelPath> . . . </class:SourceInstanceModelPath>
(28)     </class:CIM_InstModification>
(29)   </s:Body>
(30) </s:Envelope>
```

10.5 Subscription Reporting

Subscription Reporting is the ability of an implementation to report on the existing filters, collections, and subscriptions. Subscriptions can be created and deleted through the Subscribe and Unsubscribe operations. Filters and subscriptions may also be created, modified, and deleted directly using other protocol operations described in this specification. An implementation should instantiate instances that reflect the results of the operations described in this specification.

R10.5-1: It is recommended that a service create in its CIM service the requisite CIM indication-related instances when the service accepts a subscription using the Subscribe message from a Web services client. The CIM namespace in which these instances are created is beyond the scope of this specification.

The rules in the following clauses describe requirements for the content of the CIM indication-related classes if such reporting is supported as recommended in the preceding rule.

Every active subscription contains three components:

- An instance of CIM_IndicationFilter or CIM_FilterCollection that describes the indications to be delivered;
- An instance of CIM_ListenerDestinationWSManagement that describes the client-specified endpoint for delivery of indications; and
- An instance of CIM_IndicationSubscription or CIM_FilterCollectionSubscription that links the filter and the destination, and describes additional characteristics of the subscription.

10.5.1 CIM_IndicationFilter

The CIM_IndicationFilter class captures the filter used in the subscription.

R10.5.1-1: If a subscribe request contains a filter expression, a service shall create an instance of CIM_IndicationFilter and set the properties as indicated in Table 2.

Table 2 – CIM_IndicationFilter Properties

Property Name	Value
Query	Filter expression from the Subscribe request, including XML if appropriate for the indicated QueryLanguage
QueryLanguage	Dialect URI from the Subscribe request For example, if a CQL expression were used in the Subscribe request the URI would be: http://schemas.dmtf.org/wbem/cql/1/dsp0202.pdf

When subscribing to an existing filter expression, the instance of CIM_IndicationFilter already exists so a new instance is not created.

10.5.2 CIM_ListenerDestinationWSManagement

The CIM_ListenerDestinationWSManagement class captures the endpoint for event delivery.

R10.5.2-1: A service shall ensure that, for each subscribed endpoint, an instance of CIM_ListenerDestinationWSManagement exists and contains the properties as indicated in Table 3.

Table 3 – CIM_ListenerDestinationWSManagement Required Properties

Property Name	Value
Protocol	4 ("WS-Management")
Destination	The URL in the wsa:Address element of wse:NotifyTo If the delivery mode does not have a destination EPR (such as the Pull delivery mode), the WS-Addressing anonymous URI should be used as a place holder. Using the anonymous URI indicates that the event sink will contact the event source; the anonymous URI is not to be confused with the ReplyTo EPR in that request.

A WS-Management subscription contains a number of terms that extend the concept of a CIM subscription. Additional properties in CIM_ListenerDestinationWSManagement capture these extensions. In most cases, the values of the new properties come from elements in the Subscribe request. In a few cases, the values are dictated by the WS-Mananagement protocol.

These properties are likely to be managed by users and client applications, and they might be of interest to users enumerating existing subscriptions. Some small footprint implementations of WS-Management services might not wish to expose all these properties.

R10.5.2-2: If the subscribe request specifies any of the following options, the corresponding properties of the CIM_ListenerDestinationWSManagement instance should be set according to the values shown in Table 4. These guidelines might be updated by newer versions of this class; the actual MOF definition takes precedence over the information in Table 4.

Table 4 – CIM_ListenerDestinationWSManagement Optional Properties

Property Name	Value
DestinationEndTo	Similar to Destination, but applies to the EndTo EPR, if present
Locale	RFC 3066 language code from the Subscribe request, if present
ContentEncoding	The value of the ContentEncoding element from the Subscribe request, if present
DeliveryMode	A ValueMap value that captures the Delivery/@Mode URI from the Subscribe request

Property Name	Value
Heartbeat	Interval in seconds at which point a heartbeat event will be sent if no other events have been sent
SendBookmarks	True if the SendBookmarks element was present in the Subscribe request
MaxTime	The time in seconds to build a batch when using a batching delivery mode
DeliveryAuth	The security profile URI being used by the event source when delivering events through a Push delivery mode
PolymorphismMode	A ValueMap value that captures the polymorphism choice if present in the Subscribe request

In general, instances of ListenerDestinationWSManagement are not reusable because of the terms of the subscription and the rules regarding their deletion when a subscription ends. Whether instances are shared is beyond the scope of this specification.

10.5.3 CIM_IndicationSubscription and CIM_FilterCollectionSubscription

The CIM_IndicationSubscription and CIM_FilterCollectionSubscription classes capture associations between the indication filter or filter collection and the endpoint for event delivery. An instance of one of these classes represents the subscription created by the Subscribe request.

R10.5.3-1: If a Subscribe request is addressed to an instance of CIM_IndicationFilter, or results in the creation of an instance of CIM_IndicationFilter, then a service shall create an instance of CIM_IndicationSubscription and set the properties as indicated in Table 5 as part of a successful Subscribe operation.

Table 5 – Required Properties for CIM_IndicationSubscription and CIM_FilterCollectionSubscription

Property Name	Value
SubscriptionDuration	The time at which the subscription expires as indicated in the Subscribe response
OnFatalErrorPolicy = "Remove"	Not applicable
RepeatNotificationPolicy = "None"	Not applicable
SubscriptionInfo	Unique value identifying the subscription

R10.5.3-2: If a subscription request is addressed to an instance of CIM_FilterCollection, then a service shall instead create an instance of CIM_FilterCollectionSubscription with properties as indicated in Table 5.

R10.5.3-3: If a service that supports Renew created an instance of CIM_IndicationSubscription (or CIM_FilterCollectionSubscription) when processing the Subscribe message, it shall update the SubscriptionDuration to reflect the new expiration time when processing the Renew message.

WS-Eventing uses the subscription manager EPR in the SubscribeReponse message to identify the subscription. It defines the wse:Identifier element for use as a reference parameter in this EPR, but it is not required. For convenience, it is recommended that this element be used and match the SubscriptionInfo property.

R10.5.3-4: A service should populate the SubscriptionInfo field with a URI to identify the subscription and use the same value as the value of the wse:Identifier reference parameter in the SubscriptionManager EPR.

Services can use the same URI format as outlined in 2.7 of the *WS-Management Specification* for wsa:MessageID.

10.5.4 Proxy Considerations

In some cases, the WS-Management service might be a proxy or adapter to an existing system. Such implementations have the following two pieces of information to track:

- the information about the subscription between the client and the WS-Management service
- the information about the subscription between the WS-Management service and the CIM Server

The rules in this specification describe how to represent the information about the subscription between the client and the WS-Management service. The representation of the information between the WS-Management service and the CIM Server is beyond the scope of this specification.

Implementations can choose to represent this "local" subscription using similar techniques, but the information would differ in properties such as the CIM_ListenerDestination.Destination that would be the address of the WS-Management service for the local subscription. Implementations can choose to create parallel subscriptions for each or do analysis to avoid sending the same indication multiple times on the local channel.

10.6 Unsubscribe and Renew Requests

A client may extend the duration of a subscription using a wse:Renew request, if the service supports such requests.

R10.6-1: If a service supports eventing but does not support renewing subscriptions, the service may fault a wse:Renew request with the fault code wse:UnableToRenew. If a service supports eventing, the service shall not fault a wse:Renew request with fault code wsa:ActionNotSupported

Unsubscribe and Renew requests may be addressed to a service using the SubscriptionManager EPR that was returned in the SubscribeResponse message.

In lieu of using the SubscriptionManager EPR from the SubscribeResponse message, a client may construct a new SubscriptionManager EPR of a particular form that is acceptable to the service. If the ReferenceParameters of the EPR uniquely specify an existing instance of IndicationSubscription or FilterCollectionSubscription, a service is required to accept the Unsubscribe or Renew request at the normal protocol endpoint address, that is, the protocol endpoint where that subscription can be seen with Enumerate or Get. The To address of the SubscriptionManager EPR is not necessarily valid over long periods of time; the address may change because of dynamic addressing assigned to the protocol endpoint or subscription manager service.

R10.6-2: A service shall accept an Unsubscribe request or Renew request whose EPR specifies a valid instance of IndicationSubscription or FilterCollectionSubscription. A service shall accept a request of this form at the To address of the protocol endpoint at which the subscription can be accessed with Enumerate or Get operations. A service may also accept a request of this form at the To address of the SubscriptionManager EPR.

If the EPR does not specify a valid and unique IndicationSubscription or FilterCollectionSubscription, then the service shall fault the request. For instance, if a subscription has been terminated for any reason, then a SubscriptionManager EPR or a constructed EPR specifying that subscription will not be valid.

R10.6-3: A service shall delete at most one subscription as a result of an Unsubscribe request.

The Unsubscribe request shall be sufficiently specific that it removes one subscription, or none in the case of a fault for any reason.

When a subscription is terminated, a service is required to clean up data structures that were created to represent the subscription.

When a subscriber is no longer interested in receiving indications from a subscription, it can cancel the subscription using a wse:Unsubscribe request.

R10.6-4: If a service created CIM indication-related instances as described in 10.5, then the service shall delete those instances when the subscription is canceled for any reason.

In all cases, the instance of CIM_IndicationSubscription (or CIM_FilterCollectionSubscription) is deleted because this instance represents the actual subscription.

Instances of the other members of the association might be reused between subscriptions. For example, if a subscription were addressed to an existing filter (an instance of CIM_IndicationFilter), then that instance need not be deleted when the subscription is deleted. The exact ownership of these instances and a method to determine when to delete them is beyond the scope of this specification.

11 Extrinsic Methods

Invoking an extrinsic method uses the action URIs and messages defined by the *WS-CIM Mapping Specification* (clause 8.3, "CIM Methods to WSDL Mappings"). The request and response message schemas for an extrinsic method are defined in the WS-CIM schema for the CIM class that defines the method (and the request and response message schemas use the XML namespace for that class). The wsa:Action URIs are derived from the XML namespace of the class and the method name as per the *WS-CIM Mapping Specification*. The endpoint reference is transformed into SOAP headers as defined by WS-Addressing in the same way as other WS-Management operations.

When using the WS-Management Default Addressing Model, the rules for ResourceURI and selector usage are the same as those described in clause 7 of this specification.

12 Exceptions

For some CIM server implementations, invoking either an intrinsic or extrinsic method can result in the production of one or more exceptions before the corresponding method completes on the CIM server. In this case, the requested CIM operation may not be able to successfully complete and the service may not be able to return the output for the operation. The service responds with a SOAP fault message containing the exception instances according to the following rules:

R12-1: If a service receives a WS-Management request message that translates into a CIM intrinsic or extrinsic method, the execution of the method results in one or more exceptions, the requested CIM operation does not complete, and the service is not able to return the output for the operation, the service should respond with a SOAP fault.

R12-2: A service responding to a WS-Management request that translated into a CIM intrinsic or extrinsic method that did not complete and resulted in an exception should include each resultant exception object as peers in the SOAP fault's Detail element. The XML representation of each exception object shall conform to the mapping rules for CIM instances defined in the *WS-CIM Mapping Specification*.

R12-3: A service responding to a WS-Management request that translated into a CIM intrinsic or extrinsic method that did not complete and resulted in an exception should use WS-Management fault subcodes that correspond to the nature of the exception that has occurred. If the exception does not correspond to any defined WS-Management fault subcode, the service should use the wsmb:CIMException subcode.

EXAMPLE: A fault response for an extrinsic method containing an invalid method parameter that results in a CIM exception would have the following structure:

```
<env:Fault>
 <env:Code>
   <env:Value>env:Sender</env:Value>
   <env:Subcode>
    <env:Value>wsman:InvalidParameter</env:Value>
   </env:Subcode>
 </env:Code>
 <env:Reason>
   <env:Text xml:lang="en">
       The invocation of CIM method RequestStateChange
       failed because the unknown parameter Spongebob
       has been supplied.
    </env:Text>
  </env:Reason>
  <env:Detail>
    <wsman:FaultDetail>
       http://schemas.dmtf.org/wbem/wsman/1/wsman/faultDetail/InvalidName
    </wsman:FaultDetail>
    <cimerr:CIM_Error>
        …as in WS-CIM…
    </cimerr:CIM_Error>
  </env:Detail>
</env:Fault>
```

For further information on the mapping of CIM exceptions to WS-Management fault subcodes, see clause 17.

13 CIM Specific WS-Management Options

This specification relies on the WS-Management OptionSet extensibility mechanism for common scenarios.

13.1 ShowExtensions Option

Some of the optional CIM properties may be expensive to calculate; as a result, they are not included in casual queries for the resource representation. Also, in some CIM Server implementations, the CIM Server may define additional system properties that are stored along with the standard CIM properties of a given class and that are exposed using the open content model defined in the XML Schema specified in the *WS-CIM Mapping Specification*.

The use of ShowExtensions allows a client to indicate that the XML resource representation should contain the elements that are expensive to calculate and the extension elements, along with the rest of the resource properties. The ShowExtensions option may be applied to the WS-Transfer Get message, the WS-Enumeration Enumerate message, and the WS-Eventing Subscribe message.

When this option is applied to Enumerate, it communicates the desire for all resource representations returned by the enumeration sequence to include the extensions independent of whether they are returned in an EnumerateResponse or a PullResponse message.

When this option is applied to a Subscribe message, it communicates the desire for all events matching that Subscribe message to be returned with the extensions.

This specification does not define any meaning for the ShowExtensions option on other messages. If necessary, the client may place extra content in Put and Create messages using the extension mechanism defined in the *WS-CIM Mapping Specification*.

Because vendor extensions can be large or expensive to retrieve, a standard option has been defined to enable or disable the vendor extensions to be returned with the resource representation. The default is to disable the return of vendor extensions.

To show all extensions, a client sets the Option value to ShowExtensions, as follows:

```
(1) <wsman:OptionSet>
(2)   <wsman:Option name="ShowExtensions"/>
(3) <wsman:OptionSet>
```

To hide extensions, a client omits or sets the Option to FALSE or 0. Any other value or an empty element implies that the extensions should be shown.

R13.1-1: If a service receives a request with an OptionSet containing an Option named ShowExtensions in which the OptionSet header has mustUnderstand="TRUE" and the Option element has mustComply="TRUE" and the value of the Option element is FALSE or 0, the service shall return the representation in minimal form or issue a fault.

R13.1-2: If a service receives a request with an OptionSet containing an Option named ShowExtensions in which the OptionSet header has mustUnderstand="TRUE" and the Option element has mustComply="TRUE" and the value of the Option element is neither false nor 0, the service shall return the representation with additional information including the cim:Key and cim:Version attributes as per the *WS-CIM Mapping Specification* and any vendor-defined extensions or issue a fault.

R13.1-3: In the absence of this option (or mustComply requirements), a service should return the representation in minimal form or issue a fault.

EXAMPLE: The following shows an example representation from a service that has implemented CIM schema version 2.11.0 that includes extensions. Note that all the vendor-specific properties come after the class properties.

```
(1) <CIM_ComputerSystem
(2)    xmlns="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_ComputerSystem"
(3)    xmlns:cim="http://schemas.dmtf.org/wbem/wscim/1/common"
(4)    xmlns:v="http://vendor.com/..."
(5)    cim:Version="2.7.0">
(6)
(7)   <CreationClassName cim:Key="true"> ... </CreationClassName>
(8)   <Name cim:Key="true"> Blue-04 </Name>
(9)   <PrimaryOwnerName> Dave </PrimaryOwnerName>
(10)    ...
(11)    <v:PropetyCount>17</v:PropertyCount>
(12) </CIM_ComputerSystem>
```

14 Instance Representation

Instances are represented according to the XML namespace defined by the *WS-CIM Mapping Specification*. This clause defines additional constraints on that representation.

WS-CIM allows references to be represented using a variety of addressing models; however, this specification is associated with WS-Management, which uses a specific addressing model.

R14-1: A service shall accept and return only instance representations in which XML elements corresponding to CIM reference properties are represented as a WS-Addressing EPR using the wsa XML namespace defined in clause 6.

15 Fault Codes

Faults defined in this specification must use the following action URI:

http://schemas.dmtf.org/wbem/wsman/1/cimbinding/fault

15.1 wsmb:CIMException

Table 6 provides information about the wsmb:CIMException fault subcode.

Table 6 – wsmb:CIMException

Fault Subcode	**wsmb:CIMException**
Action URI	http://schemas.dmtf.org/wbem/wsman/1/cimbinding/fault
Code	s:Receiver
Reason	The CIM server encountered an exception during the processing of the request.
Detail	XML representation of CIM_Error instance
Comments	
Applicability	Any message
Remedy	Depends upon the exception

15.2 wsmb:PolymorphismModeNotSupported

Table 7 provides information about the wsmb:PolymorphismModeNotSupported fault subcode.

Table 7 – wsmb:PolymorphismModeNotSupported

Fault Subcode	**wsmb:PolymorphismModeNotSupported**
Action URI	http://schemas.dmtf.org/wbem/wsman/1/cimbinding/fault
Code	s:Sender
Reason	The resource does not support the requested polymorphism mode.
Detail	
Comments	
Applicability	wsen:Enumerate, wse:Subscribe
Remedy	Try the request again without specifying a polymorphism mode.

16 Mapping for DSP0200 CIM Operations

CIM profiles define support for CIM operations for each CIM class used in the profiles. These supported operations are defined in DSP0200. This clause outlines the WS-Management equivalent operations for each supported CIM operation that is defined in DSP0200 and additional uses of WS-Management functionality to achieve the same goal.

16.1 Supported Operations

The following CIM operations have equivalents defined by this specification:

- GetInstance: This operation is used to return a single CIM instance from the target namespace.
- DeleteInstance: This operation is used to delete a single CIM instance from the target namespace.
- ModifyInstance: This operation is used to modify a single CIM instance in the target namespace.
- CreateInstance: This operation is used to create a single CIM instance in the target namespace.
- EnumerateInstances: This operation is used to enumerate instances of a CIM Class (this includes instances in the class and any subclasses in accordance with the polymorphic nature of CIM objects) in the target Namespace.
- EnumerateInstanceNames: This operation is used to enumerate the names (model paths) of the instances of a CIM Class (this includes instances in the class and any subclasses in accordance with the polymorphic nature of CIM objects) in the target Namespace.
- Associators: This operation is used to enumerate CIM Objects (Classes or Instances) that are associated to a particular source CIM Object.
- AssociatorNames: This operation is used to enumerate the names of CIM Objects (Classes or Instances) that are associated to a particular source CIM Object.
- References: This operation is used to enumerate the association objects that refer to a particular target CIM Object (Class or Instance).
- ReferenceNames: This operation is used to enumerate the association objects that refer to a particular target CIM Object (Class or Instance).
- OpenEnumerateInstances: The OpenEnumerateInstances operation establishes and opens an enumeration session of the instances of a CIM class (including instances of its subclasses) in the target namespace. Optionally, it retrieves a first set of instances.
- OpenEnumerateInstancePaths: The OpenEnumerateInstancePaths operation establishes and opens an enumeration session of the instance paths of the instances of a CIM class (including instances of its subclasses) in the target namespace. Optionally, it retrieves a first set of instance paths.
- OpenReferenceInstances: The OpenReferenceInstances operation establishes and opens the enumeration session of association instances that refer to a particular target CIM instance in the target namespace. Optionally, it retrieves a first set of instances.
- OpenReferenceInstancePaths: The OpenReferenceInstancePaths operation establishes and opens an enumeration session of the instance paths of the association instances that refer to a particular target CIM instance in the target namespace. Optionally, it retrieves a first set of instance paths.
- OpenAssociatorInstances: The OpenAssociatorInstances operation establishes and opens an enumeration session of the instances associated with a particular source CIM instance in the target namespace. Optionally, it retrieves a first set of instances.

- OpenAssociatorInstancePaths: The OpenAssociatorInstancePaths operation establishes and opens an enumeration session of the instance paths of the instances associated with a particular source CIM instance in the target namespace. Optionally, it retrieves a first set of instance paths.
- PullInstancesWithPath: The PullInstancesWithPath operation retrieves instances including their instance paths from an open enumeration session represented by an enumeration context value.
- PullInstancePaths: The PullInstancePaths operation retrieves instance paths from an open enumeration session represented by an enumeration context value.
- CloseEnumeration: The CloseEnumeration operation closes an open enumeration session, performing an early termination of an enumeration sequence.

The following sub-sections define the mapping of the above operations over WS-Management.

16.1.1 GetInstance

The mapping defined in Table 8 shall be used for the GetInstance operation.

Table 8 – GetInstance

Operation	**GetInstance**
Operation target	CIM namespace
WS-Man operation	WS-Transfer:Get
WS-Man EPR	Class-specific ResourceURI with keys as selectors
Additional usage	None
Notes	As defined in R7.1-1, the class specified in the Resource URI needs to be the creation class of the instance it addresses.

Table 9 provides the mapping of GetInstance arguments defined in Section 2.3.2.2 of DSP0200.

Table 9 – GetInstance Arguments

Argument	**GetInstance**
InstanceName	Mapped to WS-Man EPR
LocalOnly	There is no corresponding WS-Management parameter for this argument. From the expected behavior perspective, this argument is treated as always set to false.
IncludeQualifier	There is no corresponding WS-Management parameter for this argument. From the expected behavior perspective, this argument is treated as always set to false.
IncludeClassOrigin	There is no corresponding WS-Management parameter for this argument. From the expected behavior perspective, this argument is treated as always set to false.
PropertyList[ ]	If it is NULL, then the operation is handled through WS-Transfer Get. If it is not NULL, then the operation is handled through fragment level WS-Transfer Get (see Section 4.9 of DSP0226).

Table 10 provides the mapping of status codes defined in DSP0200 to equivalent SOAP faults defined in DSP0226.

Table 10 – GetInstance Error Codes

Status Code	Equivalent SOAP Fault
CIM_ERR_ACCESS_DENIED	wsman:AccessDenied
CIM_ERR_INVALID_NAMESPACE	wsa:DestinationUnreachable
CIM_ERR_INVALID_PARAMETER	wsman:InvalidParameter
CIM_ERR_INVALID_CLASS	wsa:DestinationUnreachable
CIM_ERR_NOT_FOUND	wsa:DestinationUnreachable
CIM_ERR_FAILED	wsman:InternalError

16.1.2 DeleteInstance

The mapping defined in Table 11 shall be used for the DeleteInstance operation.

Table 11 – DeleteInstance

Operation	DeleteInstance
Operation target	CIM namespace
WS-Man operation	WS-Transfer:Delete or WS-Eventing:Unsubscribe (for CIM_IndicationSubscription and CIM_FilterCollectionSubscription)
WS-Man EPR	Class-specific ResourceURI with keys as selectors
Additional usage	None

Table 12 provides the mapping of the DeleteInstance arguments defined in Section 2.3.2.4 of DSP0200.

Table 12 – DeleteInstance Arguments

Argument	DeleteInstance
InstanceName	Mapped to WS-Man EPR

Table 13 provides the mapping of status codes defined in DSP0200 to equivalent SOAP faults defined in DSP0226.

Table 13 – DeleteInstance Error Codes

Status Code	Equivalent SOAP Fault
CIM_ERR_ACCESS_DENIED	wsman:AccessDenied
CIM_ERR_NOT_SUPPORTED (by the CIM Server for this operation)	wsa:ActionNotSupported
CIM_ERR_INVALID_NAMESPACE	wsa:DestinationUnreachable
CIM_ERR_INVALID_PARAMETER	wsman:InvalidParameter
CIM_ERR_INVALID_CLASS	wsa:DestinationUnreachable

Status Code	Equivalent SOAP Fault
CIM_ERR_NOT_SUPPORTED (This operation is not supported for the class of the specified instance, if provided.)	wsa:ActionNotSupported
CIM_ERR_NOT_FOUND	wsa:DestinationUnreachable
CIM_ERR_FAILED	wsman:InternalError

16.1.3 ModifyInstance

The mapping defined in Table 14 shall be used for the ModifyInstance operation.

Table 14 – ModifyInstance

Operation	ModifyInstance
Operation target	CIM namespace
WS-Man operation	WS-Transfer:Put or WS-Eventing:Renew (for CIM_IndicationSubscription and CIM_FilterCollectionSubscription)
WS-Man EPR	Class-specific ResourceURI with keys as selectors
Additional usage	None
Notes	As defined in R7.1-1, the class specified in the Resource URI needs to be the creation class of the instance it addresses.

Table 15 provides the mapping of the ModifyInstance arguments defined in Section 2.3.2.8 of DSP0200.

Table 15 – ModifyInstance Arguments

Argument	ModifyInstance
InstanceName	Mapped to WS-Man EPR
IncludeQualifier	There is no corresponding WS-Management parameter for this argument. From the expected behavior perspective, this argument is treated as always set to false.
PropertyList[ ]	Always set to NULL for the instances of CIM_IndicationSubscription and CIM_FilterCollectionSubscription. For instances of other classes: If it is NULL, then the operation is handled through WS-Transfer Put. If it is not NULL, then the operation is handled through fragment level WS-Transfer Put (see Section 4.10 of DSP0226).

Table 16 provides the mapping of status codes defined in DSP0200 to equivalent SOAP faults defined in DSP0226.

Table 16 – ModifyInstance Error Codes

Status Code	Equivalent SOAP Fault
CIM_ERR_ACCESS_DENIED	wsman:AccessDenied
CIM_ERR_NOT_SUPPORTED (by the CIM Server for this operation)	wsa:ActionNotSupported

Status Code	Equivalent SOAP Fault
CIM_ERR_INVALID_NAMESPACE	wsa:DestinationUnreachable
CIM_ERR_INVALID_PARAMETER	wsman:InvalidParameter
CIM_ERR_INVALID_CLASS	wsa:DestinationUnreachable
CIM_ERR_NOT_SUPPORTED (This operation is not supported for the class of the specified instance, if provided.)	wsa:ActionNotSupported
CIM_ERR_NOT_FOUND	wsa:DestinationUnreachable
CIM_ERR_FAILED	wsman:InternalError

16.1.4 CreateInstance

The mapping defined in Table 17 shall be used for the CreateInstance operation.

Table 17 – CreateInstance

Operation	CreateInstance
Operation target	CIM namespace
WS-Man operation	WS-Transfer:Create or WS-Eventing:Subscribe (for CIM_IndicationSubscription and CIM_FilterCollectionSubscription)
WS-Man EPR	Class-specific ResourceURI as factory, with only the __cimnamespace selector allowed
Additional usage	None
Notes	As defined in R7.1-1, the class specified in the Resource URI needs to be the creation class of the instance it addresses.

Table 18 provides the mapping of the CreateInstance arguments as defined in Section 2.3.2.6 of DSP0200.

Table 18 – CreateInstance Arguments

Argument	CreateInstance
InstanceName	Mapped to WS-Man EPR

Table 19 provides the mapping of status codes defined in DSP0200 to equivalent SOAP faults defined in DSP0226.

Table 19 – CreateInstance Error Codes

Status Code	Equivalent SOAP Fault
CIM_ERR_ACCESS_DENIED	wsman:AccessDenied
CIM_ERR_NOT_SUPPORTED (by the CIM Server for this operation)	wsa:ActionNotSupported
CIM_ERR_INVALID_NAMESPACE	wsa:DestinationUnreachable
CIM_ERR_INVALID_PARAMETER	wsman:InvalidParameter
CIM_ERR_INVALID_CLASS	wsa:DestinationUnreachable

Status Code	Equivalent SOAP Fault
CIM_ERR_NOT_SUPPORTED (This operation is not supported for the class of the specified instance, if provided.)	wsa:ActionNotSupported
CIM_ERR_ALREADY_EXISTS	wsman:AlreadyExists
CIM_ERR_FAILED	wsman:InternalError

16.1.5 EnumerateInstances

The mapping defined in Table 20 shall be used for the EnumerateInstances operation.

Table 20 – EnumerateInstances

Operation	EnumerateInstances
Operation target	CIM namespace
WS-Man operation	WS-Enumeration:Enumerate and WS-Enumeration:Pull (if needed)
WS-Man EPR	Class-specific ResourceURI with no selectors
Additional usage	Use wsman:EnumerationMode=EnumerateObjectAndEPR
Notes	

Table 21 provides the mapping of EnumerateInstances arguments as defined in Section 2.3.2.11 of DSP0200.

Table 21 – EnumerateInstances Arguments

Argument	EnumerateInstances
ClassName	Mapped to WS-Man EPR
LocalOnly	There is no corresponding WS-Management parameter for this argument. From the expected behavior perspective, this argument is treated as always set to false.
DeepInheritance	If true, then wsmb:PolymorphismMode modifier element value is set to IncludeSubClassProperties or wsmb:PolymorphismMode is not specified. If false, then wsmb:PolymorphismMode modifier element value is set to ExcludeSubClassProperties.
IncludeQualifier	There is no corresponding WS-Management parameter for this argument. From the expected behavior perspective, this argument is treated as always set to false.
IncludeClassOrigin	There is no corresponding WS-Management parameter for this argument. From the expected behavior perspective, this argument is treated as always set to false.
PropertyList[ ]	If it is NULL, then the operation is handled through WS-Enumeration. If it is not NULL, then the operation is handled through fragment-level enumerations (see Section 5.6 of DSP0226).

Table 22 provides the mapping of status codes defined in DSP0200 to equivalent SOAP faults defined in DSP0226.

Table 22 – EnumerateInstances Error Codes

Status Code	Equivalent SOAP Fault
CIM_ERR_ACCESS_DENIED	wsman:AccessDenied
CIM_ERR_NOT_SUPPORTED (by the CIM Server for this operation)	wsa:ActionNotSupported
CIM_ERR_INVALID_NAMESPACE	wsa:DestinationUnreachable
CIM_ERR_INVALID_PARAMETER	wsman:InvalidParameter
CIM_ERR_INVALID_CLASS	wsa:DestinationUnreachable
CIM_ERR_NOT_SUPPORTED (This operation is not supported for the class of the specified instance, if provided.)	wsa:ActionNotSupported
CIM_ERR_FAILED	wsman:InternalError

16.1.6 EnumerateInstanceNames

The mapping defined in Table 23 shall be used for the EnumerateInstanceNames operation.

Table 23 – EnumerateInstanceNames

Operation	EnumerateInstanceNames
Operation target	CIM namespace
WS-Man operation	WS-Enumeration:Enumerate and WS-Enumeration:Pull (if needed)
WS-Man EPR	Class-specific ResourceURI with no selectors
Additional usage	Use wsman:EnumerationMode=EnumerateEPR
Notes	

Table 24 provides the mapping of EnumerateInstanceNames arguments as defined in Section 2.3.2.12 of DSP0200.

Table 24 – EnumerateInstanceNames Arguments

Argument	EnumerateInstanceNames
ClassName	Mapped to WS-Man EPR

Table 25 provides the mapping of status codes defined in DSP0200 to equivalent SOAP faults defined in DSP0226.

Table 25 – EnumerateInstanceNames Error Codes

Status Code	Equivalent SOAP Fault
CIM_ERR_ACCESS_DENIED	wsman:AccessDenied
CIM_ERR_NOT_SUPPORTED (by the CIM Server for this operation)	wsa:ActionNotSupported
CIM_ERR_INVALID_NAMESPACE	wsa:DestinationUnreachable
CIM_ERR_INVALID_PARAMETER	wsman:InvalidParameter
CIM_ERR_INVALID_CLASS	wsa:DestinationUnreachable
CIM_ERR_NOT_SUPPORTED (This operation is not supported for the class of the specified instance, if provided.)	wsa:ActionNotSupported
CIM_ERR_FAILED	wsman:InternalError

16.1.7 Associators

The mapping defined in Table 26 shall be used for the Associators operation.

Table 26 – Associators

Operation	Associators
Operation target	CIM namespace
WS-Man operation	WS-Enumeration:Enumerate and WS-Enumeration:Pull (if needed)
WS-Man EPR	All-classes ResourceURI with no selectors
Additional usage	Use wsman:EnumerationMode=EnumerateObjectAndEPR Use the following association filter dialect with the wsmb:AssociatedInstances element: http://schemas.dmtf.org/wbem/wsman/1/cimbinding/associationFilter
Notes	

Table 27 provides the mapping of the Associators arguments as defined in Section 2.3.2.14 of DSP0200.

Table 27 – Associators Arguments

Argument	Associators
ObjectName	wsmb:Object value is set to ObjectName
AssocClass	If not NULL, wsmb:AssociationClassName value is set to AssocClass
ResultClass	If not NULL, wsmb:ResultClassName value is set to ResultClass
Role	If not NULL, wsmb:Role value is set to Role
ResultRole	If not NULL, wsmb:ResultRole value is set to ResultRole
IncludeQualifiers	There is no corresponding WS-Management parameter for this argument. From the expected behavior perspective, this argument is treated as always set to false.

Argument	Associators
IncludeClassOrigin	There is no corresponding WS-Management parameter for this argument. From the expected behavior perspective, this argument is treated as always set to false.
PropertyList[ ]	If it is NULL, then the operation is handled through WS-Enumeration. If it is not NULL, then the operation is handled through fragment-level enumerations (see Section 5.6 of DSP0226).

Table 28 provides the mapping of status codes defined in DSP0200 to equivalent SOAP faults defined in DSP0226.

Table 28 – Associators Error Codes

Status Code	Equivalent SOAP Fault
CIM_ERR_ACCESS_DENIED	wsman:AccessDenied
CIM_ERR_NOT_SUPPORTED (by the CIM Server for this operation)	wsa:ActionNotSupported
CIM_ERR_INVALID_NAMESPACE	wsa:DestinationUnreachable
CIM_ERR_INVALID_PARAMETER	wsman:InvalidParameter
CIM_ERR_NOT_SUPPORTED (This operation is not supported for the class of the specified instance, if provided.)	wsa:ActionNotSupported
CIM_ERR_FAILED	wsman:InternalError

16.1.8 AssociatorNames

The mapping defined in Table 29 shall be used for the AssociatorNames operation.

Table 29 – AssociatorNames

Operation	AssociatorNames
Operation target	CIM namespace
WS-Man operation	WS-Enumeration:Enumerate and WS-Enumeration:Pull (if needed)
WS-Man EPR	All-classes ResourceURI with no selectors
Additional usage	Use wsman:EnumerationMode=EnumerateEPR Use the following association filter dialect with the wsmb:AssociatedInstances element: http://schemas.dmtf.org/wbem/wsman/1/cimbinding/associationFilter
Notes	

Table 30 provides the mapping of the AssociatorNames arguments as defined in Section 2.3.2.15 of DSP0200.

Table 30 – AssociatorNames Arguments

Argument	AssociatorNames
ObjectName	wsmb:Object value is set to ObjectName
AssocClass	If not NULL, wsmb:AssociationClassName value is set to AssocClass
ResultClass	If not NULL, wsmb:ResultClassName value is set to ResultClass
Role	If not NULL, wsmb:Role value is set to Role
ResultRole	If not NULL, wsmb:ResultRole value is set to ResultRole

Table 31 provides the mapping of status codes as defined in DSP0200 to equivalent SOAP faults defined in DSP0226.

Table 31 – AssociatorNames Error Codes

Status Code	Equivalent SOAP Fault
CIM_ERR_ACCESS_DENIED	wsman:AccessDenied
CIM_ERR_NOT_SUPPORTED (by the CIM Server for this operation)	wsa:ActionNotSupported
CIM_ERR_INVALID_NAMESPACE	wsa:DestinationUnreachable
CIM_ERR_INVALID_PARAMETER	wsman:InvalidParameter
CIM_ERR_NOT_SUPPORTED (This operation is not supported for the class of the specified instance, if provided.)	wsa:ActionNotSupported
CIM_ERR_FAILED	wsman:InternalError

16.1.9 References

The mapping defined in Table 32 shall be used for the References operation.

Table 32 – References

Operation	References
Operation target	CIM namespace
WS-Man operation	WS-Enumeration:Enumerate and WS-Enumeration:Pull (if needed)
WS-Man EPR	All-classes ResourceURI with no selectors
Additional usage	Use wsman:EnumerationMode=EnumerateObjectAndEPR Use association the following filter dialect with the wsmb:AssociationInstances element: http://schemas.dmtf.org/wbem/wsman/1/cimbinding/associationFilter
Notes	

Table 33 provides the mapping of the References arguments as defined in Section 2.3.2.16 of DSP0200.

Table 33 – References Arguments

Argument	References
ObjectName	wsmb:Object value is set to ObjectName
ResultClass	If not NULL, wsmb:ResultClassName value is set to ResultClass
Role	If not NULL, wsmb:Role value is set to Role
IncludeQualifiers	There is no corresponding WS-Management parameter for this argument. From the expected behavior perspective, this argument is treated as always set to false.
IncludeClassOrigin	There is no corresponding WS-Management parameter for this argument. From the expected behavior perspective, this argument is treated as always set to false.
PropertyList[ ]	If it is NULL, then the operation is handled through WS-Enumeration. If it is not NULL, then the operation is handled through fragment-level enumerations (see Section 5.6 of DSP0226).

Table 34 provides the mapping of status codes as defined in DSP0200 to equivalent SOAP faults defined in DSP0226.

Table 34 – References Error Codes

Status Code	Equivalent SOAP Fault
CIM_ERR_ACCESS_DENIED	wsman:AccessDenied
CIM_ERR_NOT_SUPPORTED (by the CIM Server for this operation)	wsa:ActionNotSupported
CIM_ERR_INVALID_NAMESPACE	wsa:DestinationUnreachable
CIM_ERR_INVALID_PARAMETER	wsman:InvalidParameter
CIM_ERR_NOT_SUPPORTED (This operation is not supported for the class of the specified instance, if provided.)	wsa:ActionNotSupported
CIM_ERR_FAILED	wsman:InternalError

16.1.10 ReferenceNames

The mapping defined in Table 35 shall be used for the ReferenceNames operation.

Table 35 – ReferenceNames

Operation	ReferenceNames
Operation target	CIM namespace
WS-Man operation	WS-Enumeration:Enumerate and WS-Enumeration:Pull (if needed)
WS-Man EPR	All-classes ResourceURI with no selectors
Additional usage	Use wsman:EnumerationMode=EnumerateEPR Use association the following filter dialect with the wsmb:AssociationInstances element: http://schemas.dmtf.org/wbem/wsman/1/cimbinding/associationFilter
Notes	

Table 36 provides the mapping of the ReferenceNames arguments as defined in Section 2.3.2.17 of DSP0200.

Table 36 – ReferenceNames Arguments

Argument	ReferenceNames
ObjectName	wsmb:Object value is set to ObjectName
ResultClass	If not NULL, wsmb:ResultClassName value is set to ResultClass
Role	If not NULL, wsmb:Role value is set to Role

Table 37 provides the mapping of status codes as defined in DSP0200 to equivalent SOAP faults defined in DSP0226.

Table 37 – ReferenceNames Error Codes

Status Code	Equivalent SOAP Fault
CIM_ERR_ACCESS_DENIED	wsman:AccessDenied
CIM_ERR_NOT_SUPPORTED (by the CIM Server for this operation)	wsa:ActionNotSupported
CIM_ERR_INVALID_NAMESPACE	wsa:DestinationUnreachable
CIM_ERR_INVALID_PARAMETER	wsman:InvalidParameter
CIM_ERR_NOT_SUPPORTED (This operation is not supported for the class of the specified instance, if provided.)	wsa:ActionNotSupported
CIM_ERR_FAILED	wsman:InternalError

16.1.11 Pulled Enumerations

The following operations used in pulled enumerations are mapped in this section.

1) Open operations
    a) OpenEnumerateInstances
    b) OpenEnumerateInstancePaths
    c) OpenReferenceInstances
    d) OpenReferenceInstancePaths
    e) OpenAssociatorInstances
    f) OpenAssociatorInstancePaths
2) Pull operations
    a) PullInstancesWithPath
    b) PullInstancePaths
3) Other
    a) CloseEnumeration

16.1.11.1 OpenEnumerateInstances

The mapping defined in Table 38 shall be used for the OpenEnumerateInstances operation.

Table 38 – OpenEnumerateInstances

Operation	**OpenEnumerateInstances**
Operation target	CIM namespace
WS-Man operation	WS-Enumeration:Enumerate
WS-Man EPR	Class-specific ResourceURI with no selectors
Additional usage	Use wsman:EnumerationMode=EnumerateObjectAndEPR
Notes	

Table 39 provides the mapping of OpenEnumerateInstances arguments as defined in Section 5.3.2.24.3 of DSP0200.

Table 39 – OpenEnumerateInstances Arguments

Argument	**OpenEnumerateInstances**
EnumerationContext	Mapped to wsen:EnumerationContext
EndOfSequence	Mapped to wsman:EndOfSequence
ClassName	Mapped to WS-Man EPR
DeepInheritance	If true, then wsmb:PolymorphismMode modifier element value is set to IncludeSubClassProperties or wsmb:PolymorphismMode is not specified. If false, then wsmb:PolymorphismMode modifier element value is set to ExcludeSubClassProperties.
IncludeClassOrigin	There is no corresponding WS-Management parameter for this argument. From the expected behavior perspective, this argument is treated as always set to false.
PropertyList[ ]	If it is NULL, then the operation is handled through WS-Enumeration. If it is not NULL, then the operation is handled through fragment-level enumerations (see Section 5.6 of DSP0226).
FilterQueryLanguage	See section 8.3 of DSP0226
FilterQuery	See section 8.3 of DSP0226
OperationTimeOut	Mapped to wsen:Expires
ContinueOnError	There is no corresponding WS-Management parameter for this argument. From the expected behavior perspective, this argument is treated as always set to false.
MaxObjectCount	Mapped to wsman:MaxElements

Table 40 provides the mapping of status codes defined in DSP0200 to equivalent SOAP faults defined in DSP0226.

Table 40 – OpenEnumerateInstances Error Codes

Status Code	**Equivalent SOAP Fault**
CIM_ERR_ACCESS_DENIED	wsman:AccessDenied
CIM_ERR_SERVER_IS_SHUTTING_DOWN	wsman:InternalError
CIM_ERR_NOT_SUPPORTED	wsa:ActionNotSupported
CIM_ERR_INVALID_NAMESPACE	wsa:DestinationUnreachable

Status Code	Equivalent SOAP Fault
CIM_ERR_INVALID_OPERATION_TIMEOUT	wsen:InvalidExpirationTime
CIM_ERR_CONTINUATION_ON_ERROR_NOT_SUPPORTED	Not applicable.
CIM_ERR_INVALID_PARAMETER	wsman:InvalidParameter
CIM_ERR_INVALID_CLASS	wsa:DestinationUnreachable
CIM_ERR_FILTERED_ENUMERATION_NOT_SUPPORTED	wsen:FilteringNotSupported
CIM_ERR_QUERY_LANGUAGE_NOT_SUPPORTED	wsen:FilterDialectRequestedUnavailable
CIM_ERR_INVALID_QUERY	wsen:CannotProcessFilter
CIM_ERR_FAILED	wsman:InternalError

16.1.11.2 OpenEnumerateInstancePaths

The mapping defined in Table 41 shall be used for the OpenEnumerateInstancePaths operation.

Table 41 – OpenEnumerateInstancePaths

Operation	OpenEnumerateInstancePaths
Operation target	CIM namespace
WS-Man operation	WS-Enumeration:Enumerate
WS-Man EPR	Class-specific ResourceURI with no selectors
Additional usage	Use wsman:EnumerationMode=EnumerateEPR
Notes	

Table 42 provides the mapping of OpenEnumerateInstancePaths arguments as defined in Section 5.3.2.24.4 of DSP0200.

Table 42 – OpenEnumerateInstancePaths Arguments

Argument	OpenEnumerateInstancePaths
EnumerationContext	Mapped to wsen:EnumerationContext
EndOfSequence	Mapped to wsman:EndOfSequence
ClassName	Mapped to WS-Man EPR
FilterQueryLanguage	See section 8.3 of DSP0226
FilterQuery	See section 8.3 of DSP0226
OperationTimeOut	Mapped to wsen:Expires
ContinueOnError	There is no corresponding WS-Management parameter for this argument. From the expected behavior perspective, this argument is treated as always set to false.
MaxObjectCount	Mapped to wsman:MaxElements

Table 43 provides the mapping of status codes defined in DSP0200 to equivalent SOAP faults defined in DSP0226.

Table 43 – OpenEnumerateInstancePaths Error Codes

Status Code	Equivalent SOAP Fault
CIM_ERR_ACCESS_DENIED	wsman:AccessDenied
CIM_ERR_SERVER_IS_SHUTTING_DOWN	wsman:InternalError
CIM_ERR_NOT_SUPPORTED	wsa:ActionNotSupported
CIM_ERR_INVALID_NAMESPACE	wsa:DestinationUnreachable
CIM_ERR_INVALID_OPERATION_TIMEOUT	wsen:InvalidExpirationTime
CIM_ERR_CONTINUATION_ON_ERROR_NOT_SUPPORTED	Not applicable.
CIM_ERR_INVALID_PARAMETER	wsman:InvalidParameter
CIM_ERR_INVALID_CLASS	wsa:DestinationUnreachable
CIM_ERR_FILTERED_ENUMERATION_NOT_SUPPORTED	wsen:FilteringNotSupported
CIM_ERR_QUERY_LANGUAGE_NOT_SUPPORTED	wsen:FilterDialectRequestedUnavailable
CIM_ERR_INVALID_QUERY	wsen:CannotProcessFilter
CIM_ERR_FAILED	wsman:InternalError

16.1.11.3 OpenReferenceInstances

The mapping defined in Table 44 shall be used for the OpenReferenceInstances operation.

Table 44 – OpenReferenceInstances

Operation	OpenReferenceInstances
Operation target	CIM namespace
WS-Man operation	WS-Enumeration:Enumerate
WS-Man EPR	All-classes ResourceURI with no selectors
Additional usage	Use wsman:EnumerationMode=EnumerateObjectAndEPR Use association the following filter dialect with the wsmb:AssociationInstances element: http://schemas.dmtf.org/wbem/wsman/1/cimbinding/associationFilter
Notes	

Table 45 provides the mapping of OpenReferenceInstances arguments as defined in Section 5.3.2.24.5 of DSP0200.

Table 45 – OpenReferenceInstances Arguments

Argument	OpenReferenceInstances
EnumerationContext	Mapped to wsen:EnumerationContext
EndOfSequence	Mapped to wsman:EndOfSequence
InstanceName	wsmb:Object value is set to InstanceName

Argument	OpenReferenceInstances
ResultClass	If not NULL, wsmb:ResultClassName value is set to ResultClass
Role	If not NULL, wsmb:Role value is set to Role
IncludeClassOrigin	There is no corresponding WS-Management parameter for this argument. From the expected behavior perspective, this argument is treated as always set to false.
PropertyList[ ]	If it is NULL, then the operation is handled through WS-Enumeration. If it is not NULL, then the operation is handled through fragment-level enumerations (see Section 5.6 of DSP0226).
FilterQueryLanguage	See section 8.3 of DSP0226
FilterQuery	See section 8.3 of DSP0226
OperationTimeOut	Mapped to wsen:Expires
ContinueOnError	There is no corresponding WS-Management parameter for this argument. From the expected behavior perspective, this argument is treated as always set to false.
MaxObjectCount	Mapped to wsman:MaxElements

Table 46 provides the mapping of status codes defined in DSP0200 to equivalent SOAP faults defined in DSP0226.

Table 46 – OpenReferenceInstances Error Codes

Status Code	Equivalent SOAP Fault
CIM_ERR_ACCESS_DENIED	wsman:AccessDenied
CIM_ERR_SERVER_IS_SHUTTING_DOWN	wsman:InternalError
CIM_ERR_NOT_SUPPORTED	wsa:ActionNotSupported
CIM_ERR_INVALID_NAMESPACE	wsa:DestinationUnreachable
CIM_ERR_INVALID_OPERATION_TIMEOUT	wsen:InvalidExpirationTime
CIM_ERR_CONTINUATION_ON_ERROR_NOT_SUPPORTED	Not applicable.
CIM_ERR_INVALID_PARAMETER	wsman:InvalidParameter
CIM_ERR_NOT_FOUND	wsa:DestinationUnreachable
CIM_ERR_FILTERED_ENUMERATION_NOT_SUPPORTED	wsen:FilteringNotSupported
CIM_ERR_QUERY_LANGUAGE_NOT_SUPPORTED	wsen:FilterDialectRequestedUnavailable
CIM_ERR_INVALID_QUERY	wsen:CannotProcessFilter
CIM_ERR_FAILED	wsman:InternalError

16.1.11.4 OpenReferenceInstancePaths

The mapping defined in Table 47 shall be used for the OpenReferenceInstancePaths operation.

Table 47 – OpenReferenceInstancePaths

Operation	OpenReferenceInstancePaths
Operation target	CIM namespace
WS-Man operation	WS-Enumeration:Enumerate

Operation	OpenReferenceInstancePaths
WS-Man EPR	All-classes ResourceURI with no selectors
Additional usage	Use wsman:EnumerationMode=EnumerateEPR Use association the following filter dialect with the wsmb:AssociationInstances element: http://schemas.dmtf.org/wbem/wsman/1/cimbinding/associationFilter
Notes	

Table 48 provides the mapping of OpenReferenceInstancePaths arguments as defined in Section 5.3.2.24.6 of DSP0200.

Table 48 – OpenReferenceInstancePaths Arguments

Argument	OpenReferenceInstancePaths
EnumerationContext	Mapped to wsen:EnumerationContext
EndOfSequence	Mapped to wsman:EndOfSequence
InstanceName	wsmb:Object value is set to InstanceName
ResultClass	If not NULL, wsmb:ResultClassName value is set to ResultClass
Role	If not NULL, wsmb:Role value is set to Role
FilterQueryLanguage	See section 8.3 of DSP0226
FilterQuery	See section 8.3 of DSP0226
OperationTimeOut	Mapped to wsen:Expires
ContinueOnError	There is no corresponding WS-Management parameter for this argument. From the expected behavior perspective, this argument is treated as always set to false.
MaxObjectCount	Mapped to wsman:MaxElements

Table 49 provides the mapping of status codes defined in DSP0200 to equivalent SOAP faults defined in DSP0226.

Table 49 – OpenReferenceInstancePaths Error Codes

Status Code	Equivalent SOAP Fault
CIM_ERR_ACCESS_DENIED	wsman:AccessDenied
CIM_ERR_SERVER_IS_SHUTTING_DOWN	wsman:InternalError
CIM_ERR_NOT_SUPPORTED	wsa:ActionNotSupported
CIM_ERR_INVALID_NAMESPACE	wsa:DestinationUnreachable
CIM_ERR_INVALID_OPERATION_TIMEOUT	wsen:InvalidExpirationTime
CIM_ERR_CONTINUATION_ON_ERROR_NOT_SUPPORTED	Not applicable.
CIM_ERR_INVALID_PARAMETER	wsman:InvalidParameter
CIM_ERR_NOT_FOUND	wsa:DestinationUnreachable
CIM_ERR_FILTERED_ENUMERATION_NOT_SUPPORTED	wsen:FilteringNotSupported
CIM_ERR_QUERY_LANGUAGE_NOT_SUPPORTED	wsen:FilterDialectRequestedUnavailable

Status Code	Equivalent SOAP Fault
CIM_ERR_INVALID_QUERY	wsen:CannotProcessFilter
CIM_ERR_FAILED	wsman:InternalError

16.1.11.5 OpenAssociatorInstances

The mapping defined in Table 50 shall be used for the OpenAssociatorInstances operation.

Table 50 – OpenAssociatorInstances

Operation	OpenAssociatorInstances
Operation target	CIM namespace
WS-Man operation	WS-Enumeration:Enumerate
WS-Man EPR	All-classes ResourceURI with no selectors
Additional usage	Use wsman:EnumerationMode=EnumerateObjectAndEPR Use the following association filter dialect with the wsmb:AssociatedInstances element: http://schemas.dmtf.org/wbem/wsman/1/cimbinding/associationFilter
Notes	

Table 51 provides the mapping of OpenAssociatorInstances arguments as defined in Section 5.3.2.24.7 of DSP0200.

Table 51 – OpenAssociatorInstances Arguments

Argument	OpenAssociatorInstances
EnumerationContext	Mapped to wsen:EnumerationContext
EndOfSequence	Mapped to wsman:EndOfSequence
InstanceName	wsmb:Object value is set to InstanceName
AssocClass	If not NULL, wsmb:AssociationClassName value is set to AssocClass
ResultClass	If not NULL, wsmb:ResultClassName value is set to ResultClass
Role	If not NULL, wsmb:Role value is set to Role
ResultRole	If not NULL, wsmb:ResultRole value is set to ResultRole
IncludeClassOrigin	There is no corresponding WS-Management parameter for this argument. From the expected behavior perspective, this argument is treated as always set to false.
PropertyList[ ]	If it is NULL, then the operation is handled through WS-Enumeration. If it is not NULL, then the operation is handled through fragment-level enumerations (see Section 5.6 of DSP0226).
FilterQueryLanguage	See section 8.3 of DSP0226
FilterQuery	See section 8.3 of DSP0226
OperationTimeOut	Mapped to wsen:Expires
ContinueOnError	There is no corresponding WS-Management parameter for this argument. From the expected behavior perspective, this argument is treated as always set to false.
MaxObjectCount	Mapped to wsman:MaxElements

Table 52 provides the mapping of status codes defined in DSP0200 to equivalent SOAP faults defined in DSP0226.

Table 52 – OpenAssociatorInstances Error Codes

Status Code	Equivalent SOAP Fault
CIM_ERR_ACCESS_DENIED	wsman:AccessDenied
CIM_ERR_SERVER_IS_SHUTTING_DOWN	wsman:InternalError
CIM_ERR_NOT_SUPPORTED	wsa:ActionNotSupported
CIM_ERR_INVALID_NAMESPACE	wsa:DestinationUnreachable
CIM_ERR_INVALID_OPERATION_TIMEOUT	wsen:InvalidExpirationTime
CIM_ERR_CONTINUATION_ON_ERROR_NOT_SUPPORTED	Not applicable.
CIM_ERR_INVALID_PARAMETER	wsman:InvalidParameter
CIM_ERR_NOT_FOUND	wsa:DestinationUnreachable
CIM_ERR_FILTERED_ENUMERATION_NOT_SUPPORTED	wsen:FilteringNotSupported
CIM_ERR_QUERY_LANGUAGE_NOT_SUPPORTED	wsen:FilterDialectRequestedUnavailable
CIM_ERR_INVALID_QUERY	wsen:CannotProcessFilter
CIM_ERR_FAILED	wsman:InternalError

16.1.11.6 OpenAssociatorInstancePaths

The mapping defined in Table 53 shall be used for the OpenAssociatorInstancePaths operation.

Table 53 – OpenAssociatorInstancePaths

Operation	OpenAssociatorInstancePaths
Operation target	CIM namespace
WS-Man operation	WS-Enumeration:Enumerate
WS-Man EPR	All-classes ResourceURI with no selectors
Additional usage	Use wsman:EnumerationMode=EnumerateEPR Use the following association filter dialect with the wsmb:AssociatedInstances element: http://schemas.dmtf.org/wbem/wsman/1/cimbinding/associationFilter
Notes	

Table 54 provides the mapping of OpenAssociatorInstancePaths arguments as defined in Section 5.3.2.24.8 of DSP0200.

Table 54 – OpenAssociatorInstancePaths Arguments

Argument	OpenAssociatorInstancePaths
EnumerationContext	Mapped to wsen:EnumerationContext
EndOfSequence	Mapped to wsman:EndOfSequence
InstanceName	wsmb:Object value is set to InstanceName

Argument	OpenAssociatorInstancePaths
AssocClass	If not NULL, wsmb:AssociationClassName value is set to AssocClass
ResultClass	If not NULL, wsmb:ResultClassName value is set to ResultClass
Role	If not NULL, wsmb:Role value is set to Role
ResultRole	If not NULL, wsmb:ResultRole value is set to ResultRole
FilterQueryLanguage	See section 8.3 of DSP0226
FilterQuery	See section 8.3 of DSP0226
OperationTimeOut	Mapped to wsen:Expires
ContinueOnError	There is no corresponding WS-Management parameter for this argument. From the expected behavior perspective, this argument is treated as always set to false.
MaxObjectCount	Mapped to wsman:MaxElements

Table 55 provides the mapping of status codes defined in DSP0200 to equivalent SOAP faults defined in DSP0226.

Table 55 – OpenAssociatorInstancePaths Error Codes

Status Code	Equivalent SOAP Fault
CIM_ERR_ACCESS_DENIED	wsman:AccessDenied
CIM_ERR_SERVER_IS_SHUTTING_DOWN	wsman:InternalError
CIM_ERR_NOT_SUPPORTED	wsa:ActionNotSupported
CIM_ERR_INVALID_NAMESPACE	wsa:DestinationUnreachable
CIM_ERR_INVALID_OPERATION_TIMEOUT	wsen:InvalidExpirationTime
CIM_ERR_CONTINUATION_ON_ERROR_NOT_SUPPORTED	Not applicable.
CIM_ERR_INVALID_PARAMETER	wsman:InvalidParameter
CIM_ERR_NOT_FOUND	wsa:DestinationUnreachable
CIM_ERR_FILTERED_ENUMERATION_NOT_SUPPORTED	wsen:FilteringNotSupported
CIM_ERR_QUERY_LANGUAGE_NOT_SUPPORTED	wsen:FilterDialectRequestedUnavailable
CIM_ERR_INVALID_QUERY	wsen:CannotProcessFilter
CIM_ERR_FAILED	wsman:InternalError

16.1.11.7 PullInstancesWithPath

The mapping defined in Table 56 shall be used for the PullInstancesWithPath operation.

Table 56 – PullInstancesWithPath

Operation	PullInstancesWithPath
Operation target	CIM namespace
WS-Man operation	WS-Enumeration:Pull
WS-Man EPR	Class-specific ResourceURI with no selectors or All-classes ResourceURI with no selectors as specified in the corresponding WS-Enumeration:Enumerate operation

Operation	PullInstancesWithPath
Notes	The corresponding WS-Enumerate:Enumerate operation shall specify wsman:EnumerationMode=EnumerateObjectAndEPR and the association filter dialect with the appropriate element (if needed).

Table 57 provides the mapping of PullInstancesWithPath arguments as defined in Section 5.3.2.24.10 of DSP0200.

Table 57 – PullInstancesWithPath Arguments

Argument	PullInstancesWithPath
EnumerationContext	Mapped to wsen:EnumerationContext
EndOfSequence	Mapped to wsman:EndOfSequence
MaxObjectCount	Mapped to wsman:MaxElements

Table 58 provides the mapping of status codes defined in DSP0200 to equivalent SOAP faults defined in DSP0226.

Table 58 – PullInstancesWithPath Error Codes

Status Code	Equivalent SOAP Fault
CIM_ERR_ACCESS_DENIED	wsman:AccessDenied
CIM_ERR_SERVER_IS_SHUTTING_DOWN	wsman:InternalError
CIM_ERR_NOT_SUPPORTED	wsa:ActionNotSupported
CIM_ERR_INVALID_NAMESPACE	wsa:DestinationUnreachable
CIM_ERR_INVALID_PARAMETER	wsman:InvalidParameter
CIM_ERR_INVALID_ENUMERATION_CONTEXT	wsen:InvalidEnumerationContext
CIM_ERR_PULL_HAS_BEEN_ABANDONED	wsman:InternalError
CIM_ERR_SERVER_LIMITS_EXCEEDED	wsman:InternalError
CIM_ERR_FAILED	wsman:InternalError

16.1.11.8 PullInstancePaths

The mapping defined in Table 59 shall be used for the PullInstancePaths operation.

Table 59 – PullInstancePaths

Operation	PullInstancePaths
Operation target	CIM namespace
WS-Man operation	WS-Enumeration:Pull
WS-Man EPR	Class-specific ResourceURI with no selectors or All-classes ResourceURI with no selectors as specified in the corresponding WS-Enumeration:Enumerate operation
Notes	The corresponding WS-Enumerate:Enumerate operation shall specify wsman:EnumerationMode=EnumerateEPR and the association filter dialect with the appropriate element (if needed).

Table 60 provides the mapping of PullInstancePaths arguments as defined in Section 5.3.2.24.11 of DSP0200.

Table 60 – PullInstancePaths Arguments

Argument	PullInstancePaths
EnumerationContext	Mapped to wsen:EnumerationContext
EndOfSequence	Mapped to wsman:EndOfSequence
MaxObjectCount	Mapped to wsman:MaxElements

Table 61 provides the mapping of status codes defined in DSP0200 to equivalent SOAP faults defined in DSP0226.

Table 61 – PullInstancePaths Error Codes

Status Code	Equivalent SOAP Fault
CIM_ERR_ACCESS_DENIED	wsman:AccessDenied
CIM_ERR_SERVER_IS_SHUTTING_DOWN	wsman:InternalError
CIM_ERR_NOT_SUPPORTED	wsa:ActionNotSupported
CIM_ERR_INVALID_NAMESPACE	wsa:DestinationUnreachable
CIM_ERR_INVALID_PARAMETER	wsman:InvalidParameter
CIM_ERR_INVALID_ENUMERATION_CONTEXT	wsen:InvalidEnumerationContext
CIM_ERR_PULL_HAS_BEEN_ABANDONED	wsman:InternalError
CIM_ERR_SERVER_LIMITS_EXCEEDED	wsman:InternalError
CIM_ERR_FAILED	wsman:InternalError

16.1.11.9 CloseEnumeration

The mapping defined in Table 62 shall be used for the CloseEnumeration operation.

Table 62 – CloseEnumeration

Operation	CloseEnumeration
Operation target	CIM namespace
WS-Man operation	WS-Enumeration:Release

Table 63 provides the mapping of CloseEnumeration arguments as defined in Section 5.3.2.24.12 of DSP0200.

Table 63 – CloseEnumeration Arguments

Argument	CloseEnumeration
EnumerationContext	Mapped to wsen:EnumerationContext

Table 64 provides the mapping of status codes defined in DSP0200 to equivalent SOAP faults defined in DSP0226.

Table 64 – CloseEnumeration Error Codes

Status Code	Equivalent SOAP Fault
CIM_ERR_ACCESS_DENIED	wsman:AccessDenied
CIM_ERR_SERVER_IS_SHUTTING_DOWN	wsman:InternalError
CIM_ERR_NOT_SUPPORTED	wsa:ActionNotSupported
CIM_ERR_INVALID_NAMESPACE	wsa:DestinationUnreachable
CIM_ERR_INVALID_PARAMETER	wsman:InvalidParameter
CIM_ERR_INVALID_ENUMERATION_CONTEXT	wsen:InvalidEnumerationContext
CIM_ERR_PULL_CANNOT_BE_ABANDONED	wsman:InternalError
CIM_ERR_FAILED	wsman:InternalError

16.1.12 ExecQuery

This operation is supported for the CQL query language. See 8.1 for more details.

16.2 Unsupported Operations

This specification does not define equivalents for the following operations:

- GetClass
- DeleteClass
- CreateClass
- ModifyClass
- EnumerateClasses
- EnumerateClassNames
- GetProperty
- SetProperty
- GetQualifier
- SetQualifier
- DeleteQualifier
- EnumerateQualifiers
- OpenQueryInstances
- PullInstances
- EnumerationCount

17 Mapping of Error Messages to SOAP Fault Subcodes

Table 65 outlines suggested mappings of CIM error messages to corresponding subcodes to be used when returning SOAP faults.

Table 65 – CIM Error Messages with Corresponding Subcode Mappings

Message ID	Message Name	Fault Subcode
WIPG0201	Authentication failed	wsman:AccessDenied (Support may be transport-dependent.)
WIPG0202	Authorization failed	wsman:AccessDenied
WIPG0203	Operation not supported by CIM service infrastructure	wsa:ActionNotSupported
WIPG0204	CIM namespace not found	wsa:DestinationUnreachable
WIPG0205	Missing input parameter	wsmb:CIMException
WIPG0206	Duplicate input parameter	wsman:InvalidParameter
WIPG0207	Unknown input parameter	wsman:InvalidParameter
WIPG0208	Invalid input parameter value	wsman:InvalidParameter
WIPG0213	CIM instance not found	wsa:DestinationUnreachable
WIPG0214	CIM class not found	wsa:DestinationUnreachable
WIPG0216	CIM instance already exists	wsman:AlreadyExists
WIPG0218	No such CIM method	wsa:ActionNotSupported
WIPG0219	CIM method not supported by CIM class implementation	wsa:ActionNotSupported
WIPG0220	No such CIM property	wxf:InvalidRepresentation
WIPG0221	Unknown query language	wsen:FilterDialectRequestedUnavailable (if encountered while processing wsen:Enumerate) wsman:CannotProcessFilter (if encountered while processing wse:Subscribe)
WIPG0222	Query language feature not supported by WBEM service infrastructure	wsen:CannotProcessFilter (if encountered while processing wsen:Enumerate) wsman:CannotProcessFilter (for exceptions encountered while processing wse:Subscribe)
WIPG0223	Invalid query	wsen:CannotProcessFilter (if encountered while processing wsen:Enumerate) wsman:CannotProcessFilter (if encountered while processing wsen:Enumerate)
WIPG0227	Operation failure	wsman:InternalError
WIPG0228	Operation not supported by CIM class implementation	wsa:ActionNotSupported
WIPG0229	CIM method invocation not supported by WBEM service infrastructure	wsa:ActionNotSupported

18 XSD

A normative copy of the XML schemas (XML Schema Part 1, XML Schema Part 2) for this specification may be retrieved by resolving the XML namespace URIs for this specification (listed in clause 5).

19 WSDL

This specification does not define a normative WSDL document. While it is possible to define a generic WSDL document that can apply to all CIM classes, it does a disservice to developers who can provide a more specific WSDL document tailored to a specific CIM class.

R19-1: WSDL documents for a CIM class should include all WS-Transfer operations.

R19-2: WSDL documents for a CIM class or the query engine should include all WS-Enumeration operations.

R19-3: WSDL documents for a CIM class or the query engine should include all WS-Eventing operations.

R19-4: WSDL documents for a CIM class should include operations for all extrinsic methods defined by the class.

ANNEX A
(informative)

Change Log

Version	Date	Author	Description
1.0.0	2009-06-19	Rick Landau	DMTF Standard Release
1.0.0	2008-03-03	Hemal Shah	Draft Standard Release.