Document number: DSP0223

Date: 2013-10-22

Version: 1.0.2

Generic Operations

Document type: Specification

Document status: DMTF Standard

Document language: en-US

Contents

Figures

Tables

Foreword

The *Generic Operations* specification (DSP0223) was prepared by the Generic Operations Working Group of the DMTF and is now owned by the Architecture Working Group of the DMTF.

DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems management and interoperability. For information about the DMTF, see http://www.dmtf.org.

Acknowledgements

DMTF acknowledges the following individuals for their contributions to this specification:

- Andreas Maier, IBM (editor)
- Jim Davis, WBEM Solutions
- George Ericson, EMC
- Steve Hand, Symantec
- Jon Hass, Dell
- Lawrence Lamers, VMware

Document conventions

Typographical conventions

The following typographical conventions are used in this document:

- The titles of referenced documents are marked in *italics*.
- Important terms that are used for the first time are marked in *italics*.
- Generic parameters and generic types are marked in *italics*.
- The usage of terms typically links to their definition. Example: class path
- XML text is in `monospaced font`.

Experimental material

Experimental material has yet to receive sufficient review to satisfy the adoption requirements set forth by the DMTF. Experimental material is included in this document as an aid to implementers who are interested in likely future developments. Experimental material may change as implementation experience is gained. It is likely that experimental material will be included in an upcoming revision of the document. Until that time, experimental material is purely informational.

The following typographical convention indicates experimental material:

EXPERIMENTAL

Experimental material appears here.

EXPERIMENTAL

In places where this typographical convention cannot be used (for example, tables or figures), the "EXPERIMENTAL" label is used alone.

Generic Operations

1 Scope

WBEM is a set of DMTF standards that define how CIM modeled resources can be discovered, accessed and manipulated. DMTF defines a number of WBEM protocols for this purpose:

- CIM-XML: The protocol defined in the CIM Operations over HTTP Specification DSP0200, the Representation of CIM in XML Specification DSP0201 and the DTD for Representation of CIM in XML DSP0203.
- WS-MAN: The usage of the WS-Management protocol for CIM, as defined in the WS-Management CIM Binding Specification DSP0227, the WS-CIM Mapping Specification DSP0230, the Web Services for Management Specification DSP0226, and other underlying Web Services specifications.
- SM-CLP: The protocol defined in the Server Management Command Line Protocol Specification DSP0214, covering the core of the protocol common for all management profiles, and SM-CLP mapping specifications for each management profile, covering profile specific aspects of the protocol such as verbs for extrinsic methods.

As different as these protocols are, they have certain operations and semantics in common, at least when looking at it from a higher level. These common semantics can be used to define generic operations. This specification defines the model and behavior associated to these operations at a generic level, and common across the WBEM protocols.

The generic operations are expected to be used in the following areas:

- Future releases of CIM management profile specifications can define the support for intrinsic operations by referencing generic operations. Currently, they do that by referencing the operations defined for the CIM-XML protocol. Using generic operations allows the management profile specifications to become independent of protocols.
- Future and existing WBEM protocols can define their operations conformant to the generic operations. This drives more commonality across these protocols, and consequently makes it easier to support multiple protocols in client applications, server side instrumentation, and mapping bridges between protocols.
- Client APIs, server APIs and provider APIs can define their API calls conformant to the generic operations. This drives more commonality across these APIs and between these APIs and WBEM protocols, and consequently makes it easier to support multiple protocols with the same API in client libraries and server side instrumentation (e.g., provider APIs).

2 Normative references

The following referenced documents are indispensable for the application of this specification. For dated or versioned references, only the edition cited (including any corrigenda or DMTF update versions) applies. For references without a date or version, the latest published edition of the referenced document (including any corrigenda or DMTF update versions) applies.

DMTF DSP0004, *CIM Infrastructure Specification 2.6*,
http://www.dmtf.org/standards/published_documents/DSP0004_2.6.pdf

DMTF DSP0207, *WBEM URI Mapping 1.0*,
http://www.dmtf.org/standards/published_documents/DSP0207_1.0.pdf

DMTF DSP0228, *Message Registry XML Schema 1.1*,
http://schemas.dmtf.org/wbem/messageregistry/1/dsp0228_1.1.xsd

DMTF DSP8016, *WBEM Operations Message Registry 1.0*,
http://schemas.dmtf.org/wbem/messageregistry/1/dsp8016_1.0.xml

ISO/IEC Directives, Part 2:2004, *Rules for the structure and drafting of International Standards*,
http://isotc.iso.org/livelink/livelink?func=ll&objId=4230456&objAction=browse

3 Terms and definitions

In this specification, some terms have a specific meaning beyond the normal English meaning. Those terms are defined in this clause.

The terms "shall" ("required"), "shall not", "should" ("recommended"), "should not" ("not recommended"), "may", "need not" ("not required"), "can" and "cannot" in this specification are to be interpreted as described in ISO/IEC Directives, Part 2, Annex H. The terms in parenthesis are alternatives for the preceding term, for use in exceptional cases when the preceding term cannot be used for linguistic reasons. ISO/IEC Directives, Part 2, Annex H specifies additional alternatives. Occurrences of such additional alternatives shall be interpreted in their normal English meaning.

The terms "clause", "subclause", "paragraph", "annex" in this specification are to be interpreted as described in ISO/IEC Directives, Part 2, Clause 5.

The terms "normative" and "informative" in this specification are to be interpreted as described in ISO/IEC Directives, Part 2, Clause 3. In this specification, clauses, subclauses or annexes indicated with "(informative)" as well as notes and examples do not contain normative content.

The terms defined in DSP0004 apply to this specification. The following additional terms are used in this document.

3.1
class path
a special kind of object path addressing a CIM class that is accessible through a WBEM server
For details, see DSP0004.

3.2
creation class
the creation class of a CIM instance is the most derived class the instance is of
For a complete definition, see DSP0004.

3.3
duplicate object
objects in a result set that have duplicate object paths

3.4
duplicate object path
object paths in a result set that reference the same CIM object accessible through the WBEM server

3.5
effective qualifier value
The effective value of a qualifier specified on a schema element is the value that determines the qualifier behavior for the schema element, taking the qualifier propagation rules into account. For a complete definition, see DSP0004.

3.6

exposed elements of a class

The set of schema elements exposed by a class (i.e., properties and methods) is the union of the set of elements defined in the class and the set of inherited elements that are not overridden in the class. For a complete definition, see DSP0004.

3.7

generic operation

a generic operation as defined in this specification

3.8

generic operations mapping

a mapping of generic operations to the operations of some other protocol (e.g., WBEM operations) or to the calls of some API, as defined in 5.2

3.9

instance path

a special kind of object path addressing a CIM instance that is accessible through a WBEM server
For details, see DSP0004.

3.10

isolation

the set of behaviors that describe how the execution of an operation affects the execution of another, concurrent operation, as defined in 5.8.4

3.11

management profile

a management profile as defined in DSP1001
As used in this specification, the term includes all possible owners of such profiles, including other standards organizations and vendors.

3.12

namespace path

a special kind of object path addressing a CIM namespace that is accessible through a WBEM server
For details, see DSP0004.

3.13

object

a class, instance, qualifier type or namespace that is accessible through a WBEM server
For details, see DSP0004.

3.14

object path

the address of an object that is accessible through a WBEM server
For details, see DSP0004.

3.15

qualifier type path

a special kind of object path addressing a CIM qualifier type that is accessible through a WBEM server
For details, see DSP0004.

3.16

volatile property

a property in a CIM instance whose value may change as a WBEM client obtains the instance repeatedly without performing any client originated updates to the property value

3.17

WBEM client

a CIM client (see DSP0004) that supports a WBEM protocol

A WBEM client originates WBEM operations for processing by a WBEM server. This definition does not imply any particular implementation architecture or scope, such as a client library component or an entire management application. For details, see 5.1.

3.18

WBEM indication

an interaction within a WBEM protocol that is originated on a WBEM server and processed by a WBEM listener

This release of this specification does not cover WBEM indications.

3.19

WBEM listener

a CIM listener (see DSP0004) that supports a WBEM protocol

A WBEM listener processes WBEM indications originated by a WBEM server. This definition does not imply any particular implementation architecture or scope, such as a standalone demon component or an entire management application.

This release of this specification does not cover WBEM listeners.

3.20

WBEM operation

an interaction within a WBEM protocol that is originated by a WBEM client and processed by a WBEM server

For details, see 5.1.

3.21

WBEM protocol

a communications protocol between WBEM client, WBEM server and WBEM listener

A WBEM protocol defines how the WBEM operations and WBEM indications work, on top of an underlying protocol layer (for example, HTTP, SOAP, or TCP). For details, see 5.1.

3.22

WBEM protocol mapping

a mapping of generic operations to a WBEM protocol, as defined in 5.2

3.23

WBEM server

a CIM server (see DSP0004) that supports a WBEM protocol

A WBEM server processes WBEM operations originated by a WBEM client, and originates WBEM indications for processing by a WBEM listener. This definition does not imply any particular implementation architecture, such as a separation into a CIMOM and provider components. For details, see 5.1.

4 Symbols and abbreviated terms

The symbols and abbreviations defined in DSP0004 apply to this specification. The following additional abbreviations are used in this document.

4.1
API
Application Programming Interface

4.2
CIM
Common Information Model, defined by DMTF

4.3
CQL
CIM Query Language, defined in DSP0202

4.4
HTTP
Hyper Text Transfer Protocol, defined by W3C

4.5
UML
Unified Modeling Language, defined by OMG

4.6
WBEM
Web Based Enterprise Management, defined by DMTF

4.7
XML
Extensible Markup Language, defined by W3C

5 Concepts

This clause defines concepts that are the basis for the definition of the generic operations.

5.1 Generic operations model

Figure 1 shows the generic operations model using a UML sequence diagram:

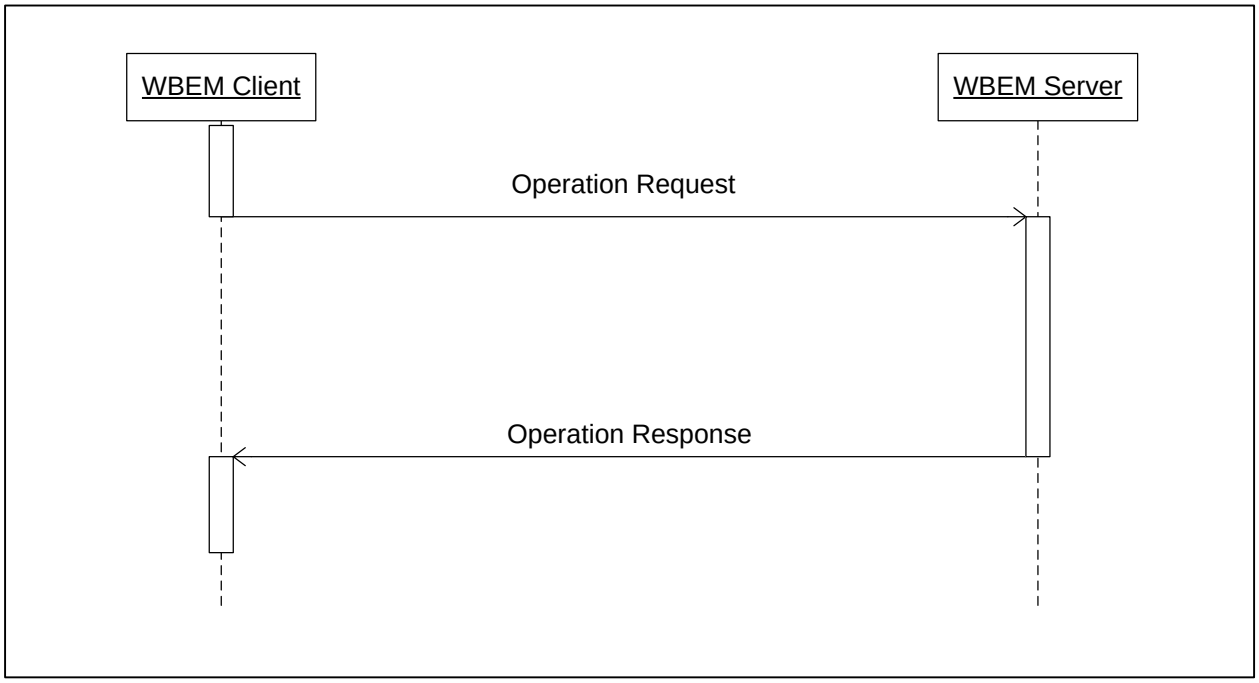

Figure 1 – Generic operations model

In the generic operations model, *operations* are logical actions directed from a WBEM client to a WBEM server. An *operation request* is sent from the client to the service when invoking the operation and an *operation response* is sent back from the service to the client upon completion of the operation.

At the level of generic operations, any *input parameters* are part of the operation request, and any *output parameters* are part of the operation response. A WBEM protocol may choose to do that differently, for example by pushing some of the input parameters to the service side in the form of options that are set, and that are used during the processing of subsequent operations.

The operation request and operation response at the level of generic operations do not necessarily need to correspond directly to messages that are flowing at the level of the WBEM protocol. For example, the operation response may be delivered asynchronously at the level of the WBEM protocol.

This abstraction of generic operations from WBEM operations allows keeping the definition of the generic operations simple and scoped to defining the operation semantics. The details about the actual message flows are left to the scope of WBEM protocols. This separation is key in order to use the same definition of generic operations for multiple WBEM protocols.

5.2 Generic operations mappings

5.2.1 Overview

Figure 2 shows mappings of generic operations to WBEM protocols and APIs. These mappings allow determining which WBEM operations or API calls need to be implemented for a particular generic operation to be supported. This is used for example when implementing management profiles that specify provisions for intrinsic operations by referencing generic operations.

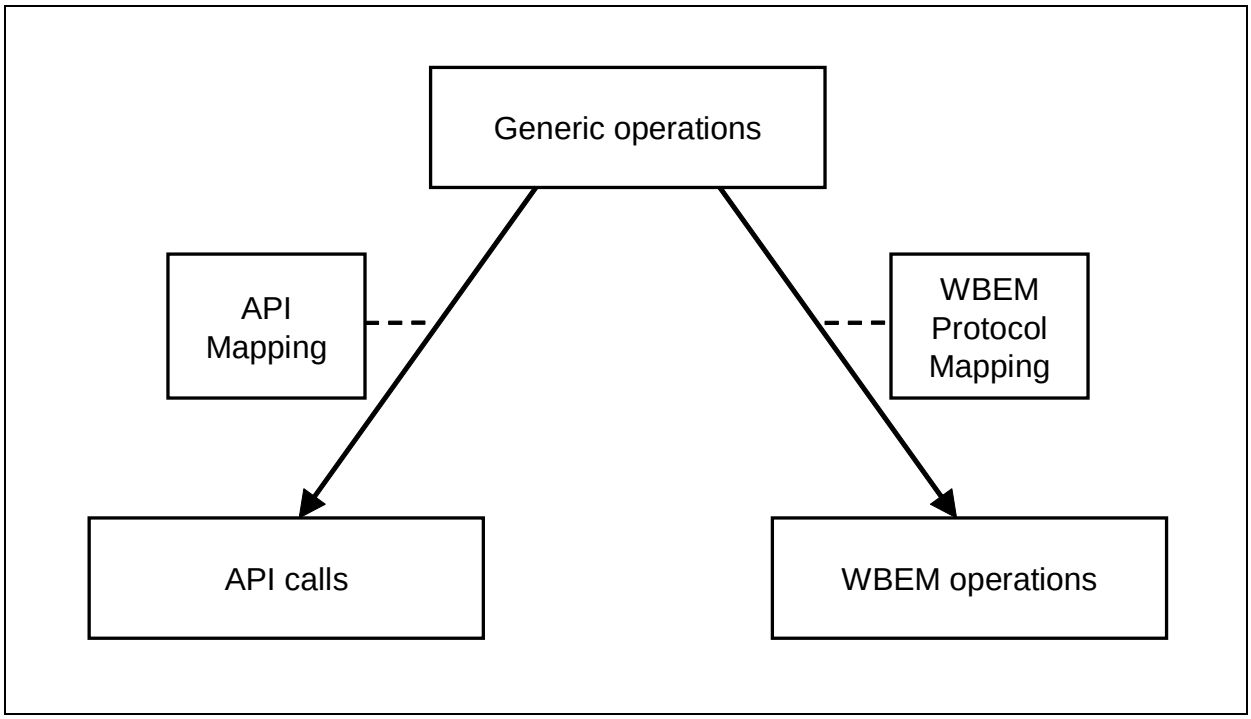

Figure 2 – Generic operations mappings

5.2.2 Recommendations

This subclause provides recommendations for specifying WBEM protocol mappings and API mappings that provide for determining the WBEM operations or API calls that support a particular generic operation, and specify conformance.

There is no requirement that WBEM protocol mappings and API mappings are defined in a separate specification (i.e., they can be defined in the specifications that define the WBEM protocol or API).

The following recommendations apply:

- WBEM protocol mappings and API mappings should define the mapping from a perspective of the generic operation (i.e., by listing the relevant generic operation at the top level).
- For each generic operation listed in the mapping, the corresponding WBEM operations or API calls should be stated that provide the functionality supporting the generic operation.
- For each parameter defined for a generic operation listed in the mapping, the corresponding parameters and return values of the WBEM operations or API calls should be stated.
- A statement should be made for each generic operation as to whether or not the operation is supported in a conformant way, as defined in 5.3.2. If the operation is supported in a non-conformant way, the deviations should be stated.
- A statement should be made for the entire WBEM protocol or API as to whether or not it is conformant to generic operations.

5.3 Conformance to generic operations

Conformance to generic operations is defined at two levels:

1) At the level of the entire WBEM protocol or API

2) At the level of single WBEM operations or single API calls

The guiding principle for conformance to generic operations is that a WBEM protocol or API call is able to completely represent the generic operations and their semantics. Functionalities of the WBEM protocol or API that go beyond the functionality of generic operations are not relevant for conformance.

5.3.1 Conformance of WBEM protocols or APIs

A WBEM protocol or API is conformant to generic operations if all generic operations defined in this specification are supported by WBEM operations or API calls in a conformant way.

Conformant WBEM protocols or APIs may define WBEM operations or API calls in addition to those that are mapped to generic operations.

5.3.2 Conformance of WBEM operations or API calls

A generic operation is supported by WBEM operations or API calls in a conformant way if all of the following is satisfied:

- The generic operation has one or more corresponding WBEM operations or API calls that provide the functionality of the generic operation. The names of these corresponding WBEM operations or API calls may be different from the name of the generic operation.
- Functionalities that are required to be supported for a generic operation are supported by the corresponding WBEM operations or API calls with the semantics defined by the generic operation.
- If functionalities that are optional to be supported for a generic operation are supported by the corresponding WBEM operations or API calls, they are supported with the semantics defined by the generic operation.
- Each parameter of a generic operation is mapped to one or more corresponding parameters of the corresponding WBEM operations or API calls
- For each parameter of a generic operation, the provisions defined in 5.3.3 are satisfied.

WBEM operations or API calls that support a generic operation in a conformant way, may support parameters or return values in addition to the parameters mapped to parameters of the corresponding generic operation. Defining additional parameters can affect interoperability between WBEM protocols.

5.3.3 Requirement levels for operation parameters

The parameters defined for generic operations each have a requirement level, as defined in this subclause. That requirement level defines whether a conformant WBEM protocol or API has to support the parameter.

The allowable requirement levels for parameters of generic operations are:

Mandatory

Operation parameters designated as mandatory shall be supported by conformant WBEM protocols or APIs with the semantics defined for the generic operation. Conformant WBEM protocols or APIs may define that supplying values for the corresponding parameters is optional if a default behavior is specified.

Conditional

Operation parameters designated as conditional shall be supported by conformant WBEM protocols or APIs if the specified condition is met. If supported, they shall be supported as

468 defined for the generic operation. Conformant WBEM protocols or APIs may define that
469 supplying values for the corresponding parameters is optional if a default behavior is specified.

470 **Optional**

471 Operation parameters designated as optional may be supported by conformant WBEM
472 protocols or APIs. If supported, they shall be supported as defined for the generic operation.
473 Conformant WBEM protocols or APIs may define that supplying values for the corresponding
474 parameters is optional if a default behavior is specified.

475 NOTE: Conformant WBEM protocols or APIs may specify that supplying values for a supported parameter is optional
476 as long as the protocol or API defines a default value for the parameter. In other words, there are two different kinds
477 of requirements related to parameters:

478 1. The requirement to support a parameter in a WBEM protocol or API as defined by its requirement level

479 2. The requirement defined by the WBEM protocol or API for supplying a value for a supported parameter when
480 invoking an operation

481 ## 5.4 Generic types

482 This specification defines the following generic data types for use by operation parameters of generic
483 operations.

484 ### 5.4.1 CIM data types

485 All CIM data types defined in DSP0004 (e.g., boolean) may be used as generic types. Values of these
486 data types can assume the (untyped) value NULL, as defined in DSP0004.

487 ### 5.4.2 NamespacePath

488 A value of the generic type *NamespacePath* represents a namespace path as defined in DSP0004.

489 Conformant WBEM protocols shall support all characteristics of *NamespacePath* values and may support
490 additional characteristics.

491 ### 5.4.3 InstancePath

492 A value of the generic type *InstancePath* represents an instance path as defined in DSP0004.

493 Conformant WBEM protocols shall support all characteristics of *InstancePath* values and may support
494 additional characteristics.

495 An instance path as defined in DSP0004 allows identifying the name of the creation class of the instance,
496 as well as the names and values of the key properties of the instance.

497 ### 5.4.4 ClassPath

498 A value of the generic type *ClassPath* represents a class path as defined in DSP0004.

499 Conformant WBEM protocols shall support all characteristics of *ClassPath* values and may support
500 additional characteristics.

501 ### 5.4.5 QualifierTypePath

502 A value of the generic type *QualifierTypePath* represents a qualifier type path as defined in DSP0004.

503 Conformant WBEM protocols shall support all characteristics of *ClassPath* values may support additional
504 characteristics.

5.4.6 InstanceSpecification

A value of the generic type *InstanceSpecification* is a representation of a CIM instance as defined for the *Instance* meta-element defined in DSP0004, containing:

- name of the creation class of the instance
- all or a subset of the static and non-static properties exposed by the creation class of the instance

Each property in an *InstanceSpecification* shall contain:

- name of the property
- value of the property
- optional: Class origin of the property
- optional: Data type of the property

InstanceSpecification does not contain the instance path of the CIM instance, because there are some situations in which the instance data is needed without an instance path. The *InstanceSpecificationWithPath* type is used when the instance path is needed in addition to the instance data.

Generic operations using this type define the rules for the optional items in the content of this type.

5.4.7 ClassSpecification

A value of the generic type *ClassSpecification* is a representation of a CIM class as defined for the *Class* meta-element defined in DSP0004, containing:

- name of the class
- name of the superclass, if any
- all or a subset of the static and non-static properties (that is, the property definitions) exposed by the class. As defined in DSP0004, the set of properties exposed by a class includes any properties inherited from superclasses, where overridden properties are included only once.
- all of the static and non-static methods exposed by the class. As defined in DSP0004, the set of methods exposed by a class includes any methods inherited from superclasses, where overridden methods are included only once.
- optional: all of the qualifiers exposed by the class that are defined on the class or any of its superclasses

Each property in a *ClassSpecification* shall contain:

- name of the property
- data type of the property
- default value of the property
- optional: all of the qualifiers exposed by the property that are defined on the property or any of its overridden properties

Each method in a *ClassSpecification* shall contain:

- name of the method
- data type of the return value of the method
- all of the parameters of the method
- optional: all of the qualifiers exposed by the method that are defined on the method or any of its overridden methods

Each parameter in that method shall contain:

- name of the parameter
- data type of the parameter
- optional: all of the qualifiers exposed by the parameter that are defined on the parameter or the corresponding parameter in any of its overridden methods

Each qualifier in any of the items above shall contain:

- name of the qualifier
- effective value of the qualifier, as seen in the scope of the class represented by *Class*

ClassSpecification does not contain the class path of the CIM class. The *ClassSpecificationWithPath* type is used when the class path is needed in addition to the class.

Generic operations using this type define the rules for the optional items in the content of this type.

5.4.8 QualifierType

A value of the generic type *QualifierType* is a representation of a CIM qualifier type as defined for the *QualifierType* meta-element defined in DSP0004 (i.e., a qualifier declaration) containing:

- name of the qualifier
- data type of the qualifier
- default value of the qualifier
- all flavors of the qualifier
- all scopes of the qualifier

QualifierType does not contain the qualifier type path of the CIM qualifier type. The *QualifierTypeWithPath* type is used when the qualifier type path is needed in addition to the qualifier type.

5.4.9 InstanceSpecificationWithPath

A value of the generic type *InstanceSpecificationWithPath* combines the content of *InstanceSpecification* and *InstancePath*.

InstanceSpecification shall represent the CIM instance referenced by *InstancePath*.

5.4.10 ClassSpecificationWithPath

A value of the generic type *ClassSpecificationWithPath* combines the content of *ClassSpecification* and *ClassPath*.

ClassSpecification shall represent the CIM class referenced by *ClassPath*.

5.4.11 QualifierTypeWithPath

A value of the generic type *QualifierTypeWithPath* combines the content of *QualifierType* and *QualifierTypePath*.

QualifierType shall represent the CIM qualifier type referenced by *QualifierTypePath*.

5.4.12 ClassName

A value of the generic type *ClassName* is the name of a CIM class, including its schema prefix.

5.4.13 PropertyName

A value of the generic type *PropertyName* is the name of a CIM property or reference.

The class defining the property is not identified by the data in this type.

5.4.14 MethodName

A value of the generic type *MethodName* is the name of a CIM method.

The class defining the method and the method signature are not identified by the data in this type.

5.4.15 ParameterValue

A value of the generic type *ParameterValue* is a parameter value used as an input or output parameter during invocation of a CIM method, containing:

- name of the parameter
- value of the parameter
- optional: Data type of the parameter

Generic operations using this type define the rules for the optional items in the content of this type.

5.4.16 ReturnValue

A value of the generic type *ReturnValue* is the value returned by the invocation of a CIM method, containing:

- return value
- optional: Data type of the return value

Generic operations using this type define the rules for the optional items in the content of this type.

5.4.17 QueryString

A value of the generic type *QueryString* is a query string in some query language. The query language is not identified by the data in this type.

5.4.18 QueryLanguage

A value of the generic type *QueryLanguage* is a query language of a query string.

5.4.19 EnumerationContext

A value of the generic type *EnumerationContext* is a value that uniquely identifies an enumeration session used in pulled instance enumeration operations. It is opaque to WBEM clients.

5.5 Success and failure

All generic operations either succeed or fail. There is no concept of "partial success".

If a generic operation succeeds, it delivers its output data back to the operation requester, and does not include any error messages.

If it fails, it delivers back one or more error messages, and no output data. For details about error messages, see 5.7.

For example, if an instance enumeration operation were able to return some instances successfully, but not all successfully, then the operation shall fail without returning any instances.

The WBEM operations mapped to generic operations by a conformant WBEM protocol shall also either succeed or fail, as described above.

5.6 Preconditions and postconditions

Each generic operation specifies a set of zero or more preconditions and a set of zero or more postconditions.

Each precondition in the set needs to be satisfied for the operation to be able to succeed. If one or more preconditions are not satisfied, the operation shall fail, indicating the unsatisfied precondition using a generic error message from the set listed for the operation that describes the unsatisfied precondition.

A successful execution of the generic operation shall guarantee that all postconditions in the set are satisfied.

5.7 Generic error messages

Each generic operation specifies a set of generic error messages. These generic error messages are DMTF standard messages (see DSP0228) from the WBEM Operations Message Registry (DSP8016). Each error message from this registry describes a particular error situation.

A conformant WBEM protocol shall support error handling in one or more of the following ways and shall state in its WBEM protocol mapping which ways are supported:

- If the WBEM protocol supports returning DMTF standard messages as part of a failure, then for each of its WBEM operations to which a generic operation was mapped, the WBEM operation shall return the generic error message defined for the generic operation that matches the error situation. The WBEM operation may return additional error messages.
- If the WBEM protocol supports returning CIM status codes as part of a failure, then for each of its WBEM operations to which a generic operation was mapped, the WBEM operation shall return the CIM status code stated in the generic error message defined for the generic operation that matches the error situation. The CIM status code values are stated in the definition of each message in DSP8016.
- Otherwise, the WBEM protocol mapping shall state for each of its WBEM operations to which a generic operation was mapped, to which of its protocol specific error conditions each generic error message corresponds that is defined by the generic operation.

The generic error messages specified for each generic operation have a requirement level defined in context of that operation. The requirement level defines whether a conformant WBEM protocol has to support the generic error message (in one or more of the ways defined above).

The allowable requirement levels for generic error messages in the context of a generic operation are:

Mandatory

Generic error messages designated as mandatory shall be supported by conformant WBEM protocols if applicable to the WBEM protocol. They shall be supported as defined in the description of the message.

Conditional

Generic error messages designated as conditional shall be supported by conformant WBEM protocols if the specified condition is met and if applicable to the WBEM protocol. If supported, they shall be supported as defined in the description of the message.

Optional

Generic error messages designated as optional may be supported by conformant WBEM protocols if applicable to the WBEM protocol. If supported, they shall be supported as defined in the description of the message.

Each generic operation designates one of its input parameters to be a "context parameter." The messages defined in the WBEM Operations Message Registry (DSP8016)) may include name and value of the context parameter in order to provide information about the invocation context.

This specification does not define any order or precedence for generic error messages to be returned by generic operations. This implies that the order in which the generic error messages are listed in the description of each generic operation has no binding significance on the order in which a conformant WBEM protocol would need to apply any tests to surface these errors, nor does the documented order require a precedence of error messages. However, the order in which the generic error messages are listed is meant to give some guidance about a typical order of precedence.

WBEM clients shall be prepared to deal with all generic error messages that are listed for a generic operation.

5.8 Consistency model

This subclause defines consistency requirements for generic operations.

Conformant WBEM protocols shall conform to the rules defined in this subclause for the WBEM operations to which the supported generic operations are mapped. WBEM protocols may define additional constraints for WBEM operations.

This specification does not define responsibilities for detecting violations to these rules.

5.8.1 Definition of ACID properties

This subclause defines atomicity, consistency, isolation and durability (ACID) properties for use by generic operations defined in this specification and by management profiles (see DSP1001).

Each generic operation defines requirements on its ACID properties. Management profiles that use generic operations to state their operation requirements inherit these requirements on ACID properties and may specify additional requirements. Profiles should not remove or weaken requirements on ACID properties defined by generic operations.

5.8.1.1 Atomicity

Operations and methods are considered *atomic* if and only if their effects on the managed environment and on CIM instances either occur completely or not at all.

Atomicity only applies to operations and methods that modify the managed environment or CIM instances through the management interface.

5.8.1.2 Update consistency

Operations and methods are considered *update-consistent* if and only if the managed environment and CIM instances are never left in an inconsistent state after a modification.

What constitutes a consistent state is defined in DSP0004 and in management profiles.

Update consistency only applies to operations and methods that modify the managed environment or CIM instances through the management interface.

5.8.1.3 Isolation

Operations and methods are considered *isolated* if and only if their results and their effects on the managed environment and on CIM instances appear to be serialized with the results and effects of any other operations and methods, as observed through the management interface.

Isolation applies to operations and methods that retrieve information through the management interface, and to operations that modify the managed environment or CIM instances through the management interface.

5.8.1.4 Durability

Operations and methods are considered *durable* if and only if their effects on the managed environment and on CIM instances will not be undone, other than by some other action that may or may not be caused through the profile defined management interface.

Durability only applies to operations and methods that modify the managed environment or CIM instances through the management interface.

5.8.2 Time consistency within a CIM instance

The property values of an instance returned by any generic operation shall represent a snapshot of the instance in the CIM namespace at some point in time.

If a WBEM protocol provides the capability to transfer an operation response in multiple parts, and a response that contains an instance is distributed over multiple parts which are transferred at different points in times, the property values of a particular CIM instance still need to satisfy the time consistency constraint.

5.8.3 Staleness of information returned

Conformant WBEM protocols should define that implementations should do a best effort to return the most current information, as far as property values of instances and also the existence of instances are concerned.

5.8.4 Isolation between operations

This specification defines no particular requirements regarding isolation between operations in addition to the other consistency rules defined in 5.8.

For example, if a CIM instance is deleted and after that another one is created, an enumeration operation executed concurrently may consistently include the instance that got deleted just before that happened, as well as the new instance after it got consistently created, hence returning a set of instances that never existed at the same time. This example satisfies all consistency rules defined in this specification.

An example where other consistency rules determine the overall behavior is a GetInstance operation executing concurrently with a ModifyInstance operation on the same instance. The consistency rules defined in 5.8.2 require that this GetInstance operation needs to return an instance that either has none or all of the modifications requested by the ModifyInstance operation.

5.8.5 Duplicate return of CIM objects or object paths

Any generic operations returning CIM object specifications or CIM object paths should not return duplicate objects or duplicate object paths.

If duplicate objects or duplicate object paths are returned, WBEM clients should consider the last occurrence of a duplicate object or duplicate object path in the sequence as the valid occurrence to work with, and should ignore all other duplicate occurrences.

DSP0004 requires that a CIM namespace in a WBEM server does not contain duplicate objects (i.e., instances, classes, qualifier types) at any point in time. However, given the rule above, the result set of a generic operation may.

An example for a situation in which duplicate instances or instance paths might be returned is a sequence of instance deletion and creation with the same key values concurrently to an enumeration operation, all in the same CIM namespace.

As a consequence, a WBEM server is not obliged to test for, correct or reject any duplicate objects or object paths in the result set of an operation.

5.8.6 Time consistency between returned CIM objects

This specification does not mandate any time consistency between the CIM objects or CIM object paths returned by generic operations.

For example, if a WBEM server processes an instance enumeration operation by contacting multiple independent infrastructure components each of which contributes instances to the combined result set, the result set may contain instances that represent different points in time.

However, the rule defined in 5.8.2 requires that consistency is maintained within each single CIM instance.

5.8.7 Order of returned CIM objects

For operations that do not support the specification of a sort order, the order of returned CIM objects is implementation dependent.

For example, if a WBEM server processes an instance enumeration operation by contacting multiple independent infrastructure components each of which contributes instances to the combined result set, the resulting order might be an arbitrary merge of the sequences of instances contributed by each component.

WBEM protocols may define additional requirements on the order of returned CIM objects.

5.8.8 Validity of returned object paths

This specification does not mandate that object paths returned to a WBEM client are still valid by the time the WBEM client attempts to use them in subsequent operations in order to address those objects.

For example: if a WBEM server returns an instance path and an operation then deletes the instance, a subsequent attempt to get the instance using the returned instance path will fail.

5.8.9 Effects of deleting an instance

Deleting an instance may affect the overall consistency because other instances depend on the instance to be deleted. Instances that depend on the instance to be deleted are called "dependent instances" in this specification.

The behavior of operations that delete instances (such as *DeleteInstance*) cannot be defined in a generally applicable way. The following options are available for defining the handling of the deletion of an instance in the presence of dependent instances (e.g., in management profiles or in the CIM schema):

- **Delete propagation:** Delete any dependent instances implicitly along with the instance to be deleted.

  Specifications using this specification need to give particular consideration to circular dependencies when defining rules for propagating deletion.

  NOTE: Such dependent instances may reside in a different CIM namespace (which may reside in a different WBEM server) than the instance to be deleted.

- **Rejection:** Reject the deletion of the instance to be deleted, leaving it to the WBEM client to delete dependent instances first.

The following options are **not** available for defining the handling of the deletion of an instance in the presence of dependent instances:

- **Deletion without propagation:** Delete the instance to be deleted but do not delete any dependent instances. This causes an inconsistent state in the model, so it has not been used for the following types of dependencies.

The following instances are considered dependent instances for this purpose:

- **Composition:** Instances associated to an instance to be deleted, via a composition where the instance to be deleted is on the aggregate side.

  The definition of the *Composition* qualifier in DSP0004 requires that this case is handled by propagating the deletion of the aggregate instance to any aggregated instances and their composition instances.

- **Key propagation:** Instances of classes that have propagated keys (key properties exposing a value of TRUE for the *Propagated* qualifier, i.e., weak instances) are considered dependents of the instance from which the keys propagate (i.e., the strong instance).

  The definition of the *Propagated* qualifier in DSP0004 requires that this case is handled by propagating the deletion of the strong instance to any weak instances and their association instances.

- **Referencing associations:** Association instances that reference the instance to be deleted.

  This case shall be handled with any or a combination of the following options:

  – by propagating the deletion of the referenced instance to its referencing association instance

  – by rejecting the deletion of the referenced instance to be deleted.

- **Qualifier defined delete propagation:** Instances to be deleted as a result of *IfDelete* and *Delete* qualifiers, as defined in DSP0004.

  Support of the *IfDelete* and *Delete* qualifiers by a WBEM server is optional, as defined in DSP0004.

This concept can be used to propagate deletion from an instance to its referencing association instance, from an association instance to its referenced instances, and in combination also between associated instances.

The definition of the *IfDelete* and *Delete* qualifiers in DSP0004 requires that this case is handled by propagating the deletion of an instance to which the *IfDelete* qualifier applies, to any instances to which the corresponding *Delete* qualifier applies.

- **Multiplicity underflow:** Instances associated to an instance to be deleted via an association with a minimum multiplicity (as defined with *Min* qualifier in the schema, or as constrained by management profiles) larger than 0 on the reference to the instance to be deleted, if the deletion would violate the minimum multiplicity that is required.

  EXAMPLE: Association AB references class A with *Min (2)* and references class B. Therefore, each instance of B is supposed to be associated via AB with least two instances of A. If an instance of A is to be deleted, and there is only one other instance of A associated to the instance of B that is associated with the instance of A to be deleted, the minimum multiplicity would be violated by the deletion.

  This case shall be handled with any or a combination of the following options:

  – by propagating the deletion of the instance to be deleted to its associated instance defining the multiplicity constraint, and the association instance.

  – by rejecting the original deletion.

6 Generic operations

This clause defines the generic operations. They are listed in Table 1, grouped by their headings.

Table 1 – List of generic operations

Group	Generic Operation	Description
Instance operations	GetInstance	See 6.3.1
	DeleteInstance	See 6.3.2
	ModifyInstance	See 6.3.3
	CreateInstance	See 6.3.4
Direct instance enumeration operations	EnumerateInstances	See 6.4.1
	EnumerateInstanceNames	See 6.4.2
	Associators	See 6.4.3
	AssociatorNames	See 6.4.4
	References	See 6.4.5
	ReferenceNames	See 6.4.6
Pulled instance enumeration operations	OpenEnumerateInstances	See 6.5.3
	OpenEnumerateInstancePaths	See 6.5.4
	OpenAssociators	See 6.5.5
	OpenAssociatorPaths	See 6.5.6
	OpenReferences	See 6.5.7
	OpenReferencePaths	See 6.5.8
	OpenQueryInstances	See 6.5.9
	PullInstancesWithPath	See 6.5.11
	PullInstancePaths	See 6.5.12
	PullInstances	See 6.5.13
	CloseEnumeration	See 6.5.14
	EnumerationCount	See 6.5.15
Method invocation operations	InvokeMethod	See 6.6.1
	InvokeStaticMethod	See 6.6.2
Class operations	GetClass	See 6.7.1
	DeleteClass	See 6.7.2
	ModifyClass	See 6.7.3
	CreateClass	See 6.7.4

Group	Generic Operation	Description
Class enumeration operations	EnumerateClasses	See 6.8.1
	EnumerateClassNames	See 6.8.2
	AssociatorClasses	See 6.8.3
	AssociatorClassPaths	See 6.8.4
	ReferenceClasses	See 6.8.5
	ReferenceClassPaths	See 6.8.6
Qualifier type operations	GetQualifierType	See 6.9.1
	DeleteQualifierType	See 6.9.2
	ModifyQualifierType	See 6.9.3
	CreateQualifierType	See 6.9.4
	EnumerateQualifierTypes	See 6.9.5

6.1 Description format

The generic operations are described using the following format. Items in angle brackets (e.g., "<name>") need to be replaced by some other text, as described further down in this subclause.

Purpose:

<Short description of the purpose of the operation.>

Operation Input Parameters:

Generic Name	Generic Type	Requirement	Description
<diname>	<ditype>	<direq>	<Description of the operation parameter, including any conditions for requirement level Conditional> <The text "(Context Parameter)" for the parameter that is supposed to be displayed in messages, as defined in 5.7>
…	…	…	…

Operation Output Parameters:

Generic Name	Generic Type	Requirement	Description
<diname>	<ditype>	<direq>	<Description of the operation parameter, including any conditions for requirement level Conditional>
…	…	…	…

Description:

<A detailed description of the semantics of the operation including all conditions and behaviors except those listed under Preconditions and Postconditions>

Preconditions:

- <List of additional preconditions for the operation, in plain text. Preconditions pertain to the state before an operation gets invoked. They have nothing to do with the execution of the operation or any effects the operation causes. They represent the conditions that are required to be met in order for the operation to have a chance to execute successfully. Although not required for preconditions, this specification uses "*shall*" to specify preconditions.>

Postconditions:

- <List of additional postconditions for the operation, in plain text. Postconditions describe the state after an operation has been executed successfully. In other words, they represent the guarantees an implementation needs to give in the case of successful execution.>

Error messages:

Message ID	Message Name	Requirement	Sources	Additional Description
<msgid>	<msgname>	<msgreq>	<msgsrc>	<Any description in addition to the description in the message registry>
...	...	...		...

The items in angle brackets that are not already described in the format above, have the following meaning:

<diname> Generic name of the operation parameter.

<ditype> Generic type of the operation parameter, as defined in 5.4.

<direq> Requirement level of the operation parameter, as defined in 5.3.3.

<msgid> Message ID of the message, as defined in a DMTF message registry. The message ID is the concatenation of the values of the XML attributes `MESSAGE/MESSAGE_ID@PREFIX` and `MESSAGE/MESSAGE_ID@SEQUENCE_NUMBER`.

<msgname> Message name of the message, as defined in a DMTF message registry. The message name is the value of the XML attribute `MESSAGE@NAME`.

<msgreq> Requirement level of the message, as defined in 5.7.

<msgsrc> Sources of the message. One or more values may be specified. Valid values are:

Infrastructure – the message is implemented by the common infrastructure portion of the WBEM server.

Class implem. – the message is implemented by the class specific portion of the WBEM server.

The message sources information is a recommendation only, for implementations of a WBEM server that distinguish between a common infrastructure portion (e.g., CIMOM) and class specific portion (e.g., providers).

6.2 Common operation parameters for all operations

This subclause defines commonly used operation parameters for the operations. The description of the individual operations references these operation parameters as appropriate. However, not every operation uses every one of these operation parameters.

6.2.1 IncludeClassOrigin

The *IncludeClassOrigin* operation input parameter controls whether class origin information is returned for any element in any returned object. Class origin information indicates which class defines the element.

Support for the *IncludeClassOrigin* operation parameter is conditional on support in the WBEM protocol for client side control of returning class origin information.

If the WBEM protocol does not support client side control of returning class origin information, then the *IncludeClassOrigin* operation parameter shall not be supported and class origin information shall be included for any element in any object returned by the operation.

If the WBEM protocol supports client side control of returning class origin information, then the *IncludeClassOrigin* operation parameter shall be supported. If the *IncludeClassOrigin* operation parameter is TRUE, then class origin information shall be included for any element in any object returned by the operation. If the *IncludeClassOrigin* operation parameter is FALSE, then class origin information shall not be included for any element in any object returned by the operation.

For operations returning instances, the elements are properties only (more precisely, their values). For operations returning classes, the elements are properties and methods (more precisely, their definitions).

6.2.2 IncludeQualifiers

The *IncludeQualifiers* operation input parameter controls whether qualifier values are returned for any returned CIM element in any returned class of a class operation.

Support for the *IncludeQualifiers* operation parameter in a conformant WBEM protocol is mandatory.

If *IncludeQualifiers* is TRUE, then any returned class and any returned CIM element within each returned class shall contain qualifier values for those qualifiers that have a value different from the default value defined in the declaration of the qualifier type. Any other qualifier values should not be included.

NOTE: In order to inspect the scope and default value of any qualifiers that are not included in the returned class, a WBEM client can use operation *EnumerateQualifierTypes* to retrieve the qualifier type declarations that exist in a namespace.

If *IncludeQualifiers* is FALSE, then any returned class and any returned CIM element within each returned class shall not contain any qualifier values.

6.2.3 <element>List

The operation output parameters *InstanceList*, *InstancePathList*, *ClassList*, *ClassPathList*, and *QualifierTypeList* contain a sequence of elements, and are referred to as the *result set* of the operation.

The sequence is ordered in the sense that there is a relation of "before" and "after" between elements in the sequence and the sequence has a beginning and an end. However, this does not imply that the sequence is sorted according to some criteria.

Clause 5.8 defines rules for dealing with duplicate objects or duplicate object paths in the result set of an operation.

6.3 Instance operations

This subclause defines instance operations (operations that target a single CIM instance, or create a CIM instance).

6.3.1 GetInstance

Purpose:

Retrieves a CIM instance.

Operation Input Parameters:

Generic Name	Generic Type	Requirement	Description
InstancePath	InstancePath	Mandatory	Instance path of the instance to be retrieved (Context Parameter)
IncludeClassOrigin	boolean	Conditional	Indicates whether class origin information for any returned properties is to be included, as defined in 6.2.1 Condition: WBEM protocol supports client side control of returning class origin information
IncludedProperties	PropertyName [ ]	Optional	NULL, or unordered set of property names, acting as a restricting filter on the properties included in the returned instance

Operation Output Parameters:

Generic Name	Generic Type	Requirement	Description
Instance	InstanceSpecification	Mandatory	Representation of the retrieved instance

Description:

The *GetInstance* operation retrieves a representation of the instance referenced by *InstancePath*.

As defined in the description of the *InstancePath* type, the instance path of the instance to be retrieved is interpreted in a non-polymorphic way, i.e., it references the specified instance only and does not include any instances with the same key values in subclasses.

The set of properties to be included in the retrieved instance shall be determined using the following algorithm:

- Initially, the set of properties to be included is the set of properties exposed by the creation class of the instance. This includes all the duplicates of any duplicate non-overridden properties.
- If the *IncludedProperties* operation input parameter is supported by the WBEM protocol and if its value is not NULL, it acts as a restricting filter on the properties to be included in the returned instance such that any properties exposed by the creation class of the instance that are not named in that operation parameter are removed from the set of properties to be included. Any duplicate or invalid property names in the *IncludedProperties* operation input parameter shall be ignored. A non-NULL empty *IncludedProperties* list removes all properties from the set of properties to be included.

- Conformant WBEM protocols may specify rules that cause properties with a value of NULL to be removed from the set of properties to be included.

Preconditions:

- The instance referenced by *InstancePath* shall exist in the namespace. If this is not satisfied, the operation shall fail, indicating WIPG0213.

Postconditions:

- The instance shall have been returned with the properties as defined in the Description paragraph for this operation.
- Requirements on ACID properties:
  - Atomicity: N/A
  - Update Consistency: N/A
  - Isolation: Required
  - Durability: N/A

Error messages:

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure, class implem.	
WIPG0214	Class not found	Mandatory	Infrastructure	
WIPG0228	Operation not supported by class implementation	Mandatory	Class implem.	
WIPG0213	Instance not found	Mandatory	Class implem.	
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

6.3.2 DeleteInstance

Purpose:

Deletes a CIM instance.

Operation Input Parameters:

Generic Name	Generic Type	Requirement	Description
InstancePath	InstancePath	Mandatory	Instance path of the instance to be deleted (Context Parameter)

Operation Output Parameters:

None.

Description:

The *DeleteInstance* operation deletes the instance referenced by *InstancePath*.

The existence of other CIM instances may depend on the instance to be deleted. There are multiple types of dependent instances, and multiple options to handle such dependent instances, as defined in 5.8.9.

NOTE: Any dependent instances that are deleted may reside in a different CIM namespace (which may reside in a different WBEM server) than the instance referenced by *InstancePath*.

In case of error, the consistency requirements defined in DSP0004 cannot be guaranteed, but should be attempted to be satisfied in a best effort approach. Such an approach may be to delete non-dependent instances first. In case of error, only a subset of the instances to be deleted may have been deleted, but each instance shall have either been deleted completely or not at all.

The effects of the deletion of any CIM instances on any underlying resources shall be defined elsewhere. For example, a management profile may define that the lifecycle of the CIM instance is coupled with the lifecycle of some underlying resource, and that this resource shall be deleted when the instance is deleted.

Preconditions:

- The instance referenced by *InstancePath* shall exist in the namespace. If this is not satisfied, the operation shall fail, indicating WIPG0213.

Postconditions:

- The instance referenced by *InstancePath* shall have been deleted.
- Any implicit deletions of dependent CIM instances shall have happened, as defined in 5.8.9.
- Any effects of the deletion of all of these CIM instances on any underlying resources shall have happened.
- The consistency requirements defined in DSP0004 shall be satisfied for any instances related to the deleted instances.
- Requirements on ACID properties:
  - Atomicity: Required, if dependent instances are handled by rejection, as defined in 5.8.9. Recommended, if dependent instances are handled by delete propagation, as defined in 5.8.9.

989 – Update Consistency: Required, if dependent instances are handled by rejection, as defined
990 in 5.8.9. Recommended, if dependent instances are handled by delete propagation, as
991 defined in 5.8.9.

992 – Isolation: Required, if dependent instances are handled by rejection, as defined in 5.8.9.
993 Recommended, if dependent instances are handled by delete propagation, as defined in
994 5.8.9.

995 – Durability: Required.

996 **Error messages:**
997

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure, class implem.	
WIPG0214	Class not found	Mandatory	Infrastructure	
WIPG0228	Operation not supported by class implementation	Mandatory	Class implem.	
WIPG0213	Instance not found	Mandatory	Class implem.	
WIPG0246	Instance cannot be deleted due to referencing association	Optional	Class implem.	
WIPG0247	Instance cannot be deleted due to multiplicity underflow	Optional	Class implem.	
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

6.3.3 ModifyInstance

Purpose:

Changes property values of a CIM instance.

Operation Input Parameters:

Generic Name	Generic Type	Requirement	Description
InstancePath	InstancePath	Mandatory	Instance path of the instance to be modified (Context Parameter)
ModifiedInstance	InstanceSpecification	Mandatory	Representation of the modified instance, specifying the new property values
IncludedProperties	PropertyName [ ]	Optional	NULL, or unordered set of property names, acting as a restricting filter on the properties to be modified

Operation Output Parameters:

None.

Description:

The *ModifyInstance* operation changes property values of the instance referenced by *InstancePath*.

The set of properties to be changed shall be determined using the following algorithm:

- Initially, the set of properties to be changed is the set of properties specified in *ModifiedInstance*.
- If the *IncludedProperties* operation input parameter is supported by the WBEM protocol and if its value is not NULL, it acts as a restricting filter on the properties to be changed such that any properties exposed by the creation class of the instance that are not named in that operation parameter are removed from the set of properties to be changed. Any duplicate or invalid property names in the *IncludedProperties* operation input parameter shall be ignored. A non-NULL empty *IncludedProperties* list removes all properties from that set.
- Any key properties and non-modifiable properties are removed from the set of properties to be changed. As a result, specifying such properties in *ModifiedInstance* or *IncludedProperties* does not cause an error.

NOTE: The modifiability of properties can be defined in the schema and in management profiles.

Conformant WBEM protocols may restrict *ModifiedInstance* to specify all properties exposed by the creation class of the instance referenced by *InstancePath*.

Preconditions:

- The instance referenced by *InstancePath* shall exist in the namespace. If this is not satisfied, the operation shall fail, indicating WIPG0213.
- The creation class of *ModifiedInstance* shall be the creation class of the instance referenced by *InstancePath* or a superclass of that class. If this is not satisfied, the operation shall fail, indicating WIPG0208.

- Any properties specified in *ModifiedInstance* shall be from the set of properties exposed by the creation class of *ModifiedInstance*. If this is not satisfied, the operation shall fail, indicating WIPG0208.

Postconditions:

- The values of the properties shall have been modified as defined in the Description paragraph for this operation.
- The values of key properties and non-modifiable properties shall not have been modified.
- Other properties may have changed as a result of side effects of changing properties, behavior defined in referencing specifications, or volatility of properties.
- The consistency requirements defined in DSP0004 shall be satisfied for the modified instance.
- Requirements on ACID properties:
  - Atomicity: Required
  - Update Consistency: Required
  - Isolation: Required
  - Durability: Required

Error messages:

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure, class implem.	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure, class implem.	
WIPG0214	Class not found	Mandatory	Infrastructure	
WIPG0228	Operation not supported by class implementation	Mandatory	Class implem.	
WIPG0213	Instance not found	Mandatory	Class implem.	
WIPG0220	No such property	Mandatory	Class implem.	
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

6.3.4 CreateInstance

Purpose:

Creates a CIM instance.

Operation Input Parameters:

Generic Name	Generic Type	Requirement	Description
ClassPath	ClassPath	Mandatory	Class path of CIM class specifying namespace and creation class for the instance to be created (Context Parameter)
NewInstance	InstanceSpecification	Optional	Instance specifying the initial property values for the instance to be created

Operation Output Parameters:

Generic Name	Generic Type	Requirement	Description
InstancePath	InstancePath	Mandatory	Instance path of the new instance

Description:

The *CreateInstance* operation creates a CIM instance in the namespace specified in *ClassPath* from the creation class specified in *ClassPath*, and returns the instance path of the new instance.

As defined in the description of the *ClassPath* type, the class path of the CIM class to be used as a creation class for the instance is interpreted in a non-polymorphic way, i.e., it references the specified class only and not any subclasses. In other words, the instance is created from the specified class only. As a result, the specified class becomes the creation class of the instance.

The newly created instance shall have all properties exposed by the creation class specified in *ClassPath*.

For each property, the initial value shall be determined as defined in the following **default** rules:

- If the *NewInstance* operation input parameter is supported, and if the property is included in *NewInstance*, its value is used. That is also the case if that value is NULL.
- Else, if a default value is declared for the property, that value is used.

These default rules allow specifying key properties and non-writeable properties in *NewInstance*. In other words, the creation of an instance does not have the restrictions a subsequent modification has.

As defined in DSP1001, management profiles may specify any such rules, overriding these default rules. This may result in rejecting, respecting or replacing the values of any properties specified in *NewInstance*, as well as respecting or replacing the default values of any properties not specified in *NewInstance.*

Volatile properties may change their values immediately after the instance has been created.

Instance creation based upon input data other than initial property values can be done using CIM methods. For example, creation of an instance of *CIM_ComputerSystem* representing a virtual computer system could be done using a *CreateVirtualComputerSystem( )* method taking a higher-level specification of the virtual computer system as input.

Other CIM instances may come into existence implicitly during the course of processing the *CreateInstance* operation. As defined in DSP1001, management profiles may specify the rules for such implicitly created instances.

Any such implicitly created instances may reside in a different CIM namespace (which may reside in a different WBEM server) than the namespace specified in *ClassPath*.

In case of error, the consistency requirements defined in DSP0004 should be attempted to be satisfied in a best effort approach. In case of error, only a subset of the instances to be created may have been created, but each instance shall have either been created completely or not at all.

As defined in DSP1001, management profiles may specify the effects of the creation of CIM instances on their underlying resources. For example, a management profile may define that the lifecycle of the CIM instance is coupled with the lifecycle of some underlying resource, and that this resource shall be created when the instance is created.

Preconditions:

- The instance to be created shall not exist in the namespace specified by *ClassPath*. If this is not satisfied, the operation shall fail, indicating WIPG0216.
- The CIM class referenced by *ClassPath* shall exist in the namespace. If this is not satisfied, the operation shall fail, indicating WIPG0214.
- The creation class of *NewInstance* shall be the class referenced by *ClassPath* or a superclass of that class. If this is not satisfied, the operation shall fail, indicating WIPG0208.
- Any properties specified in *NewInstance* shall be from the set of properties exposed by the class referenced by *ClassPath*. If this is not satisfied, the operation shall fail, indicating WIPG0208.

Postconditions:

- The instance shall have been created as defined in the Description paragraph for this operation.
- Any management profile defined implicit creations of other CIM instances shall have happened.
- Any management profile defined effects of the creation of all of these CIM instances on any underlying resources shall have happened.
- Requirements on ACID properties:
  - Atomicity: Required
  - Update Consistency: Required
  - Isolation: Required
  - Durability: Required

Error messages:

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service infrastructure	Mandatory	Infrastructure	

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure, class implem.	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure, class implem.	
WIPG0214	Class not found	Mandatory	Infrastructure	
WIPG0228	Operation not supported by class implementation	Mandatory	Class implem.	
WIPG0216	Instance already exists	Mandatory	Class implem.	
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

6.4 Direct instance enumeration operations

This subclause defines direct instance enumeration operations (operations that enumerate CIM instances and return them directly as a result of the operation).

6.4.1 EnumerateInstances

Purpose:

Enumerate the CIM instances of a class and return these instances.

Operation Input Parameters:

Generic Name	Generic Type	Requirement	Description
EnumClassPath	ClassPath	Mandatory	Class path of CIM class used for the enumeration (Context Parameter)
IncludeClassOrigin	boolean	Conditional	Indicates whether class origin information for any returned properties is to be included, as defined in 6.2.1 Condition: WBEM protocol supports client side control of returning class origin information
IncludedProperties	PropertyName [ ]	Optional	NULL, or unordered set of property names to be included, acting as a restricting filter on the properties included in the returned instances
ExcludeSubclass-Properties	boolean	Optional	Indicates whether properties added by subclasses of the specified class are to be excluded, acting as a restricting filter on the properties included in the returned instances

Operation Output Parameters:

Generic Name	Generic Type	Requirement	Description
InstanceList	InstanceSpecificationWithPath [ ]	Mandatory	Sequence of the enumerated instances with their instance paths

Description:

The *EnumerateInstances* operation enumerates all CIM instances of the class referenced by *EnumClassPath*, including instances of any of its subclasses, and returns these instances together with their instance paths.

All of the instances returned shall exist in the same namespace as the class referenced by *EnumClassPath*.

An instance is included in the result set if and only if it exists in the namespace specified in *EnumClassPath*, and its creation class is the class specified in *EnumClassPath* or a subclass of that class.

The result set should not contain any duplicate instances, as defined in 5.8.4. Because the result set contains only instances that exist in the same namespace, a determination of duplicate instances (for example by the Client) can be done on the basis of their model paths only.

The set of properties to be included in any instances in the result set shall be determined using the following algorithm:

- Initially, the set of properties to be included is the set of properties exposed by the creation class of the instance. This includes all the duplicates of any duplicate non-overridden properties.
- If the *IncludedProperties* operation input parameter is supported by the WBEM protocol and if its value is not NULL, it acts as a restricting filter on the properties to be included in the returned instances such that any properties exposed by the creation class of the instance that are not named in that operation parameter are removed from the set of properties to be included. Any duplicate or invalid property names in the *IncludedProperties* operation input parameter shall be ignored. A non-NULL empty *IncludedProperties* list removes all properties from the set of properties to be included.
- If the *ExcludeSubclassProperties* operation input parameter is supported by the WBEM protocol and if its value is TRUE, it acts as a restricting filter on the properties to be included in the returned instances such that any properties not exposed by the class referenced by *EnumClassPath* are removed from the set of properties to be included. In other words, the set of properties is restricted to the properties exposed by the enumeration class.
- Conformant WBEM protocols may specify rules that cause properties with a value of NULL to be removed from the set of properties to be included.

Preconditions:

- The CIM class referenced by *EnumClassPath* shall exist in the namespace. If this is not satisfied, the operation shall fail, indicating WIPG0214.

Postconditions:

- The enumerated instances with their instance paths shall have been returned as described in the Description paragraph for this operation.
- Requirements on ACID properties:
  - Atomicity: N/A
  - Update Consistency: N/A
  - Isolation: Required at the level of single instances, as defined in 5.8.
  - Durability: N/A

Error Messages:

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0240	WBEM service limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure, class implem.	
WIPG0214	Class not found	Mandatory	Infrastructure	
WIPG0228	Operation not supported by class implementation	Mandatory	Class implem.	
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

6.4.2 EnumerateInstanceNames

Purpose:

Enumerate the CIM instances of a class and return their instance paths.

Operation Input Parameters:

Generic Name	Generic Type	Requirement	Description
EnumClassPath	ClassPath	Mandatory	Class path of CIM class used for the enumeration (Context Parameter)

Operation Output Parameters:

Generic Name	Generic Type	Requirement	Description
InstancePathList	InstancePath [ ]	Mandatory	Sequence of instance paths of the enumerated instances

Description:

The *EnumerateInstanceNames* operation enumerates all CIM instances of the class referenced by *EnumClassPath*, and returns the instance paths of these instances.

An instance is included in the result set if and only if it exists in the namespace specified in *EnumClassPath*, and its creation class is the class specified in *EnumClassPath* or a subclass of that class.

The result set should not contain any duplicate instances, as defined in 5.8.4. Because the result set contains only instances that exist in the same namespace, a determination of duplicate instances (for example by the Client) can be done on the basis of their model paths only.

Preconditions:

- The CIM class referenced by *EnumClassPath* shall exist in the namespace. If this is not satisfied, the operation shall fail, indicating WIPG0214.

Postconditions:

- The instance paths of the enumerated instances shall have been returned as described in the Description paragraph for this operation.
- Requirements on ACID properties:
  - Atomicity: N/A
  - Update Consistency: N/A
  - Isolation: Required at the level of single instances, as defined in 5.8.
  - Durability: N/A

Error messages:

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0240	WBEM service limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure, class implem.	
WIPG0214	Class not found	Mandatory	Infrastructure	
WIPG0228	Operation not supported by class implementation	Mandatory	Class implem.	
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

6.4.3 Associators

Purpose:

Enumerate CIM instances that are associated with a given source instance and return those instances together with their instance paths.

Operation Input Parameters:

Generic Name	Generic Type	Requirement	Description
SourceInstancePath	InstancePath	Mandatory	Instance path of the source instance (Context Parameter)
AssociationClass-Name	ClassName	Mandatory	NULL, or name of the association class, acting as a restricting filter on the returned instances
AssociatedClass-Name	ClassName	Mandatory	NULL, or name of the associated class on any far end of the association, acting as a restricting filter on the returned instances
SourceRoleName	PropertyName	Mandatory	NULL, or name of the role on the source end of the association, acting as a restricting filter on the returned instances
AssociatedRoleName	PropertyName	Mandatory	NULL, or name of the role on any far end of the association, acting as a restricting filter on the returned instances
IncludeClassOrigin	boolean	Conditional	Indicates whether class origin information for any returned properties is to be included, as defined in 6.2.1 Condition: WBEM protocol supports client side control of returning class origin information
IncludedProperties	PropertyName [ ]	Optional	NULL, or unordered set of property names to be included, acting as a restricting filter on the properties included in the returned instances
ExcludeSubclass-Properties	boolean	Optional	Indicates whether properties added by subclasses of the associated class are to be excluded, acting as a restricting filter on the properties included in the returned instances

Operation Output Parameters:

Generic Name	Generic Type	Requirement	Description
InstanceList	InstanceSpecificationWithPath [ ]	Mandatory	Sequence of the associated instances with their instance paths

Description:

The *Associators* operation enumerates instances that are associated with a given source instance and returns these instances together with their instance paths.

The set of associated instances to be returned shall be determined using the following algorithm:

- Initially, the set of instances to be returned is the set of all instances associated to the source instance specified in *SourceInstancePath*. The associations may be instances of different association classes.

  The result set should not contain any duplicate instances, as defined in 5.8.4. However, different far ends may reference the same instance, and in such cases, the instance shall be contained in the result set once for each such reference.

- If the *AssociationClassName* operation input parameter is not NULL, it acts as a restricting filter on the instances to be returned such that each instance that is associated with the source instance using an association whose creation class or one of its superclasses does not have the name specified in *AssociationClassName*, is removed from the set of instances to be returned. There shall be no validity checking performed for the *AssociationClassName* operation input parameter.

- If the *AssociatedClassName* operation input parameter is not NULL, it acts as a restricting filter on the instances to be returned such that each instance whose creation class or one of its superclasses does not have the name specified in *AssociatedClassName*, is removed from the set of instances to be returned. There shall be no validity checking performed for the *AssociatedClassName* operation input parameter.

- If the *SourceRoleName* operation input parameter is not NULL, it acts as a restricting filter on the instances to be returned such that each instance that is associated with the source instance using an association class that has a role name on the source end that is not the role name specified in *SourceRoleName*, is removed from the set of instances to be returned. There shall be no validity checking performed for the *SourceRoleName* operation input parameter.

- If the *AssociatedRoleName* operation input parameter is not NULL, it acts as a restricting filter on the instances to be returned such that each instance that is associated with the source instance using an association class that has a role name on the end referencing that instance that is not the role name specified in *AssociatedRoleName*, is removed from the set of instances to be returned. There shall be no validity checking performed for the *AssociatedRoleName* operation input parameter.

The set of properties to be included in each returned associated instance shall be determined using the following algorithm:

- Initially, the set of properties to be included is the set of properties exposed by the creation class of the instance. This includes all the duplicates of any duplicate non-overridden properties.

- If the *IncludedProperties* operation input parameter is supported by the WBEM protocol and if its value is not NULL, it acts as a restricting filter on the properties to be included in the returned instances such that any properties exposed by the creation class of the instance that are not named in that operation parameter are removed from the set of properties to be included. Any duplicate or invalid property names in the *IncludedProperties* operation input parameter shall be ignored. A non-NULL empty *IncludedProperties* list removes all properties from the set of properties to be included.

- If the *ExcludeSubclassProperties* operation input parameter is supported by the WBEM protocol and if its value is TRUE, it acts as a restricting filter on the properties to be included in the returned instances such that any properties not exposed by the class specified in *AssociatedClassName* are removed from the set of properties to be included.

- Conformant WBEM protocols may specify rules that cause properties with a value of NULL to be removed from the set of properties to be included.

Preconditions:

- The instance referenced by *SourceInstancePath* shall exist in the namespace. If this is not satisfied, the operation shall fail, indicating WIPG0213.
- The *IncludedProperties* operation parameter, if supported by the WBEM protocol, shall only be specified with a non-NULL value if the *AssociatedClassName* operation input parameter is also non-NULL. If this is not satisfied, the operation shall fail, indicating WIPG0208.
- The *ExcludeSubclassProperties* operation parameter, if supported by the WBEM protocol, shall only be specified with a TRUE value if the *AssociatedClassName* operation input parameter is non-NULL. If this is not satisfied, the operation shall fail, indicating WIPG0208.

NOTE: Specifying a non-NULL value for *AssociatedClassName* ensures that the associated instances have the class specified in *AssociatedClassName* as a common superclass.

Postconditions:

- The associated instances with their instance paths shall have been returned as described in the Description paragraph for this operation.
- Requirements on ACID properties:
  - Atomicity: N/A
  - Update Consistency: N/A
  - Isolation: Required at the level of single instances, as defined in 5.8.
  - Durability: N/A

Error Messages:

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0240	WBEM service limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure, class implem.	
WIPG0214	Class not found	Mandatory	Infrastructure	
WIPG0228	Operation not supported by class implementation	Mandatory	Class implem.	
WIPG0213	Instance not found	Mandatory	Class implem.	

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

6.4.4 AssociatorNames

Purpose:

Enumerate CIM instances that are associated with a given source instance and return their instance paths.

Operation Input Parameters:

Generic Name	Generic Type	Requirement	Description
SourceInstancePath	InstancePath	Mandatory	Instance path of the source instance (Context Parameter)
AssociationClass-Name	ClassName	Mandatory	NULL, or name of the association class, acting as a restricting filter on the returned instances
AssociatedClass-Name	ClassName	Mandatory	NULL, or name of the associated class on any far end of the association, acting as a restricting filter on the returned instances
SourceRoleName	PropertyName	Mandatory	NULL, or name of the role on the source end of the association, acting as a restricting filter on the returned instances
AssociatedRoleName	PropertyName	Mandatory	NULL, or name of the role on any far end of the association, acting as a restricting filter on the returned instances

Operation Output Parameters:

Generic Name	Generic Type	Requirement	Description
InstancePathList	InstancePath [ ]	Mandatory	Sequence of the instance paths of the associated instances

Description:

The *AssociatorNames* operation enumerates the instance paths of instances that are associated with a given source instance and returns these instance paths.

The set of associated instances of which instance paths are to be returned shall be determined using the following algorithm:

- Initially, the set of instances to be returned is the set of all instances associated to the source instance specified in *SourceInstancePath*. The associations may be instances of different association classes.

  The result set should not contain any duplicate instances, as defined in 5.8.4. However, different association instances may reference the same instance on one of their far ends,

and in such cases, the instance shall be contained in the result set once for each such reference.

- If the *AssociationClassName* operation input parameter is not NULL, it acts as a restricting filter on the instances to be returned such that each instance that is associated with the source instance using an association whose creation class or one of its superclasses does not have the name specified in *AssociationClassName*, is removed from the set of instances to be returned. There shall be no validity checking performed for the *AssociationClassName* operation input parameter.
- If the *AssociatedClassName* operation input parameter is not NULL, it acts as a restricting filter on the instances to be returned such that each instance whose creation class or one of its superclasses does not have the name specified in *AssociatedClassName*, is removed from the set of instances to be returned. There shall be no validity checking performed for the *AssociatedClassName* operation input parameter.
- If the *SourceRoleName* operation input parameter is not NULL, it acts as a restricting filter on the instances to be returned such that each instance that is associated with the source instance using an association class that has a role name on the source end that is not the role name specified in *SourceRoleName*, is removed from the set of instances to be returned. There shall be no validity checking performed for the *SourceRoleName* operation input parameter.
- If the *AssociatedRoleName* operation input parameter is not NULL, it acts as a restricting filter on the instances to be returned such that each instance that is associated with the source instance using an association class that has a role name on the end referencing that instance that is not the role name specified in *AssociatedRoleName*, is removed from the set of instances to be returned. There shall be no validity checking performed for the *AssociatedRoleName* operation input parameter.

The consistency model defined in 5.8 applies.

Preconditions:

- The instance referenced by *SourceInstancePath* shall exist in the namespace. If this is not satisfied, the operation shall fail, indicating WIPG0213.

Postconditions:

- The instance paths of the associated instances shall have been returned as described in the Description paragraph for this operation.
- Requirements on ACID properties:
  - Atomicity: N/A
  - Update Consistency: N/A
  - Isolation: Required at the level of single instances, as defined in 5.8.
  - Durability: N/A

1331 **Error Messages:**
1332

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0240	WBEM service limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure, class implem.	
WIPG0214	Class not found	Mandatory	Infrastructure	
WIPG0228	Operation not supported by class implementation	Mandatory	Class implem.	
WIPG0213	Instance not found	Mandatory	Class implem.	
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

6.4.5 References

Purpose:

Enumerate CIM association instances that reference a given source instance and return these instances together with their instance path.

Operation Input Parameters:

Generic Name	Generic Type	Requirement	Description
SourceInstancePath	InstancePath	Mandatory	Instance path of the source instance (Context Parameter)
AssociationClass-Name	ClassName	Mandatory	NULL, or name of the association class, acting as a restricting filter on the returned instances
AssociatedClass-Name	ClassName	Mandatory	NULL, or name of the associated class on any far end of the association, acting as a restricting filter on the returned instances
SourceRoleName	PropertyName	Mandatory	NULL, or name of the role on the source end of the association, acting as a restricting filter on the returned instances
AssociatedRoleName	PropertyName	Mandatory	NULL, or name of the role on any far end of the association, acting as a restricting filter on the returned instances
IncludeClassOrigin	boolean	Conditional	Indicates whether class origin information for any returned properties is to be included, as defined in 6.2.1 Condition: WBEM protocol supports client side control of returning class origin information.
IncludedProperties	PropertyName [ ]	Optional	NULL, or unordered set of property names to be included, acting as a restricting filter on the properties included in the returned instances
ExcludeSubclass-Properties	boolean	Optional	Indicates whether properties added by subclasses of the association class are to be excluded, acting as a restricting filter on the properties included in the returned instances

Operation Output Parameters:

Generic Name	Generic Type	Requirement	Description
InstanceList	InstanceSpecificationWithPath [ ]	Mandatory	Sequence of the association instances with their instance paths

Description:

The *References* operation enumerates association instances that reference the specified source instance and returns these instances together with their instance paths.

The set of association instances to be returned shall be determined using the following algorithm:

- Initially, the set of instances to be returned is the set of all instances referencing the source instance specified in *SourceInstancePath*. These associations may be instances of different association classes.
- If the *AssociationClassName* operation input parameter is not NULL, it acts as a restricting filter on the instances to be returned such that each association instance whose creation class or one of its superclasses does not have the name specified in *AssociationClassName*, is removed from the set of instances to be returned. There shall be no validity checking performed for the *AssociationClassName* operation input parameter.
- If the *AssociatedClassName* operation input parameter is not NULL, it acts as a restricting filter on the instances to be returned such that each association instance whose creation class has a set of far ends none of which is referencing a class where that class or one of its superclasses has the name specified in *AssociatedClassName*, is removed from the set of instances to be returned. There shall be no validity checking performed for the *AssociatedClassName* operation input parameter.
- If the *SourceRoleName* operation input parameter is not NULL, it acts as a restricting filter on the instances to be returned such that each association instance whose creation class does not have the role name specified in *SourceRoleName* on the end referencing the source instance, is removed from the set of instances to be returned. There shall be no validity checking performed for the *SourceRoleName* operation input parameter.
- If the *AssociatedRoleName* operation input parameter is not NULL, it acts as a restricting filter on the instances to be returned such that each association instance whose creation class has a set of far ends none of which has the role name specified in *AssociatedRoleName*, is removed from the set of instances to be returned. There shall be no validity checking performed for the *AssociatedRoleName* operation input parameter.

The consistency model defined in 5.8 applies.

The set of properties to be included in each returned association instance shall be determined using the following algorithm:

- Initially, the set of properties to be included is the set of properties exposed by the creation class of the instance. This includes all the duplicates of any duplicate non-overridden properties.
- If the *IncludedProperties* operation input parameter is supported by the WBEM protocol and if its value is not NULL, it acts as a restricting filter on the properties to be included in the returned instances such that any properties exposed by the creation class of the instance that are not named in that operation parameter are removed from the set of properties to be included. Any duplicate or invalid property names in the *IncludedProperties* operation input parameter shall be ignored. A non-NULL empty *IncludedProperties* list removes all properties from the set of properties to be included
- If the *ExcludeSubclassProperties* operation input parameter is supported by the WBEM protocol and if its value is TRUE, it acts as a restricting filter on the properties to be included in the returned instances such that any properties not exposed by the class specified in *AssociationClassName* are removed from the set of properties to be included.
- Conformant WBEM protocols may specify rules that cause properties with a value of NULL to be removed from the set of properties to be included.

Preconditions:

- The instance referenced by *SourceInstancePath* shall exist in the namespace. If this is not satisfied, the operation shall fail, indicating WIPG0213.
- The *IncludedProperties* operation parameter, if supported by the WBEM protocol, shall only be specified with a non-NULL value if the *AssociationClassName* operation input parameter is also non-NULL. If this is not satisfied, the operation shall fail, indicating WIPG0208.
- The *ExcludeSubclassProperties* operation parameter, if supported by the WBEM protocol, shall only be specified with a TRUE value if the *AssociationClassName* operation input parameter is non-NULL. If this is not satisfied, the operation shall fail, indicating WIPG0208.

NOTE: Specifying a non-NULL value for *AssociationClassName* ensures that the association instances have the class specified in *AssociationClassName* as a common superclass.

Postconditions:

- The association instances with their instance paths shall have been returned as described in the Description paragraph for this operation.
- Requirements on ACID properties:
  - Atomicity: N/A
  - Update Consistency: N/A
  - Isolation: Required at the level of single instances, as defined in 5.8.
  - Durability: N/A

Error Messages:

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0240	WBEM service limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure, class implem.	
WIPG0214	Class not found	Mandatory	Infrastructure	
WIPG0228	Operation not supported by class implementation	Mandatory	Class implem.	
WIPG0213	Instance not found	Mandatory	Class implem.	

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

6.4.6 ReferenceNames

Purpose:

Enumerate CIM association instances that reference a given source instance and return their instance paths.

Operation Input Parameters:

Generic Name	Generic Type	Requirement	Description
SourceInstancePath	InstancePath	Mandatory	Instance path of the source instance (Context Parameter)
AssociationClass-Name	ClassName	Mandatory	NULL, or name of the association class, acting as a restricting filter on the returned instances
AssociatedClass-Name	ClassName	Mandatory	NULL, or name of the associated class on any far end of the association, acting as a restricting filter on the returned instances
SourceRoleName	PropertyName	Mandatory	NULL, or name of the role on the source end of the association, acting as a restricting filter on the returned instances
AssociatedRoleName	PropertyName	Mandatory	NULL, or name of the role on any far end of the association, acting as a restricting filter on the returned instances

Operation Output Parameters:

Generic Name	Generic Type	Requirement	Description
InstancePathList	InstancePath [ ]	Mandatory	Sequence of the instance paths of the association instances

Description:

The *ReferenceNames* operation enumerates the instance paths of association instances that reference the specified source instance and returns these instance paths.

The set of association instances of which instance paths are to be returned shall be determined using the following algorithm:

- Initially, the set of instances to be returned is the set of all instances referencing the source instance specified in *SourceInstancePath*. These associations may be instances of different association classes.
- If the *AssociationClassName* operation input parameter is not NULL, it acts as a restricting filter on the instances to be returned such that each association instance whose creation

class or one of its superclasses does not have the name specified in *AssociationClassName*, is removed from the set of instances to be returned. There shall be no validity checking performed for the *AssociationClassName* operation input parameter.

- If the *AssociatedClassName* operation input parameter is not NULL, it acts as a restricting filter on the instances to be returned such that each association instance whose creation class has a set of far ends none of which is referencing a class where that class or one of its superclasses has the name specified in *AssociatedClassName*, is removed from the set of instances to be returned. There shall be no validity checking performed for the *AssociatedClassName* operation input parameter.
- If the *SourceRoleName* operation input parameter is not NULL, it acts as a restricting filter on the instances to be returned such that each association instance whose creation class does not have the role name specified in *SourceRoleName* on the end referencing the source instance, is removed from the set of instances to be returned. There shall be no validity checking performed for the *SourceRoleName* operation input parameter.
- If the *AssociatedRoleName* operation input parameter is not NULL, it acts as a restricting filter on the instances to be returned such that each association instance whose creation class has a set of far ends none of which has the role name specified in *AssociatedRoleName*, is removed from the set of instances to be returned. There shall be no validity checking performed for the *AssociatedRoleName* operation input parameter.

The consistency model defined in 5.8 applies.

Preconditions:

- The instance referenced by *SourceInstancePath* shall exist in the namespace. If this is not satisfied, the operation shall fail, indicating WIPG0213.

Postconditions:

- The instance paths of the association instances shall have been returned as described in the Description paragraph for this operation.
- Requirements on ACID properties:
  - Atomicity: N/A
  - Update Consistency: N/A
  - Isolation: Required at the level of single instances, as defined in 5.8.
  - Durability: N/A

Error Messages:

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0240	WBEM service limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure, class implem.	
WIPG0214	Class not found	Mandatory	Infrastructure	
WIPG0228	Operation not supported by class implementation	Mandatory	Class implem.	
WIPG0213	Instance not found	Mandatory	Class implem.	
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

6.5 Pulled instance enumeration operations

This subclause defines pulled instance enumeration operations (operations that enumerate CIM instances and return them by means of subsequent pull operations).

The common pattern for these operations is that an enumeration session gets established through an "Open" operation, also establishing the kind of operation and the kind of items to be returned (instances or instance paths of instances), and subsequent repeated executions of a "Pull" operation on the enumeration session are used to retrieve the items. Optionally, the "Open" operation can also pull a first set of items.

The pulled instance enumeration operations consist of the following individual operations:

- Open operations:

  OpenEnumerateInstances – Open an enumeration of instances of a class

  OpenEnumerateInstancePaths – Open an enumeration of the instance paths of instances of a class

  OpenAssociators – Open an enumeration of instances associated to a source instance

  OpenAssociatorPaths – Open an enumeration of the instance paths of instances associated to a source instance

  OpenReferences – Open an enumeration of association instances referencing a source instance

  OpenReferencePaths – Open an enumeration of the instance paths of association instances referencing a source instance

  OpenQueryInstances – Open an enumeration of instances representing a query result

- Pull operations:

  PullInstancesWithPath – Pull operation for retrieving instances with paths

  PullInstancePaths – Pull operation for retrieving instance paths

  PullInstances – Pull operation for retrieving instances without paths representing query results

- Other operations:

  CloseEnumeration – Close an open enumeration

  EnumerationCount – Estimate number of items in an open enumeration

6.5.1 General behavioral rules

A central concept of the pulled instance enumeration operations is the "enumeration session". An enumeration session can be thought of as a context in which the operations perform their work, and which determines the set of objects to be returned. In order to process the operations related to an enumeration session, some of the operation parameters of the Open operation need to be maintained as long as the enumeration session is open, as well as some state data about where the enumeration session is with respect to objects already returned.

From a WBEM client's perspective, an enumeration session is represented as an enumeration context value. A successful Open operation establishes the enumeration session and returns an enumeration context value representing the open enumeration session. The enumeration context value is used as an operation input/output parameter in subsequent Pull operations on that enumeration session. The

enumeration context value shall uniquely identify the open enumeration session within the target CIM namespace of the Open operation that established the enumeration session. This does not require the enumeration context value to be time-unique, i.e., it may be reused for a new enumeration session after the old enumeration session was closed. It is valid for a WBEM server to use NULL as an enumeration context value representing a closed enumeration session, but a WBEM client shall not rely on that to detect that an enumeration session has been closed.

Defining the enumeration context value in Pull operations not only as an operation input parameter but also as an operation output parameter allows the WBEM server to change the enumeration context value during the execution of a Pull operation. This allows for different implementation approaches for the WBEM server, which are transparent for the WBEM client.

Example approaches are:

- maintaining any state data describing the enumeration session internally in the WBEM server. In this approach, the enumeration context value does not need to change in subsequent Pull operations. It is used by the WBEM server only to identify the internal state data for the open enumeration session, but it is not used to store any of the state data in it. A variation of this approach is to hand back modified enumeration context values for additional WBEM server side sequence checking.
- maintaining any state data describing the enumeration session on the WBEM client side only. In this approach, all state data is stored in the enumeration context value, and the WBEM server does not maintain any state data about the enumeration session, essentially being completely stateless with respect to the enumeration session.
- a combination of the two previous approaches

A WBEM server may support keeping enumeration sessions open across connection terminations and shutdowns of the server. Objects may be created, deleted or modified concurrently with an enumeration session that involves these objects. Such changes may or may not be reflected in the enumeration set. Therefore, there is no guarantee to the WBEM client that the enumeration set represents a consistent snapshot of its objects at a point in time. However, the WBEM server should make a best effort attempt for the returned enumeration set to represent a consistent snapshot of its objects at a point in time. The order of objects in the enumeration set is undefined.

This specification does not define any restrictions on the number of enumeration sessions that can be established or executed on concurrently in the same WBEM server or by the same WBEM client. This remains true even if the enumeration sets of such concurrently established enumeration sessions contain the same objects.

With the exception of CloseEnumeration, all operations on a particular enumeration session shall be executed sequentially. An enumeration session can be open or closed. The enumeration session is considered open if operations using its enumeration context value as an operation input parameter can be executed successfully. It is opened by the successful completion of an Open operation and closed by one of the following:

- Successful completion of a CloseEnumeration operation
- Successful completion of an Open or Pull operation that has its *EndOfSequence* operation output parameter set to TRUE. In other words, reaching the end of the enumeration set closes the enumeration session implicitly
- Unsuccessful completion of a Pull operation when *ContinueOnError* had not been requested
- WBEM server side decision to close the enumeration session based upon an operation timeout
- WBEM server side decision to close an enumeration session during an operation on that enumeration session based upon exceeding server limits.

A conformant WBEM server may support closure of enumeration sessions based upon exceeding server limits. Potential examples for such a decision may be Pull operations with no objects requested that are repeated with a high frequency on the same enumeration session, or EnumerationCount operations repeated with a high frequency on the same enumeration session. If a WBEM server supports closure of enumeration sessions based upon exceeding server limits, it shall make the decision to close an enumeration session during an operation on that enumeration session. (There is no way to indicate the reason for the closure if the decision is made elsewhere.)

6.5.2 Common operation parameters for the open operations

This subclause defines commonly used operation parameters for the Open operations. The description of the individual Open operations references these operation parameters as appropriate. However, not every Open operation uses every one of these common operation parameters.

6.5.2.1 EnumerationContext

The *EnumerationContext* operation output parameter is the enumeration context value representing the enumeration session. See 6.5.1 for a definition of the concepts of *enumeration session* and *enumeration context value*.

6.5.2.2 EndOfSequence

NOTE: This operation output parameter is also used for Pull operations.

The *EndOfSequence* operation output parameter indicates whether the enumeration session is exhausted.

If *EndOfSequence* is TRUE upon successful completion of an operation, no more objects are available and the WBEM server shall have closed the enumeration session, releasing any possibly allocated resources related to the enumeration session.

If the returned enumeration set is empty, it is valid for a WBEM server to set *EndOfSequence* to TRUE, even if *MaxObjectCount* was 0. In this case, the enumeration session will be closed upon successful completion of the operation.

If *EndOfSequence* is FALSE upon successful completion of an operation, there may be additional elements available and the WBEM server shall not have closed the enumeration session.

6.5.2.3 FilterQueryLanguage and FilterQueryString

The *FilterQueryLanguage* and *FilterQueryString* operation input parameters define a filter query that acts as an additional restricting filter on the set of instances about which information is returned (that is, the instances themselves or their instance paths).

Support for the *FilterQueryLanguage* and *FilterQueryString* operation parameters is conditional on support in the WBEM protocol for filter queries in pulled instance enumeration operations.

If the WBEM protocol supports filter queries in pulled instance enumeration operations, the following rules apply:

- If *FilterQueryLanguage* is not NULL, additional filtering is requested and the following rules apply:
  - *FilterQueryLanguage* shall specify a valid query language and *FilterQueryString* shall be a valid query in that query language. Neither the query language nor the format of the filter query is defined by this specification. Conformant WBEM protocols shall define a mechanism whereby WBEM servers can declare the set of query languages that are valid for *FilterQueryLanguage*.

1587 – A filter query may specify any result set (e.g., SELECT list), but because the purpose
1588 of the filter query is to restrict the set of instances about which information is returned,
1589 its result set shall be ignored. The filter query shall not define any ordering criteria.
1590 The filter query shall not define any grouping of objects. Operations using filter queries
1591 may specify additional constraints on the filter query.

1592 – If the WBEM server infrastructure does not support filtered enumerations, the WBEM
1593 server shall return failure with message WIPG0237 (Filter queries not supported by
1594 WBEM server infrastructure).

1595 – If the CIM class implementation does not support filtered enumerations, the WBEM
1596 server shall return failure with message WIPG0244 (Filter queries not supported by
1597 class implementation).

1598 • If *FilterQueryLanguage* is NULL, no additional filtering shall take place, and *FilterQueryString*
1599 shall be NULL.

1600 – If *FilterQueryString* is not NULL, the WBEM server shall return failure with message
1601 WIPG0208 (Invalid operation input parameter value).

1602 If the WBEM protocol does not support filter queries in pulled instance enumeration operations, no
1603 additional filtering shall take place.

1604 #### 6.5.2.4 OperationTimeout

1605 The *OperationTimeout* operation input parameter determines the "operation timeout". The operation
1606 timeout is the minimum time the WBEM server shall maintain the open enumeration session after the last
1607 Open or Pull operation (unless the enumeration session was closed during that last operation). If the
1608 operation timeout is exceeded, the WBEM server may close the enumeration session at any time,
1609 releasing any possibly allocated resources related to the enumeration session.

1610 Support for the *OperationTimeout* operation parameter in a conformant WBEM protocol is mandatory.

1611 An *OperationTimeout* of 0 means that there is no operation timeout, i.e., the enumeration session is never
1612 closed based on time.

1613 If *OperationTimeout* is NULL, the WBEM server shall choose an operation timeout.

1614 All other values for *OperationTimeout* specify the operation timeout in seconds.

1615 A WBEM server may restrict the set of allowable values for *OperationTimeout*. This specifically includes
1616 the possibility for the WBEM server to not allow 0 (no timeout). If the specified value is not an allowable
1617 value, the WBEM server shall return failure with error message WIPG0242 (Invalid timeout). Conformant
1618 WBEM protocols shall define a mechanism whereby WBEM servers can declare the allowable values for
1619 *OperationTimeout*.

1620 #### 6.5.2.5 ContinueOnError

1621 The *ContinueOnError* operation input parameter, if TRUE, requests continuation on error. Continuation on
1622 error is the ability to resume an enumeration session successfully after a Pull operation that returned an
1623 error. A conformant WBEM server may support continuation on error. Conformant WBEM protocols shall
1624 define a mechanism whereby WBEM servers can declare support for continuation on error.

1625 Support for the *ContinueOnError* operation parameter is conditional on support in the WBEM protocol for
1626 client side control of continuation on error for pulled instance enumeration operations.

1627 If the WBEM protocol supports client side control of continuation on error for pulled instance enumeration
1628 operations, the following rules apply:

- If a WBEM server does not support continuation on error and if *ContinueOnError* is TRUE, it shall return failure with error message WIPG0235 (Continuation on error not supported).
- If a WBEM server supports continuation on error, it shall support it as follows: If *ContinueOnError* is TRUE, the enumeration session shall remain open when a Pull operation returns failure, and any subsequent successful Pull operations shall return the set of elements that would have been returned if the failing Pull operations had been successful, subject to the consistency rules defined in 5.8. If *ContinueOnError* is FALSE, the enumeration session shall be closed when a Pull operation returns failure.

If the WBEM protocol does not support client side control of continuation on error for pulled instance enumeration operations, it shall define requirements for the behavior of the WBEM server with respect to continuation on error.

6.5.2.6 MaxObjectCount

NOTE: This operation output parameter is also used for Pull operations.

The *MaxObjectCount* operation input parameter defines the maximum number of objects that may be returned by this operation. Any uint32 number is valid, including 0. The WBEM server may deliver any number of objects up to *MaxObjectCount* but shall not deliver more than *MaxObjectCount* objects.

Support for the *MaxObjectCount* operation parameter in a conformant WBEM protocol is mandatory.

A conformant WBEM server implementation may choose to never return any elements during an operation, regardless of the value of *MaxObjectCount*.

A WBEM client may use a *MaxObjectCount* value of 0 to specify that it does not want to retrieve any instances in the operation.

6.5.3 OpenEnumerateInstances

Purpose:

Establish and open an enumeration session for enumerating the instances of a class (including instances of its subclasses), and optionally retrieve a first set of instances.

Operation Input Parameters:

Generic Name	Generic Type	Requirement	Description
EnumClassPath	ClassPath	Mandatory	Class path of CIM class used for the enumeration (Context Parameter)
FilterQueryString	QueryString	Conditional	NULL, or query string of a filter query that is acting as an additional restricting filter on the set of instances to be returned, as defined in 6.5.2.3 Condition: WBEM protocol supports filter queries for pulled instance enumeration operations.
FilterQueryLanguage	QueryLanguage	Conditional	NULL, or query language of the filter query specified in *FilterQueryString*, as defined in 6.5.2.3 Condition: WBEM protocol supports filter queries for pulled instance enumeration operations.

Generic Name	Generic Type	Requirement	Description
IncludeClassOrigin	boolean	Conditional	Indicates whether class origin information for any returned properties is to be included, as defined in 6.2.1 Condition: WBEM protocol supports client side control of returning class origin information
IncludedProperties	PropertyName [ ]	Optional	NULL, or unordered set of property names to be included, acting as a restricting filter on the properties included in the returned instances
ExcludeSubclass-Properties	boolean	Optional	Indicates whether properties added by subclasses of the class used for the enumeration are to be excluded, acting as a restricting filter on the properties included in the returned instances
OperationTimeout	uint32	Mandatory	Operation timeout, as defined in 6.5.2.4
ContinueOnError	boolean	Conditional	Indicates whether the enumeration session should be continued in case of error, as defined in 6.5.2.5 Condition: WBEM protocol supports client side control of continuation on error for pulled instance enumeration operations.
MaxObjectCount	uint32	Mandatory	Maximum number of instances that may be returned by this operation, as defined in 6.5.2.6

Operation Output Parameters:

Generic Name	Generic Type	Requirement	Description
InstanceList	InstanceSpecificationWithPath [ ]	Mandatory	Sequence of instances with their instance paths of the first set of instances
EnumerationContext	EnumerationContext	Mandatory	Enumeration context value, as defined in 6.5.2.1
EndOfSequence	boolean	Mandatory	Indicates end of sequence for the enumeration session, as defined in 6.5.2.2

Description:

The *OpenEnumerateInstances* operation establishes and opens an enumeration session for enumerating all CIM instances of the class referenced by *EnumClassPath*, including instances of any of its subclasses. Retrieval of a first set of those instances together with their instance paths may be requested by setting *MaxObjectCount* to a value > 0.

The set of instances to be returned throughout the entire enumeration session shall be determined using the following algorithm:

- Initially, the set of instances to be returned is the set of instances in the namespace specified in *EnumClassPath*, whose creation class is the class specified in *EnumClassPath* or a subclass of that class.
- If the WBEM protocol supports filter queries for pulled instance enumeration operations (that is, the *FilterQueryString* and *FilterQueryLanguage* operation parameters) and *FilterQueryLanguage* is not NULL, *FilterQueryString* acts as a restricting filter on the

1671 instances to be returned such that any instances not selected by the filter query for its
1672 result set are removed from the set of instances. The filter query shall query only the class
1673 specified in *EnumClassPath*. See also 6.5.2.3.

1674 The set of instances to be returned throughout the entire enumeration session should not contain
1675 any duplicate instances, as defined in 5.8.4. Because the set of returned instances contains only
1676 instances that exist in the same namespace, a determination of duplicate instances (for example by
1677 a WBEM client) can be done on the basis of their model paths only.

1678 The set of instances to be returned in the *InstanceList* operation parameter is the first set of
1679 instances from the set of instances to be returned throughout the entire enumeration session, such
1680 that no more than *MaxObjectCount* instances are returned. Returning no instances does not imply
1681 that the enumeration session has been exhausted. Only the *EndOfSequence* operation output
1682 parameter indicates whether the enumeration session has been exhausted.

1683 The set of properties to be included in any returned instances shall be determined using the following
1684 algorithm:

1685 • Initially, the set of properties to be included is the set of properties exposed by the creation
1686 class of the instance. This includes all the duplicates of any duplicate non-overridden
1687 properties.

1688 • If the *IncludedProperties* operation input parameter is supported by the WBEM protocol
1689 and if its value is not NULL, it acts as a restricting filter on the properties to be included in
1690 the returned instances such that any properties exposed by the creation class of the
1691 instance that are not named in that operation parameter are removed from the set of
1692 properties to be included. Any duplicate or invalid property names in the
1693 *IncludedProperties* operation input parameter shall be ignored. A non-NULL empty
1694 *IncludedProperties* list removes all properties from the set of properties to be included.

1695 • If the *ExcludeSubclassProperties* operation input parameter is supported by the WBEM
1696 protocol and if its value is TRUE, it acts as a restricting filter on the properties to be
1697 included in the returned instances such that any properties not exposed by the class
1698 referenced by *EnumClassPath* are removed from the set of properties to be included. In
1699 other words, the set of properties is restricted to the properties exposed by the
1700 enumeration class.

1701 • Conformant WBEM protocols may specify rules that cause properties with a value of NULL
1702 to be removed from the set of properties to be included.

1703 **Preconditions:**

1704 • The CIM class referenced by *EnumClassPath* shall exist in the namespace. If this is not
1705 satisfied, the operation shall fail, indicating WIPG0214.

1706 • If a filter query is specified,

1707 – the query language specified in the *FilterQueryLanguage* operation parameter shall be
1708 valid. If this is not satisfied, the operation shall fail, indicating WIPG0221.

1709 – the query specified in the *FilterQueryString* operation parameter shall be a valid query in
1710 the query language specified in the *FilterQueryLanguage* operation parameter. If this is not
1711 satisfied, the operation shall fail, indicating WIPG0222 or WIPG0223.

1712 **Postconditions:**

1713 • The enumeration session shall have been established and opened.

1714 • A first set of instances with their instance paths shall have been returned as described in the
1715 Description paragraph for this operation.

- Requirements on ACID properties:
  - Atomicity: Required (related to the creation of an enumeration context that is maintained by the WBEM server)
  - Update Consistency: N/A
  - Isolation: Required at the level of single instances, as defined in 5.8.
  - Durability: Required (related to creation of an enumeration context that is maintained by the WBEM server)

Error Messages:

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0240	WBEM service limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0242	Invalid timeout	Mandatory	Infrastructure, class implem.	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure, class implem.	
WIPG0235	Continuation on error not supported	Mandatory	Infrastructure, class implem.	
WIPG0237	Filter queries not supported by WBEM service infrastructure	Optional	Infrastructure	
WIPG0244	Filter queries not supported by class implementation	Optional	Class implem.	
WIPG0221	Unknown query language	Mandatory	Infrastructure, class implem.	
WIPG0222	Query language feature not supported	Mandatory	Infrastructure, class implem.	
WIPG0223	Invalid query	Mandatory	Infrastructure, class implem.	
WIPG0214	Class not found	Mandatory	Infrastructure	
WIPG0228	Operation not supported by class implementation	Mandatory	Class implem.	

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

6.5.4 OpenEnumerateInstancePaths

Purpose:

Establish and open an enumeration session for enumerating the instances of a class (including instances of its subclasses), and optionally retrieve a first set of instance paths of those instances.

Operation Input Parameters:

Generic Name	Generic Type	Requirement	Description
EnumClassPath	ClassPath	Mandatory	Class path of CIM class used for the enumeration (Context Parameter)
FilterQueryString	QueryString	Conditional	NULL, or query string of a filter query that is acting as an additional restricting filter on the set of enumerated instance paths, as defined in 6.5.2.3 Condition: WBEM protocol supports filter queries for pulled instance enumeration operations.
FilterQueryLanguage	QueryLanguage	Conditional	NULL, or query language of the filter query specified in *FilterQueryString*, as defined in 6.5.2.3. Condition: WBEM protocol supports filter queries for pulled instance enumeration operations.
OperationTimeout	uint32	Mandatory	Operation timeout, as defined in 6.5.2.4
ContinueOnError	boolean	Conditional	Indicates whether the enumeration session should be continued in case of error, as defined in 6.5.2.5 Condition: WBEM protocol supports client side control of continuation on error for pulled instance enumeration operations.
MaxObjectCount	uint32	Mandatory	Maximum number of instance paths that may be returned by this operation, as defined in 6.5.2.6

Operation Output Parameters:

Generic Name	Generic Type	Requirement	Description
InstancePathList	InstancePath [ ]	Mandatory	Sequence of instance paths of the first set of instances

Generic Name	Generic Type	Requirement	Description
EnumerationContext	EnumerationContext	Mandatory	Enumeration context value, as defined in 6.5.2.1
EndOfSequence	boolean	Mandatory	Indicates end of sequence for the enumeration session, as defined in 6.5.2.2

Description:

The *OpenEnumerateInstancePaths* operation establishes and opens an enumeration session for enumerating the CIM instance paths of all instances of the class referenced by *EnumClassPath*, including of instances of any of its subclasses. Retrieval of a first set of those instance paths may be requested by setting *MaxObjectCount* to a value > 0.

The set of instances from which instance paths are to be returned throughout the entire enumeration session shall be determined using the following algorithm:

- Initially, the set of instances to be returned is the set of instances in the namespace specified in *EnumClassPath*, whose creation class is the class specified in *EnumClassPath* or a subclass of that class.
- If the WBEM protocol supports filter queries for pulled instance enumeration operations (that is, the *FilterQueryString* and *FilterQueryLanguage* operation parameters) and *FilterQueryLanguage* is not NULL, *FilterQueryString* acts as a restricting filter on the instances to be returned such that any instances not selected by the filter query for its result set are removed from the set of instances. The filter query shall query only the class specified in *EnumClassPath*. See also 6.5.2.3.

The set of instance paths to be returned throughout the entire enumeration session should not contain any duplicate instance paths, as defined in 5.8.4. Because the instances referenced by the set of returned instance paths contains only instances that exist in the same namespace, a determination of duplicate instance paths can be done on the basis of their model paths only.

The set of instance paths to be returned in the *InstancePathList* operation parameter is the first set of instance paths from the set of instance paths to be returned throughout the entire enumeration session, such that no more than *MaxObjectCount* instance paths are returned. Returning no instance paths does not imply that the enumeration session has been exhausted. Only the *EndOfSequence* operation output parameter indicates whether the enumeration session has been exhausted.

Preconditions:

- The CIM class referenced by *EnumClassPath* shall exist in the namespace. If this is not satisfied, the operation shall fail, indicating WIPG0214.
- If a filter query is specified,
  - the query language specified in the *FilterQueryLanguage* operation parameter shall be valid. If this is not satisfied, the operation shall fail, indicating WIPG0221.
  - the query specified in the *FilterQueryString* operation parameter shall be a valid query in the query language specified in the *FilterQueryLanguage* operation parameter. If this is not satisfied, the operation shall fail, indicating WIPG0222 or WIPG0223.

Postconditions:

- The enumeration session shall have been established and opened.
- A first set of instance paths shall have been returned as described in the Description paragraph for this operation.

- Requirements on ACID properties:
  - Atomicity: Required (related to the creation of an enumeration context that is maintained by the WBEM server)
  - Update Consistency: N/A
  - Isolation: Required at the level of single instances, as defined in 5.8.
  - Durability: Required (related to creation of an enumeration context that is maintained by the WBEM server)

Error Messages:

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0240	WBEM service limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0242	Invalid timeout	Mandatory	Infrastructure, class implem.	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure, class implem.	
WIPG0235	Continuation on error not supported	Mandatory	Infrastructure, class implem.	
WIPG0237	Filter queries not supported by WBEM service infrastructure	Optional	Infrastructure	
WIPG0244	Filter queries not supported by class implementation	Optional	Class implem.	
WIPG0221	Unknown query language	Mandatory	Infrastructure, class implem.	
WIPG0222	Query language feature not supported	Mandatory	Infrastructure, class implem.	
WIPG0223	Invalid query	Mandatory	Infrastructure, class implem.	
WIPG0214	Class not found	Mandatory	Infrastructure	
WIPG0228	Operation not supported by class implementation	Mandatory	Class implem.	

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

6.5.5 OpenAssociators

Purpose:

Establish and open an enumeration session for enumerating instances that are associated with a given source instance, and optionally retrieve a first set of those instances.

Operation Input Parameters:

Generic Name	Generic Type	Requirement	Description
SourceInstancePath	InstancePath	Mandatory	Instance path of the source instance (Context Parameter)
AssociationClass-Name	ClassName	Mandatory	NULL, or name of the association class, acting as a restricting filter on the returned instances
AssociatedClass-Name	ClassName	Mandatory	NULL, or name of the associated class on any far end of the association, acting as a restricting filter on the returned instances
SourceRoleName	PropertyName	Mandatory	NULL, or name of the role on the source end of the association, acting as a restricting filter on the returned instances
AssociatedRoleName	PropertyName	Mandatory	NULL, or name of the role on any far end of the association, acting as a restricting filter on the returned instances
FilterQueryString	QueryString	Conditional	NULL, or query string of a filter query that is acting as an additional restricting filter on the set of returned instances, as defined in 6.5.2.3 Condition: WBEM protocol supports filter queries for pulled instance enumeration operations.
FilterQueryLanguage	QueryLanguage	Conditional	NULL, or query language of the filter query specified in *FilterQueryString*, as defined in 6.5.2.3 Condition: WBEM protocol supports filter queries for pulled instance enumeration operations.
IncludeClassOrigin	boolean	Conditional	Indicates whether class origin information for any returned properties is to be included, as defined in 6.2.1 Condition: WBEM protocol supports client side control of returning class origin information.
IncludedProperties	PropertyName [ ]	Optional	NULL, or unordered set of property names to be included, acting as a restricting filter on the properties included in the returned instances

Generic Name	Generic Type	Requirement	Description
ExcludeSubclass-Properties	boolean	Optional	Indicates whether properties added by subclasses of the association class are to be excluded, acting as a restricting filter on the properties included in the returned instances
OperationTimeout	uint32	Mandatory	Operation timeout, as defined in 6.5.2.4
ContinueOnError	boolean	Conditional	Indicates whether the enumeration session should be continued in case of error, as defined in 6.5.2.5 Condition: WBEM protocol supports client side control of continuation on error for pulled instance enumeration operations.
MaxObjectCount	uint32	Mandatory	Maximum number of instances that may be returned by this operation, as defined in 6.5.2.6

Operation Output Parameters:

Generic Name	Generic Type	Requirement	Description
InstanceList	InstanceSpecificationWithPath [ ]	Mandatory	Sequence of instances with their instance paths of the first set of instances
EnumerationContext	EnumerationContext	Mandatory	Enumeration context value, as defined in 6.5.2.1
EndOfSequence	boolean	Mandatory	Indicates end of sequence for the enumeration session, as defined in 6.5.2.2

Description:

The *OpenAssociators* operation establishes and opens an enumeration session for enumerating instances that are associated with the specified source instance. Retrieval of a first set of those instances together with their instance paths may be requested by setting *MaxObjectCount* to a value > 0.

The set of instances to be returned throughout the entire enumeration session shall be determined using the following algorithm:

- Initially, the set of instances to be returned is the set of all instances associated to the source instance specified in *SourceInstancePath*. These associations may be instances of different association classes.

  The result set should not contain any duplicate instances, as defined in 5.8.4. However, different far ends may reference the same instance, and in such cases, the instance shall be contained in the result set once for each such reference.

- If the *AssociationClassName* operation input parameter is not NULL, it acts as a restricting filter on the instances to be returned such that each instance that is associated with the source instance using an association whose creation class or one of its superclasses does not have the name specified in *AssociationClassName*, is removed from the set of instances to be returned. There shall be no validity checking performed for the *AssociationClassName* operation input parameter.

- If the *AssociatedClassName* operation input parameter is not NULL, it acts as a restricting filter on the instances to be returned such that each instance whose creation class or one of its superclasses does not have the name specified in *AssociatedClassName*, is removed from the set of instances to be returned. There shall be no validity checking performed for the *AssociatedClassName* operation input parameter.
- If the *SourceRoleName* operation input parameter is not NULL, it acts as a restricting filter on the instances to be returned such that each instance that is associated with the source instance using an association class that has a role name on the source end that is not the role name specified in *SourceRoleName*, is removed from the set of instances to be returned. There shall be no validity checking performed for the *SourceRoleName* operation input parameter.
- If the *AssociatedRoleName* operation input parameter is not NULL, it acts as a restricting filter on the instances to be returned such that each instance that is associated with the source instance using an association class that has a role name on the end referencing that instance that is not the role name specified in *AssociatedRoleName*, is removed from the set of instances to be returned. There shall be no validity checking performed for the *AssociatedRoleName* operation input parameter.
- If the WBEM protocol supports filter queries for pulled instance enumeration operations (that is, the *FilterQueryString* and *FilterQueryLanguage* operation parameters) and *FilterQueryLanguage* is not NULL, *FilterQueryString* acts as a restricting filter on the instances to be returned such that any instances not selected by the filter query for its result set are removed from the set of instances. The filter query shall query only the class specified in *AssociatedClassName* (e.g., in the CQL FROM-clause). See also 6.5.2.3.

The set of instances to be returned throughout the entire enumeration session should not contain any duplicate instances, as defined in 5.8.4. Because the set of returned instances contains only instances that exist in the same namespace, a determination of duplicate instances can be done on the basis of their model paths only.

The set of instances to be returned in the *InstanceList* operation parameter is the first set of instances from the set of instances to be returned throughout the entire enumeration session, such that no more than *MaxObjectCount* instances are returned. Returning no instances does not imply that the enumeration session has been exhausted. Only the *EndOfSequence* operation output parameter indicates whether the enumeration session has been exhausted.

The set of properties to be included in any returned instances shall be determined using the following algorithm:

- Initially, the set of properties to be included is the set of properties exposed by the creation class of the instance. This includes all the duplicates of any duplicate non-overridden properties.
- If the *IncludedProperties* operation input parameter is supported by the WBEM protocol and if its value is not NULL, it acts as a restricting filter on the properties to be included in the returned instances such that any properties exposed by the creation class of the instance that are not named in that operation parameter are removed from the set of properties to be included. Any duplicate or invalid property names in the *IncludedProperties* operation input parameter shall be ignored. A non-NULL empty *IncludedProperties* list removes all properties from the set of properties to be included.
- If the *ExcludeSubclassProperties* operation input parameter is supported by the WBEM protocol and if its value is TRUE, it acts as a restricting filter on the properties to be included in the returned instances such that any properties not exposed by the class specified in *AssociatedClassName* are removed from the set of properties to be included.

- Conformant WBEM protocols may specify rules that cause properties with a value of NULL to be removed from the set of properties to be included.

Preconditions:

- The instance referenced by *SourceInstancePath* shall exist in the namespace. If this is not satisfied, the operation shall fail, indicating WIPG0213.
- If a filter query is specified,
  - the query language specified in the *FilterQueryLanguage* operation parameter shall be valid. If this is not satisfied, the operation shall fail, indicating WIPG0221.
  - the query specified in the *FilterQueryString* operation parameter shall be a valid query in the query language specified in the *FilterQueryLanguage* operation parameter. If this is not satisfied, the operation shall fail, indicating WIPG0222 or WIPG0223.
  - the *AssociatedClassName* operation input parameter shall be non-NULL. If this is not satisfied, the operation shall fail, indicating WIPG0208.
- The *IncludedProperties* operation parameter, if supported by the WBEM protocol, shall only be specified with a non-NULL value if the *AssociatedClassName* operation input parameter is also non-NULL. If this is not satisfied, the operation shall fail, indicating WIPG0208.
- The *ExcludeSubclassProperties* operation parameter, if supported by the WBEM protocol, shall only be specified with a TRUE value if the *AssociatedClassName* operation input parameter is non-NULL. If this is not satisfied, the operation shall fail, indicating WIPG0208.

NOTE: Specifying a non-NULL value for *AssociatedClassName* ensures that the associated instances have the class specified in *AssociatedClassName* as a common superclass.

Postconditions:

- The enumeration session shall have been established and opened.
- A first set of instances with their instance paths shall have been returned as described in the Description paragraph for this operation.
- Requirements on ACID properties:
  - Atomicity: Required (related to the creation of an enumeration context that is maintained by the WBEM server)
  - Update Consistency: N/A
  - Isolation: Required at the level of single instances, as defined in 5.8.
  - Durability: Required (related to creation of an enumeration context that is maintained by the WBEM server)

Error Messages:

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0240	WBEM service limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0204	Namespace not found	Mandatory	Infrastructure	

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0203	Operation not supported by WBEM service infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0242	Invalid timeout	Mandatory	Infrastructure, class implem.	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure, class implem.	
WIPG0235	Continuation on error not supported	Mandatory	Infrastructure, class implem.	
WIPG0237	Filter queries not supported by WBEM service infrastructure	Optional	Infrastructure	
WIPG0244	Filter queries not supported by class implementation	Optional	Class implem.	
WIPG0221	Unknown query language	Mandatory	Infrastructure, class implem.	
WIPG0222	Query language feature not supported	Mandatory	Infrastructure, class implem.	
WIPG0223	Invalid query	Mandatory	Infrastructure, class implem.	
WIPG0228	Operation not supported by class implementation	Mandatory	Class implem.	
WIPG0213	Instance not found	Mandatory	Class implem.	
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

6.5.6 OpenAssociatorPaths

Purpose:

Establish and open an enumeration session for enumerating the instance paths of instances that are associated with a given source instance, and optionally retrieve a first set of those instance paths.

Operation Input Parameters:

Generic Name	Generic Type	Requirement	Description
SourceInstancePath	InstancePath	Mandatory	Instance path of the source instance (Context Parameter)

Generic Name	Generic Type	Requirement	Description
AssociationClass-Name	ClassName	Mandatory	NULL, or name of the association class, acting as a restricting filter on the returned instance paths
AssociatedClass-Name	ClassName	Mandatory	NULL, or name of the associated class on any far end of the association, acting as a restricting filter on the returned instance paths
SourceRoleName	PropertyName	Mandatory	NULL, or name of the role on the source end of the association, acting as a restricting filter on the returned instance paths
AssociatedRoleName	PropertyName	Mandatory	NULL, or name of the role on any far end of the association, acting as a restricting filter on the returned instance paths
FilterQueryString	QueryString	Conditional	NULL, or query string of a filter query that is acting as an additional restricting filter on the set of returned instance paths, as defined in 6.5.2.3 Condition: WBEM protocol supports filter queries for pulled instance enumeration operations.
FilterQueryLanguage	QueryLanguage	Conditional	NULL, or query language of the filter query specified in *FilterQueryString*, as defined in 6.5.2.3 Condition: WBEM protocol supports filter queries for pulled instance enumeration operations.
OperationTimeout	uint32	Mandatory	Operation timeout, as defined in 6.5.2.4
ContinueOnError	boolean	Conditional	Indicates whether the enumeration session should be continued in case of error, as defined in 6.5.2.5 Condition: WBEM protocol supports client side control of continuation on error for pulled instance enumeration operations.
MaxObjectCount	uint32	Mandatory	Maximum number of instances that may be returned by this operation, as defined in 6.5.2.6

Operation Output Parameters:

Generic Name	Generic Type	Requirement	Description
InstancePathList	InstancePath [ ]	Mandatory	Sequence of instance paths of the first set of instance paths
EnumerationContext	EnumerationContext	Mandatory	Enumeration context value, as defined in 6.5.2.1
EndOfSequence	boolean	Mandatory	Indicates end of sequence for the enumeration session, as defined in 6.5.2.2

Description:

The *OpenAssociatorPaths* operation establishes and opens an enumeration session for enumerating the instance paths of instances that are associated with the specified source instance. Retrieval of a first set of those instance paths may be requested by setting *MaxObjectCount* to a value > 0.

The set of instances of which instance paths are to be returned throughout the entire enumeration session shall be determined using the following algorithm:

- Initially, the set of instances to be returned is the set of all instances associated to the source instance specified in *SourceInstancePath*. These associations may be instances of different association classes.

  The result set should not contain any duplicate instance paths, as defined in 5.8.4. However, different far ends may reference the same instance, and in such cases, the instance path shall be contained in the result set once for each such reference.

- If the *AssociationClassName* operation input parameter is not NULL, it acts as a restricting filter on the instances to be returned such that each instance that is associated with the source instance using an association whose creation class or one of its superclasses does not have the name specified in *AssociationClassName*, is removed from the set of instances to be returned. There shall be no validity checking performed for the *AssociationClassName* operation input parameter.

- If the *AssociatedClassName* operation input parameter is not NULL, it acts as a restricting filter on the instances to be returned such that each instance whose creation class or one of its superclasses does not have the name specified in *AssociatedClassName*, is removed from the set of instances to be returned. There shall be no validity checking performed for the *AssociatedClassName* operation input parameter.

- If the *SourceRoleName* operation input parameter is not NULL, it acts as a restricting filter on the instances to be returned such that each instance that is associated with the source instance using an association class that has a role name on the source end that is not the role name specified in *SourceRoleName*, is removed from the set of instances to be returned. There shall be no validity checking performed for the *SourceRoleName* operation input parameter.

- If the *AssociatedRoleName* operation input parameter is not NULL, it acts as a restricting filter on the instances to be returned such that each instance that is associated with the source instance using an association class that has a role name on the end referencing that instance that is not the role name specified in *AssociatedRoleName*, is removed from the set of instances to be returned. There shall be no validity checking performed for the *AssociatedRoleName* operation input parameter.

- If the WBEM protocol supports filter queries for pulled instance enumeration operations (that is, the *FilterQueryString* and *FilterQueryLanguage* operation parameters) and *FilterQueryLanguage* is not NULL, *FilterQueryString* acts as a restricting filter on the instances to be returned such that any instances not selected by the filter query for its result set are removed from the set of instances. The filter query shall query only the class specified in *AssociatedClassName* (e.g., in the CQL FROM-clause). See also 6.5.2.3.

The set of instance paths to be returned throughout the entire enumeration session should not contain any duplicate instance paths, as defined in 5.8.4. Because the set of returned instance paths references only instances in the same namespace, a determination of duplicate instance paths can be done on the basis of their model paths only.

The set of instance paths to be returned in the *InstancePathList* operation parameter is the first set of instance paths from the set of instance paths to be returned throughout the entire enumeration session, such that no more than *MaxObjectCount* instance paths are returned. Returning no instance

paths does not imply that the enumeration session has been exhausted. Only the *EndOfSequence* operation output parameter indicates whether the enumeration session has been exhausted.

Preconditions:

- The instance referenced by *SourceInstancePath* shall exist in the namespace. If this is not satisfied, the operation shall fail, indicating WIPG0213.
- If a filter query is specified,
  - the query language specified in the *FilterQueryLanguage* operation parameter shall be valid. If this is not satisfied, the operation shall fail, indicating WIPG0221.
  - the query specified in the *FilterQueryString* operation parameter shall be a valid query in the query language specified in the *FilterQueryLanguage* operation parameter. If this is not satisfied, the operation shall fail, indicating WIPG0222 or WIPG0223.
  - the *AssociatedClassName* operation input parameter shall be non-NULL. If this is not satisfied, the operation shall fail, indicating WIPG0208.

NOTE: Specifying a non-NULL value for *AssociatedClassName* ensures that the associated instances have the class specified in *AssociatedClassName* as a common superclass.

Postconditions:

- The enumeration session shall have been established and opened.
- A first set of instance paths shall have been returned as described in the Description paragraph for this operation.
- Requirements on ACID properties:
  - Atomicity: Required (related to the creation of an enumeration context that is maintained by the WBEM server)
  - Update Consistency: N/A
  - Isolation: Required at the level of single instances, as defined in 5.8.
  - Durability: Required (related to creation of an enumeration context that is maintained by the WBEM server)

Error Messages:

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0240	WBEM service limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0242	Invalid timeout	Mandatory	Infrastructure, class implem.	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure, class implem.	
WIPG0235	Continuation on error not supported	Mandatory	Infrastructure, class implem.	
WIPG0237	Filter queries not supported by WBEM service infrastructure	Optional	Infrastructure	
WIPG0244	Filter queries not supported by class implementation	Optional	Class implem.	
WIPG0221	Unknown query language	Mandatory	Infrastructure, class implem.	
WIPG0222	Query language feature not supported	Mandatory	Infrastructure, class implem.	
WIPG0223	Invalid query	Mandatory	Infrastructure, class implem.	
WIPG0228	Operation not supported by class implementation	Mandatory	Class implem.	
WIPG0213	Instance not found	Mandatory	Class implem.	
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

6.5.7 OpenReferences

Purpose:

Establish and open an enumeration session for enumerating the association instances that reference a given source instance, and optionally retrieve a first set of those instances.

Operation Input Parameters:

Generic Name	Generic Type	Requirement	Description
SourceInstancePath	InstancePath	Mandatory	Instance path of the source instance (Context Parameter)
AssociationClass-Name	ClassName	Mandatory	NULL, or name of the association class, acting as a restricting filter on the returned instances
AssociatedClass-Name	ClassName	Mandatory	NULL, or name of the associated class on any far end of the association, acting as a restricting filter on the returned instances

Generic Name	Generic Type	Requirement	Description
SourceRoleName	PropertyName	Mandatory	NULL, or name of the role on the source end of the association, acting as a restricting filter on the returned instances
AssociatedRoleName	PropertyName	Mandatory	NULL, or name of the role on any far end of the association, acting as a restricting filter on the returned instances
FilterQueryString	QueryString	Conditional	NULL, or query string of a filter query that is acting as an additional restricting filter on the set of returned instances, as defined in 6.5.2.3 Condition: WBEM protocol supports filter queries for pulled instance enumeration operations.
FilterQueryLanguage	QueryLanguage	Conditional	NULL, or query language of the filter query specified in *FilterQueryString*, as defined in 6.5.2.3 Condition: WBEM protocol supports filter queries for pulled instance enumeration operations.
IncludeClassOrigin	boolean	Conditional	Indicates whether class origin information for any returned properties is to be included, as defined in 6.2.1 Condition: WBEM protocol supports client side control of returning class origin information.
IncludedProperties	PropertyName [ ]	Optional	NULL, or unordered set of property names to be included, acting as a restricting filter on the properties included in the returned instances
ExcludeSubclass-Properties	boolean	Optional	Indicates whether properties added by subclasses of the association class are to be excluded, acting as a restricting filter on the properties included in the returned instances
OperationTimeout	uint32	Mandatory	Operation timeout, as defined in 6.5.2.4
ContinueOnError	boolean	Conditional	Indicates whether the enumeration session should be continued in case of error, as defined in 6.5.2.5 Condition: WBEM protocol supports client side control of continuation on error for pulled instance enumeration operations.
MaxObjectCount	uint32	Mandatory	Maximum number of instances that may be returned by this operation, as defined in 6.5.2.6

Operation Output Parameters:

Generic Name	Generic Type	Requirement	Description
InstanceList	InstanceSpecificationWithPath [ ]	Mandatory	Sequence of instances with their instance paths of the first set of instances
EnumerationContext	EnumerationContext	Mandatory	Enumeration context value, as defined in 6.5.2.1

Generic Name	Generic Type	Requirement	Description
EndOfSequence	boolean	Mandatory	Indicates end of sequence for the enumeration session, as defined in 6.5.2.2

Description:

The *OpenReferences* operation establishes and opens an enumeration session for enumerating the association instances that reference the specified source instance. Retrieval of a first set of those instances together with their instance paths may be requested by setting *MaxObjectCount* to a value > 0.

The set of instances to be returned throughout the entire enumeration session shall be determined using the following algorithm:

- Initially, the set of instances to be returned is the set of all instances referencing the source instance specified in *SourceInstancePath*. These associations may be instances of different association classes.
- If the *AssociationClassName* operation input parameter is not NULL, it acts as a restricting filter on the instances to be returned such that each association instance whose creation class or one of its superclasses does not have the name specified in *AssociationClassName*, is removed from the set of instances to be returned. There shall be no validity checking performed for the *AssociationClassName* operation input parameter.
- If the *AssociatedClassName* operation input parameter is not NULL, it acts as a restricting filter on the instances to be returned such that each association instance whose creation class has a set of far ends none of which is referencing a class where that class or one of its superclasses has the name specified in *AssociatedClassName*, is removed from the set of instances to be returned. There shall be no validity checking performed for the *AssociatedClassName* operation input parameter.
- If the *SourceRoleName* operation input parameter is not NULL, it acts as a restricting filter on the instances to be returned such that each association instance whose creation class does not have the role name specified in *SourceRoleName* on the end referencing the source instance, is removed from the set of instances to be returned. There shall be no validity checking performed for the *SourceRoleName* operation input parameter.
- If the *AssociatedRoleName* operation input parameter is not NULL, it acts as a restricting filter on the instances to be returned such that each association instance whose creation class has a set of far ends none of which has the role name specified in *AssociatedRoleName*, is removed from the set of instances to be returned. There shall be no validity checking performed for the *AssociatedRoleName* operation input parameter.
- If the WBEM protocol supports filter queries for pulled instance enumeration operations (that is, the *FilterQueryString* and *FilterQueryLanguage* operation parameters) and *FilterQueryLanguage* is not NULL, *FilterQueryString* acts as a restricting filter on the instances to be returned such that any instances not selected by the filter query for its result set are removed from the set of instances. The filter query shall query only the class specified in *AssociationClassName* (e.g., in the CQL FROM-clause). See also 6.5.2.3.

The set of instances to be returned throughout the entire enumeration session should not contain any duplicate instances, as defined in 5.8.4. Because the set of returned instances contains only instances that exist in the same namespace, so any determination of duplicate instances (for example by a WBEM client) may be done on the basis of their model paths.

The set of instances to be returned in the *InstanceList* operation parameter is the first set of instances from the set of instances to be returned throughout the entire enumeration session, such

that no more than *MaxObjectCount* instances are returned. Returning no instances does not imply that the enumeration session has been exhausted. Only the *EndOfSequence* operation output parameter indicates whether the enumeration session has been exhausted.

The set of properties to be included in any returned instances shall be determined using the following algorithm:

- Initially, the set of properties to be included is the set of properties exposed by the creation class of the instance. This includes all the duplicates of any duplicate non-overridden properties.
- If the *IncludedProperties* operation input parameter is supported by the WBEM protocol and if its value is not NULL, it acts as a restricting filter on the properties to be included in the returned instances such that any properties exposed by the creation class of the instance that are not named in that operation parameter are removed from the set of properties to be included. Any duplicate or invalid property names in the *IncludedProperties* operation input parameter shall be ignored. A non-NULL empty *IncludedProperties* list removes all properties from the set of properties to be included.
- If the *ExcludeSubclassProperties* operation input parameter is supported by the WBEM protocol and if its value is TRUE, it acts as a restricting filter on the properties to be included in the returned instances such that any properties not exposed by the class specified in *AssociationClassName* are removed from the set of properties to be included.
- Conformant WBEM protocols may specify rules that cause properties with a value of NULL to be removed from the set of properties to be included.

Preconditions:

- The instance referenced by *SourceInstancePath* shall exist in the namespace. If this is not satisfied, the operation shall fail, indicating WIPG0213.
- If a filter query is specified,
  - the query language specified in the *FilterQueryLanguage* operation parameter shall be valid. If this is not satisfied, the operation shall fail, indicating WIPG0221.
  - the query specified in the *FilterQueryString* operation parameter shall be a valid query in the query language specified in the *FilterQueryLanguage* operation parameter. If this is not satisfied, the operation shall fail, indicating WIPG0222 or WIPG0223.
  - the *AssociationClassName* operation input parameter shall be non-NULL. If this is not satisfied, the operation shall fail, indicating WIPG0208.
- The *IncludedProperties* operation parameter, if supported by the WBEM protocol, shall only be specified with a non-NULL value if the *AssociationClassName* operation input parameter is also non-NULL. If this is not satisfied, the operation shall fail, indicating WIPG0208.
- The *ExcludeSubclassProperties* operation parameter, if supported by the WBEM protocol, shall only be specified with a TRUE value if the *AssociationClassName* operation input parameter is non-NULL. If this is not satisfied, the operation shall fail, indicating WIPG0208.

NOTE: Specifying a non-NULL value for *AssociationClassName* ensures that the association instances have the class specified in *AssociationClassName* as a common superclass.

Postconditions:

- The enumeration session shall have been established and opened.
- A first set of instances with their instance paths shall have been returned as described in the Description paragraph for this operation.
- Requirements on ACID properties:

- Atomicity: Required (related to the creation of an enumeration context that is maintained by the WBEM server)
- Update Consistency: N/A
- Isolation: Required at the level of single instances, as defined in 5.8.
- Durability: Required (related to creation of an enumeration context that is maintained by the WBEM server)

Error Messages:

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0240	WBEM service limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0242	Invalid timeout	Mandatory	Infrastructure, class implem.	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure, class implem.	
WIPG0235	Continuation on error not supported	Mandatory	Infrastructure, class implem.	
WIPG0237	Filter queries not supported by WBEM service infrastructure	Optional	Infrastructure	
WIPG0244	Filter queries not supported by class implementation	Optional	Class implem.	
WIPG0221	Unknown query language	Mandatory	Infrastructure, class implem.	
WIPG0222	Query language feature not supported	Mandatory	Infrastructure, class implem.	
WIPG0223	Invalid query	Mandatory	Infrastructure, class implem.	
WIPG0228	Operation not supported by class implementation	Mandatory	Class implem.	
WIPG0213	Instance not found	Mandatory	Class implem.	
WIPG0243	Timeout	Optional	Infrastructure, class implem.	

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

6.5.8 OpenReferencePaths

Purpose:

Establish and open an enumeration session for enumerating the instance paths of association instances that reference a given source instance, and optionally retrieve a first set of those instance paths.

Operation Input Parameters:

Generic Name	Generic Type	Requirement	Description
SourceInstancePath	InstancePath	Mandatory	Instance path of the source instance (Context Parameter)
AssociationClass-Name	ClassName	Mandatory	NULL, or name of the association class, acting as a restricting filter on the returned instance paths
AssociatedClass-Name	ClassName	Mandatory	NULL, or name of the associated class on any far end of the association, acting as a restricting filter on the returned instance paths
SourceRoleName	PropertyName	Mandatory	NULL, or name of the role on the source end of the association, acting as a restricting filter on the returned instance paths
AssociatedRoleName	PropertyName	Mandatory	NULL, or name of the role on any far end of the association, acting as a restricting filter on the returned instance paths
FilterQueryString	QueryString	Conditional	NULL, or query string of a filter query that is acting as an additional restricting filter on the set of returned instance paths, as defined in 6.5.2.3 Condition: WBEM protocol supports filter queries for pulled instance enumeration operations.
FilterQueryLanguage	QueryLanguage	Conditional	NULL, or query language of the filter query specified in *FilterQueryString*, as defined in 6.5.2.3 Condition: WBEM protocol supports filter queries for pulled instance enumeration operations.
OperationTimeout	uint32	Mandatory	Operation timeout, as defined in 6.5.2.4
ContinueOnError	boolean	Conditional	Indicates whether the enumeration session should be continued in case of error, as defined in 6.5.2.5 Condition: WBEM protocol supports client side control of continuation on error for pulled instance enumeration operations.

Generic Name	Generic Type	Requirement	Description
MaxObjectCount	uint32	Mandatory	Maximum number of instance paths that may be returned by this operation, as defined in 6.5.2.6

Operation Output Parameters:

Generic Name	Generic Type	Requirement	Description
InstancePathList	InstancePath [ ]	Mandatory	Sequence of instance paths of the first set of instance paths
EnumerationContext	EnumerationContext	Mandatory	Enumeration context value, as defined in 6.5.2.1
EndOfSequence	boolean	Mandatory	Indicates end of sequence for the enumeration session, as defined in 6.5.2.2

Description:

The *OpenReferencePaths* operation establishes and opens an enumeration session for enumerating the instance paths of association instances that reference the specified source instance. Retrieval of a first set of those instance paths may be requested by setting *MaxObjectCount* to a value > 0.

The set of instances of which instance paths are to be returned throughout the entire enumeration session shall be determined using the following algorithm:

- Initially, the set of instances to be returned is the set of all instances referencing the source instance specified in *SourceInstancePath*. These associations may be instances of different association classes.
- If the *AssociationClassName* operation input parameter is not NULL, it acts as a restricting filter on the instances to be returned such that each association instance whose creation class or one of its superclasses does not have the name specified in *AssociationClassName*, is removed from the set of instances to be returned. There shall be no validity checking performed for the *AssociationClassName* operation input parameter.
- If the *AssociatedClassName* operation input parameter is not NULL, it acts as a restricting filter on the instances to be returned such that each association instance whose creation class has a set of far ends none of which is referencing a class where that class or one of its superclasses has the name specified in *AssociatedClassName*, is removed from the set of instances to be returned. There shall be no validity checking performed for the *AssociatedClassName* operation input parameter.
- If the *SourceRoleName* operation input parameter is not NULL, it acts as a restricting filter on the instances to be returned such that each association instance whose creation class does not have the role name specified in *SourceRoleName* on the end referencing the source instance, is removed from the set of instances to be returned. There shall be no validity checking performed for the *SourceRoleName* operation input parameter.
- If the *AssociatedRoleName* operation input parameter is not NULL, it acts as a restricting filter on the instances to be returned such that each association instance whose creation class has a set of far ends none of which has the role name specified in *AssociatedRoleName*, is removed from the set of instances to be returned. There shall be no validity checking performed for the *AssociatedRoleName* operation input parameter.
- If the WBEM protocol supports filter queries for pulled instance enumeration operations (that is, the *FilterQueryString* and *FilterQueryLanguage* operation parameters) and

FilterQueryLanguage is not NULL, *FilterQueryString* acts as a restricting filter on the instances to be returned such that any instances not selected by the filter query for its result set are removed from the set of instances. The filter query shall query only the class specified in *AssociationClassName* (e.g., in the CQL FROM-clause). See also 6.5.2.3.

The set of instance paths to be returned throughout the entire enumeration session should not contain any duplicate instance paths, as defined in 5.8.4. Because the set of returned instance paths references only instances that exist in the same namespace, a determination of duplicate instance paths can be done on the basis of their model paths only.

The set of instance paths to be returned in the *InstancePathList* operation parameter is the first set of instance paths from the set of instance paths to be returned throughout the entire enumeration session, such that no more than *MaxObjectCount* instances are returned. Returning no instance paths does not imply that the enumeration session has been exhausted. Only the *EndOfSequence* operation output parameter indicates whether the enumeration session has been exhausted.

Preconditions:

- The instance referenced by *SourceInstancePath* shall exist in the namespace. If this is not satisfied, the operation shall fail, indicating WIPG0213.
- If a filter query is specified,
  - the query language specified in the *FilterQueryLanguage* operation parameter shall be valid. If this is not satisfied, the operation shall fail, indicating WIPG0221.
  - the query specified in the *FilterQueryString* operation parameter shall be a valid query in the query language specified in the *FilterQueryLanguage* operation parameter. If this is not satisfied, the operation shall fail, indicating WIPG0222 or WIPG0223.
  - the *AssociationClassName* operation input parameter shall be non-NULL. If this is not satisfied, the operation shall fail, indicating WIPG0208.

NOTE: Specifying a non-NULL value for *AssociationClassName* ensures that the association instances have the class specified in *AssociationClassName* as a common superclass.

Postconditions:

- The enumeration session shall have been established and opened.
- A first set of instance paths shall have been returned as described in the Description paragraph for this operation.
- Requirements on ACID properties:
  - Atomicity: Required (related to the creation of an enumeration context that is maintained by the WBEM server)
  - Update Consistency: N/A
  - Isolation: Required at the level of single instances, as defined in 5.8.
  - Durability: Required (related to creation of an enumeration context that is maintained by the WBEM server)

Error Messages:

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0240	WBEM service limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0242	Invalid timeout	Mandatory	Infrastructure, class implem.	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure, class implem.	
WIPG0235	Continuation on error not supported	Mandatory	Infrastructure, class implem.	
WIPG0237	Filter queries not supported by WBEM service infrastructure	Optional	Infrastructure	
WIPG0244	Filter queries not supported by class implementation	Optional	Class implem.	
WIPG0221	Unknown query language	Mandatory	Infrastructure, class implem.	
WIPG0222	Query language feature not supported	Mandatory	Infrastructure, class implem.	
WIPG0223	Invalid query	Mandatory	Infrastructure, class implem.	
WIPG0228	Operation not supported by class implementation	Mandatory	Class implem.	
WIPG0213	Instance not found	Mandatory	Class implem.	
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

6.5.9 OpenQueryInstances

Purpose:

Establish and open an enumeration session for enumerating the instances of a query result, and optionally retrieve a first set of instances.

2161 **Operation Input Parameters:**
2162

Generic Name	Generic Type	Requirement	Description
NamespacePath	NamespacePath	Mandatory	Namespace path of the CIM namespace the query is executed in (Context Parameter)
QueryString	QueryString	Mandatory	Query string of a query that defines the set of instances to be returned
QueryLanguage	QueryLanguage	Mandatory	Query language of the query specified in *QueryString*
ReturnQueryResult-Class	boolean	Mandatory	Indicates whether a class definition of the query result should be returned in *QueryResultClass*
OperationTimeout	uint32	Mandatory	Operation timeout, as defined in 6.5.2.4
ContinueOnError	boolean	Conditional	Indicates whether the enumeration session should be continued in case of error, as defined in 6.5.2.5 Condition: WBEM protocol supports client side control of continuation on error for pulled instance enumeration operations.
MaxObjectCount	uint32	Mandatory	Maximum number of instances that may be returned by this operation, as defined in 6.5.2.6

2163 **Operation Output Parameters:**
2164

Generic Name	Generic Type	Requirement	Description
InstanceList	InstanceSpecification [ ]	Mandatory	Sequence of instances of the first set of instances
QueryResultClass	ClassSpecification	Mandatory	Representation of a class definition for the query result
EnumerationContext	EnumerationContext	Mandatory	Enumeration context value, as defined in 6.5.2.1
EndOfSequence	boolean	Mandatory	Indicates end of sequence for the enumeration session, as defined in 6.5.2.2

2165 **Description:**

2166 The *OpenQueryInstances* operation establishes and opens an enumeration session for enumerating
2167 the instances representing the result of the query specified in *QueryString* in the CIM namespace
2168 referenced by *NamespacePath*. Retrieval of a first set of those instances may be requested by
2169 setting *MaxObjectCount* to a value > 0.

2170 The set of instances to be returned in the *InstanceList* operation parameter is the first set of
2171 instances from the set of instances to be returned throughout the entire enumeration session, such
2172 that no more than *MaxObjectCount* instances are returned. Returning no instances in the
2173 *InstanceList* operation parameter does not imply that the enumeration session has been exhausted.
2174 Only the *EndOfSequence* operation output parameter indicates whether the enumeration session
2175 has been exhausted.

The returned instances are only representations of instances that have no corresponding addressable instances residing in the WBEM server.

If *QueryLanguage* is not NULL, it shall specify a valid query language and *QueryString* shall be a valid query in that query language. Neither the query language nor the format of the filter query is defined by this specification. Conformant WBEM protocols shall specify a mechanism for determining the set of query languages that are valid for *QueryLanguage*. The simplest way to do this is to list the set of valid query languages.

The value of the *ReturnQueryResultClass* operation input parameter controls whether or not a class definition is returned in the *QueryResultClass* operation output parameter. If FALSE, then *QueryResultClass* shall be NULL. If TRUE, then the value of *QueryResultClass* shall be a class definition that defines the properties of each instance of the query result. The name of this class shall be CIM_QueryResult. This class is only a representation of a class that has no corresponding addressable class residing in the WBEM server.

Preconditions:

- The CIM namespace referenced by *NamespacePath* shall exist. If this is not satisfied, the operation shall fail, indicating WIPG0204.
- The query language specified in the *QueryLanguage* operation parameter shall be a valid query language. If this is not satisfied, the operation shall fail, indicating WIPG0221.
- The query specified in the *QueryString* operation parameter shall be a valid query in the query language specified in the *QueryLanguage* operation parameter. If this is not satisfied, the operation shall fail, indicating WIPG0222 or WIPG0223.

Postconditions:

- The enumeration session shall have been established and opened.
- A first set of instances shall have been returned as described in the Description paragraph for this operation.
- Requirements on ACID properties:
  - Atomicity: Required (related to the creation of an enumeration context that is maintained by the WBEM server)
  - Update Consistency: N/A
  - Isolation: Required at the level of single instances, as defined in 5.8.
  - Durability: Required (related to creation of an enumeration context that is maintained by the WBEM server)

Error Messages:

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0240	WBEM service limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service infrastructure	Mandatory	Infrastructure	

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0242	Invalid timeout	Mandatory	Infrastructure, class implem.	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure, class implem.	
WIPG0235	Continuation on error not supported	Mandatory	Infrastructure, class implem.	
WIPG0221	Unknown query language	Mandatory	Infrastructure, class implem.	
WIPG0222	Query language feature not supported	Mandatory	Infrastructure, class implem.	
WIPG0223	Invalid query	Mandatory	Infrastructure, class implem.	
WIPG0228	Operation not supported by class implementation	Mandatory	Class implem.	
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

6.5.10 Common operation parameters for the pull operations

This subclause defines commonly used operation parameters for the Pull operations. The description of the individual Pull operations references these operation parameters as appropriate. However, not every Pull operation uses every one of these common operation parameters.

6.5.10.1 NamespacePath

The *NamespacePath* operation input parameter references the CIM namespace identified by the context parameter of the Open operation that established and opened the enumeration session.

6.5.10.2 EnumerationContext

The *EnumerationContext* operation input/output parameter is the enumeration context value representing the enumeration session to be used.

Support for the *EnumerationContext* operation parameter in a conformant WBEM protocol is mandatory.

When invoking the Pull operation, the enumeration session represented by *EnumerationContext* shall be open. The enumeration session shall have been established using one of the Open operations whose type of enumerated element matches the Pull operation. For the first Pull operation on an enumeration session, the value of *EnumerationContext* shall be the enumeration context value returned by a successful Open operation that established and opened that enumeration session. For any subsequent Pull operations on that enumeration session, the value of *EnumerationContext* shall be the value of *EnumerationContext* as returned by the previous Pull operation on the same enumeration session.

After completing the Pull operation, the enumeration session represented by *EnumerationContext* shall be open or closed.

6.5.10.3 EndOfSequence

The *EndOfSequence* operation output parameter when used in Pull operations behaves as defined in 6.5.2.2

6.5.10.4 MaxObjectCount

The *MaxObjectCount* operation input parameter when used in Pull operations behaves as defined in 6.5.2.6.

6.5.11 PullInstancesWithPath

Purpose:

Retrieve the next set of instances together with their instance paths from an open enumeration session.

Operation Input Parameters:

Generic Name	Generic Type	Requirement	Description
NamespacePath	NamespacePath	Mandatory	Namespace path of the CIM namespace, as defined in 6.5.10.1 (Context Parameter)
EnumerationContext	EnumerationContext	Mandatory	Enumeration context value, as defined in 6.5.10.2
MaxObjectCount	uint32	Mandatory	Maximum number of instances that may be returned by this operation, as defined in 6.5.10.4

Operation Output Parameters:

Generic Name	Generic Type	Requirement	Description
InstanceList	InstanceSpecificationWithPath [ ]	Mandatory	Sequence of instances with their instance paths of the retrieved set of instances
EnumerationContext	EnumerationContext	Mandatory	Enumeration context value, as defined in 6.5.10.2
EndOfSequence	boolean	Mandatory	Indicates end of sequence for the enumeration session, as defined in 6.5.10.3

Description:

The *PullInstancesWithPath* operation retrieves the next set of instances together with their instance paths from an open enumeration session.

The enumeration session shall have been established using one of the following operations:

- OpenEnumerateInstances

- OpenAssociators
- OpenReferences

The set of instances to be returned in the *InstanceList* operation parameter is the next set of instances from the set of instances to be returned throughout the entire enumeration session, such that no more than *MaxObjectCount* instances are returned. Returning no instances does not imply that the enumeration session has been exhausted. Only the *EndOfSequence* operation output parameter indicates whether the enumeration session has been exhausted.

The set of properties to be included in any retrieved instances shall be the as determined using the Open operation that established the enumeration session.

Preconditions:

- The enumeration session identified by *EnumerationContext* shall be open. If this is not satisfied, the operation shall fail, indicating WIPG0241.
- The value of *EnumerationContext* shall be the enumeration context value returned by the previous Open or Pull operation on the same enumeration session. If this is not satisfied, the operation shall fail, indicating WIPG0241.

Postconditions:

- The set of instances with their instance paths shall have been returned as described in the Description paragraph for this operation.
- Requirements on ACID properties:
  - Atomicity: Required (related to updates to an enumeration context that is maintained by the WBEM server)
  - Update Consistency: N/A
  - Isolation: Required at the level of single instances, as defined in 5.8.
  - Durability: Required (related to updates to an enumeration context that is maintained by the WBEM server)

Error Messages:

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0240	WBEM service limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0203	Operation not supported by WBEM service infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure, class implem.	
WIPG0228	Operation not supported by class implementation	Mandatory	Class implem.	
WIPG0241	Invalid enumeration context	Mandatory	Class implem.	
WIPG0238	Pull operation has been abandoned due to enumeration context closure	Mandatory	Class implem.	
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

6.5.12 PullInstancePaths

Purpose:

Retrieve the next set of instance paths from an open enumeration session.

Operation Input Parameters:

Generic Name	Generic Type	Requirement	Description
NamespacePath	NamespacePath	Mandatory	Namespace path of the CIM namespace, as defined in 6.5.10.1 (Context Parameter)
EnumerationContext	EnumerationContext	Mandatory	Enumeration context value, as defined in 6.5.10.2
MaxObjectCount	uint32	Mandatory	Maximum number of instance paths that may be returned by this operation, as defined in 6.5.10.4

Operation Output Parameters:

Generic Name	Generic Type	Requirement	Description
InstancePathList	InstancePath [ ]	Mandatory	Sequence of retrieved instance paths
EnumerationContext	EnumerationContext	Mandatory	Enumeration context value, as defined in 6.5.10.2
EndOfSequence	boolean	Mandatory	Indicates end of sequence for the enumeration session, as defined in 6.5.10.3

Description:

The *PullInstancePaths* operation retrieves the next set of instance paths from an open enumeration session.

The enumeration session shall have been established using one of the following operations:

- OpenEnumerateInstancePaths
- OpenAssociatorPaths
- OpenReferencePaths

The set of instance paths to be returned in the *InstancePathList* operation parameter is the next set of instance paths from the set of instance paths to be returned throughout the entire enumeration session, such that no more than *MaxObjectCount* instance paths are returned. Returning no instance paths does not imply that the enumeration session has been exhausted. Only the *EndOfSequence* operation output parameter indicates whether the enumeration session has been exhausted.

Preconditions:

- The enumeration session identified by *EnumerationContext* shall be open. If this is not satisfied, the operation shall fail, indicating WIPG0241.
- The value of *EnumerationContext* shall be the enumeration context value returned by the previous Open or Pull operation on the same enumeration session. If this is not satisfied, the operation shall fail, indicating WIPG0241.

Postconditions:

- The set of instance paths shall have been returned as described in the Description paragraph for this operation.
- Requirements on ACID properties:
  - Atomicity: Required (related to updates to an enumeration context that is maintained by the WBEM server)
  - Update Consistency: N/A
  - Isolation: Required at the level of single instances, as defined in 5.8.
  - Durability: Required (related to updates to an enumeration context that is maintained by the WBEM server)

Error Messages:

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0240	WBEM service limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0203	Operation not supported by WBEM service infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure, class implem.	

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0228	Operation not supported by class implementation	Mandatory	Class implem.	
WIPG0241	Invalid enumeration context	Mandatory	Class implem.	
WIPG0238	Pull operation has been abandoned due to enumeration context closure	Mandatory	Class implem.	
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

6.5.13 PullInstances

Purpose:

Retrieve the next set of instances from an open enumeration session.

Operation Input Parameters:

Generic Name	Generic Type	Requirement	Description
NamespacePath	NamespacePath	Mandatory	Namespace path of the CIM namespace, as defined in 6.5.10.1 (Context Parameter)
EnumerationContext	EnumerationContext	Mandatory	Enumeration context value, as defined in 6.5.10.2
MaxObjectCount	uint32	Mandatory	Maximum number of instances that may be returned by this operation, as defined in 6.5.10.4

Operation Output Parameters:

Generic Name	Generic Type	Requirement	Description
InstanceList	InstanceSpecification [ ]	Mandatory	Sequence of retrieved instances
EnumerationContext	EnumerationContext	Mandatory	Enumeration context value, as defined in 6.5.10.2
EndOfSequence	boolean	Mandatory	Indicates end of sequence for the enumeration session, as defined in 6.5.10.3

Description:

The *PullInstancesWithPath* operation retrieves the next set of instances without their instance paths from an open enumeration session.

The enumeration session shall have been established using one of the following operations:

- *OpenQueryInstances*

The set of instances to be returned in the *InstanceList* operation parameter is the next set of instances from the set of instances to be returned throughout the entire enumeration session, such that no more than *MaxObjectCount* instances are returned. Returning no instances does not imply that the enumeration session has been exhausted. Only the *EndOfSequence* operation output parameter indicates whether the enumeration session has been exhausted.

The set of properties to be included in any retrieved instances shall be the as determined using the Open operation that established the enumeration session.

Preconditions:

- The enumeration session identified by *EnumerationContext* shall be open. If this is not satisfied, the operation shall fail, indicating WIPG0241.
- The value of *EnumerationContext* shall be the enumeration context value returned by the previous Open or Pull operation on the same enumeration session. If this is not satisfied, the operation shall fail, indicating WIPG0241.

Postconditions:

- The set of instances shall have been returned as described in the Description paragraph for this operation.
- Requirements on ACID properties:
  - Atomicity: Required (related to updates to an enumeration context that is maintained by the WBEM server)
  - Update Consistency: N/A
  - Isolation: Required at the level of single instances, as defined in 5.8.
  - Durability: Required (related to updates to an enumeration context that is maintained by the WBEM server)

Error Messages:

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0240	WBEM service limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0203	Operation not supported by WBEM service infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure, class implem.	
WIPG0228	Operation not supported by class implementation	Mandatory	Class implem.	

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0241	Invalid enumeration context	Mandatory	Class implem.	
WIPG0238	Pull operation has been abandoned due to enumeration context closure	Mandatory	Class implem.	
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

6.5.14 CloseEnumeration

Purpose:

Close an open enumeration session.

Operation Input Parameters:

Generic Name	Generic Type	Requirement	Description
NamespacePath	NamespacePath	Mandatory	Namespace path of the CIM namespace, as defined in 6.5.10.1 (Context Parameter)
EnumerationContext	EnumerationContext	Mandatory	Enumeration context value, as defined in 6.5.10.2

Operation Output Parameters:

None.

Description:

The *CloseEnumeration* operation closes the open enumeration session identified by *EnumerationContext*.

The enumeration session shall have been established using any of the Open operations.

Enumeration sessions are closed implicitly when exhausted, so this operation only needs to be used when terminating an enumeration sequence before it is exhausted.

Preconditions:

- The enumeration session identified by *EnumerationContext* shall be open. If this is not satisfied, the operation shall fail, indicating WIPG0241.
- The value of *EnumerationContext* shall be the enumeration context value returned by the previous Open or Pull operation on the same enumeration session. If this is not satisfied, the operation shall fail, indicating WIPG0241.

Postconditions:

- The enumeration session identified by *EnumerationContext* is closed.
- Requirements on ACID properties:

- Atomicity: Required (related to updates to or deletion of an enumeration context that is maintained by the WBEM server)
- Update Consistency: N/A
- Isolation: Required
- Durability: Required (related to updates to or deletion of an enumeration context that is maintained by the WBEM server)

Error Messages:

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0203	Operation not supported by WBEM service infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure, class implem.	
WIPG0228	Operation not supported by class implementation	Mandatory	Class implem.	
WIPG0241	Invalid enumeration context	Mandatory	Class implem.	
WIPG0239	Pull operation cannot be abandoned	Mandatory	Class implem.	
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

6.5.15 EnumerationCount

Purpose:

Estimate the total number of remaining items in an open enumeration session.

Operation Input Parameters:

Generic Name	Generic Type	Requirement	Description
NamespacePath	NamespacePath	Mandatory	Namespace path of the CIM namespace, as defined in 6.5.10.1 (Context Parameter)
EnumerationContext	EnumerationContext	Mandatory	Enumeration context value, as defined in 6.5.10.2

2385 **Operation Output Parameters:**
2386

Generic Name	Generic Type	Requirement	Description
EnumerationCount	uint64	Mandatory	NULL, or estimated number of remaining items

2387 **Description:**

2388 The *EnumerationCount* operation estimates the total number of remaining items in the open
2389 enumeration session identified by *EnumerationContext*.

2390 The enumeration session shall have been established using any of the Open operations.

2391 If not NULL, the *EnumerationCount* operation output parameter is an estimated count of the number
2392 of items remaining to be retrieved with subsequent Pull operations. Thus, executing this operation
2393 immediately after opening the enumeration session provides an estimate of the total number of items
2394 that will be returned in the enumeration set.

2395 If the WBEM server cannot or will not return an estimated count, it may respond with success and
2396 the NULL value in the *EnumerationCount* operation output parameter.

2397 This mechanism is intended to assist WBEM clients in determining the overall size of an
2398 enumeration set and of the number of items remaining in the enumeration session. However,
2399 because it is an estimate and not an exact number, it should not be used for determining the end of
2400 an enumeration sequence, i.e., in place of the *EndOfSequence* operation output parameter on Open
2401 and Pull operations.

2402 **Preconditions:**

2403 • The enumeration session identified by *EnumerationContext* shall be open. If this is not satisfied,
2404 the operation shall fail, indicating WIPG0241.

2405 • The value of *EnumerationContext* shall be the enumeration context value returned by the
2406 previous Open or Pull operation on the same enumeration session. If this is not satisfied, the
2407 operation shall fail, indicating WIPG0241.

2408 **Postconditions:**

2409 • Requirements on ACID properties:

2410 – Atomicity: N/A

2411 – Update Consistency: N/A

2412 – Isolation: Required

2413 – Durability: N/A

2414 **Error Messages:**
2415

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0240	WBEM service limits are exceeded	Optional	Infrastructure, class implem.	

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0203	Operation not supported by WBEM service infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure, class implem.	
WIPG0228	Operation not supported by class implementation	Mandatory	Class implem.	
WIPG0241	Invalid enumeration context	Mandatory	Class implem.	
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

6.6 Method invocation operations

This subclause defines operations for the invocation of CIM methods.

6.6.1 InvokeMethod

Purpose:

Invoke a CIM method using an instance path.

Operation Input Parameters:

Generic Name	Generic Type	Requirement	Description
InstancePath	InstancePath	Mandatory	Instance path of the instance the method is invoked on (Context Parameter)
MethodName	MethodName	Mandatory	Name of the method being invoked
InParmValues	ParameterValue [ ]	Mandatory	Unordered set of named input parameter values of the method

Operation Output Parameters:

Generic Name	Generic Type	Requirement	Description
OutParmValues	ParameterValue [ ]	Mandatory	Unordered set of named output parameter values of the method
ReturnValue	ReturnValue	Mandatory	Return value of the method

Description:

Invoke a CIM method using an instance path. The method may be static or non-static.

Conformant WBEM protocols shall define a mapping for the invocation of CIM methods using an instance path, including a mapping of the operation parameters defined in the tables above. These rules may map the method invocation to a single operation, map each method to its own separate operation, or define any other appropriate mapping.

If the implementation of the method could be invoked, the operation is considered successful, regardless of what the semantics of any return values or output parameters is. For example, if a method defines that a particular return value indicates an error condition, the method invocation was still successful from a perspective of the invocation operation.

Preconditions:

- The instance referenced by *InstancePath* shall exist in the namespace. If this is not satisfied, the operation shall fail, indicating WIPG0213.
- The method to be invoked shall be exposed by the creation class of the instance referenced by *InstancePath*. If this is not satisfied, the operation shall fail, indicating WIPG0218.

Postconditions:

- The CIM method shall have been invoked.
- Requirements on ACID properties:
  - Atomicity: Recommended
  - Update Consistency: Recommended
  - Isolation: Recommended
  - Durability: Required

Error Messages:

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0240	WBEM service limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0229	Method invocation not supported by WBEM service infrastructure	Mandatory	Infrastructure	
WIPG0218	No such method	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure, class implem.	
WIPG0213	Instance not found	Mandatory	Class implem.	
WIPG0219	Method not supported by class implementation	Mandatory	Class implem.	
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

6.6.2 InvokeStaticMethod

Purpose:

Invoke a static CIM method using a class path.

Operation Input Parameters:

Generic Name	Generic Type	Requirement	Description
ClassPath	ClassPath	Mandatory	Class path of the CIM class the method is invoked on (Context Parameter)
MethodName	MethodName	Mandatory	Name of the method being invoked
InParmValues	ParameterValue [ ]	Mandatory	Unordered set of named input parameter values of the method

Operation Output Parameters:

Generic Name	Generic Type	Requirement	Description
OutParmValues	ParameterValue [ ]	Mandatory	Unordered set of named output parameter values of the method
ReturnValue	ReturnValue	Mandatory	Return value of the method

Description:

Invoke a static CIM method using a class path.

Conformant WBEM protocols shall define a mapping for the invocation of CIM methods using a class path, including a mapping of the operation parameters defined in the tables above. These rules may map the method invocation to a single operation, map each method to its own separate operation, or define any other appropriate mapping.

If the implementation of the method could be invoked, the operation is considered successful, regardless of what the semantics of any return values or output parameters is. For example, if a method defines that a particular return value indicates an error condition, the method invocation was still successful from a perspective of the invocation operation.

Preconditions:

- The instance referenced by *InstancePath* shall exist in the namespace. If this is not satisfied, the operation shall fail, indicating WIPG0213.
- The method to be invoked shall be exposed by the creation class of the instance referenced by *InstancePath*. If this is not satisfied, the operation shall fail, indicating WIPG0218.

Postconditions:

- The CIM method shall have been invoked.
- Requirements on ACID properties:
  - Atomicity: Recommended
  - Update Consistency: Recommended
  - Isolation: Recommended
  - Durability: Required

Error Messages:

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0240	WBEM service limits are exceeded	Optional	Infrastructure, class implem.	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0229	Method invocation not supported by WBEM service infrastructure	Mandatory	Infrastructure	
WIPG0218	No such method	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure, class implem.	
WIPG0214	Class not found	Mandatory	Class implem.	
WIPG0219	Method not supported by class implementation	Mandatory	Class implem.	
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

6.7 Class operations

This subclause defines class operations (operations that target a single CIM class or create a CIM class). These operations include dealing with qualifier values defined on classes.

6.7.1 GetClass

Purpose:

Retrieve a CIM class.

Operation Input Parameters:

Generic Name	Generic Type	Requirement	Description
ClassPath	ClassPath	Mandatory	Class path of the CIM class (Context Parameter)
IncludeQualifiers	boolean	Mandatory	Indicates whether qualifier values on any returned CIM elements are to be included, as defined in 6.2.2
IncludeClassOrigin	boolean	Conditional	Indicates whether class origin information for any returned CIM elements within a class is to be included, as defined in 6.2.1 Condition: WBEM protocol supports client side control of returning class origin information.
IncludedProperties	PropertyName [ ]	Optional	NULL, or unordered set of property names to be included, acting as a restricting filter on the properties included in the returned class

Operation Output Parameters:

Generic Name	Generic Type	Requirement	Description
Class	ClassSpecification-WithPath	Mandatory	Representation of the CIM class and its class path

Description:

The *GetClass* operation retrieves a representation of the CIM class referenced by *ClassPath*.

The set of properties to be included in the retrieved class shall be determined using the following algorithm:

- Initially, the set of properties to be included is the set of properties exposed by the class to be retrieved. This includes all the duplicates of any duplicate non-overridden properties.
- If the *IncludedProperties* operation input parameter is supported by the WBEM protocol and if its value is not NULL, it acts as a restricting filter on the properties to be included in the returned class such that any properties exposed by the class to be retrieved that are not named in that operation parameter are removed from the set of properties to be included. Any duplicate or invalid property names in the *IncludedProperties* operation input parameter shall be ignored. A non-NULL empty *IncludedProperties* list removes all properties from the set of properties to be included.

Preconditions:

- The CIM class referenced by *ClassPath* shall exist in the namespace. If this is not satisfied, the operation shall fail, indicating WIPG0214.

Postconditions:

- The CIM class shall have been returned as defined in the Description paragraph for this operation.
- Requirements on ACID properties:
  - Atomicity: N/A
  - Update Consistency: N/A
  - Isolation: Required
  - Durability: N/A

Error Messages:

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure	
WIPG0214	Class not found	Mandatory	Infrastructure	
WIPG0243	Timeout	Optional	Infrastructure	
WIPG0227	Other failure	Optional	Infrastructure	

6.7.2 DeleteClass

Purpose:

Delete a CIM class.

Operation Input Parameters:

Generic Name	Generic Type	Requirement	Description
ClassPath	ClassPath	Mandatory	Class path of the CIM class to be deleted (Context Parameter)

Generic Name	Generic Type	Requirement	Description
DeleteDependents	Boolean	Optional	**EXPERIMENTAL:** Indicates whether dependent classes and instances are to be deleted as well

Operation Output Parameters:

None.

Description:

The *DeleteClass* operation deletes the CIM class referenced by *ClassPath*.

EXPERIMENTAL

If the WBEM protocol supports the *DeleteDependents* operation parameter, the following rules apply:

- If *DeleteDependents* is TRUE, any classes that depend on the class referenced by *ClassPath* in the way described below shall be deleted, and any instances of the class referenced by *ClassPath* and of any classes depending on it shall be deleted according to the rules defined for the *DeleteInstance* operation. If these rules cause the rejection of an instance deletion, the *DeleteClass* operation shall fail.
- If *DeleteDependents* is FALSE, the *DeleteClass* operation shall fail if any classes exist that depend on the class referenced by *ClassPath* in the way described below, or if the class referenced by *ClassPath* has any instances.

EXPERIMENTAL

If the WBEM protocol does not support the *DeleteDependents* operation parameter, the *DeleteClass* operation shall fail if any classes exist that depend on the class referenced by *ClassPath* in the way described below, or if the class referenced by *ClassPath* has any instances.

For the purpose of the *DeleteClass* operation, the following classes are considered depending on the class referenced by *ClassPath*:

- Any subclasses of any class depending on the class referenced by *ClassPath*.
- Any association classes referencing any class depending on the class referenced by *ClassPath*.
- Any classes defining a method with a parameter or a return value that is
  - a reference to any class depending on the class referenced by *ClassPath*, or
  - an embedded instance of any class depending on the class referenced by *ClassPath*, or
  - an embedded class depending on the class referenced by *ClassPath*.
- Any classes defining a property that is
  - an embedded instance of any class depending on the class referenced by *ClassPath*, or
  - an embedded class depending on the class referenced by *ClassPath*.

Any classes or instances that are automatically deleted may reside in a different CIM namespace (which may reside in a different WBEM server) than the class referenced by *ClassPath*.

In case of error, the consistency requirements defined in DSP0004 cannot be guaranteed, but should be attempted to be satisfied in a best effort approach. In case of error, only a subset of the elements to be deleted may have been deleted, but each element shall have either been deleted completely or not at all. Also, classes shall only be deleted if all of its instances could be deleted successfully.

NOTE: In a non-transactional implementation, this requires an order of deletion that starts with those elements that do not depend on the deletion of other elements.

Preconditions:

- The CIM class referenced by *ClassPath* shall exist in the namespace. If this is not satisfied, the operation shall fail, indicating WIPG0214.

Postconditions:

- The CIM class referenced by *ClassPath* shall have been deleted.
- If *DeleteDependents* was TRUE:
  - any dependent classes and instances shall have been deleted as defined in the Description paragraph for this operation, and
  - any management profile defined implicit deletions of other CIM instances shall have happened, and
  - any management profile defined effects of the deletion of all of these CIM instances on any underlying resources shall have happened.
- The consistency requirements defined in DSP0004 shall be satisfied for any classes and instances related to the deleted classes and instances.
- Requirements on ACID properties:
  - Atomicity: Required, if dependent classes and instances are handled by rejection, as defined in 5.8.9. Recommended, if dependent classes and instances are handled by delete propagation, as defined in 5.8.9.
  - Update Consistency: Required, if dependent classes and instances are handled by rejection, as defined in 5.8.9. Recommended, if dependent classes and instances are handled by delete propagation, as defined in 5.8.9.
  - Isolation: Required, if dependent classes and instances are handled by rejection, as defined in 5.8.9. Recommended, if dependent classes and instances are handled by delete propagation, as defined in 5.8.9.
  - Durability: Required

Error Messages:

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure, class implem.	
WIPG0214	Class not found	Mandatory	Infrastructure	
WIPG0224	Class has subclasses	Mandatory	Infrastructure	
WIPG0225	Class has instances	Mandatory	Infrastructure, class implem.	

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0230	Class has referencing association classes	Mandatory	Infrastructure	
WIPG0243	Timeout	Optional	Infrastructure, class implem.	
WIPG0227	Other failure	Optional	Infrastructure, class implem.	

6.7.3 ModifyClass

Purpose:

Change the definition of a CIM class.

Operation Input Parameters:

Generic Name	Generic Type	Requirement	Description
ClassPath	ClassPath	Mandatory	Class path of the CIM class to be changed. (Context Parameter)
ModifiedClass	ClassSpecification	Mandatory	Class specifying the new class definition

Operation Output Parameters:

None.

Description:

The *ModifyClass* operation changes the definition of the CIM class referenced by *ClassPath*.

Within the restrictions specified in the preconditions, the definition of the class referenced by *ClassPath* is replaced with the definition specified in *ModifiedClass*, as follows:

- Any elements previously defined in the class to be changed that are not specified in *ModifiedClass* shall be removed from the class to be changed.
- Any elements previously defined in the class to be changed that are also specified in *ModifiedClass* shall be replaced with the definition from *ModifiedClass*.
- Any elements not previously defined in the class to be changed that are specified in *ModifiedClass* shall be added to the class to be changed, as defined in *ModifiedClass*.

Any instances whose creation class is the class referenced by *ClassPath* or one of its subclasses shall be changed to reflect the changes to the class, as follows:

- Added properties are reflected using the rules defined in the *ModifyInstance* operation when processing a list of these new properties with their values set to their class defined default values, or NULL where no class defined default value is defined.

Any other changes to the class that are compatible with the preconditions do not affect existing instances, for the following reasons:

- A compatible removal of properties from a class can only happen for overridden properties or for properties that move to a superclass, both of which is equivalent to potential changes

of qualifier values and the default property value. Changes of qualifier values do not affect instances. A changed default value only affects new instances, but not existing instances.

- A compatible change of existing property definitions can only include potential changes of qualifier values and the default property value. Changes of qualifier values do not affect instances. A changed default value only affects new instances, but not existing instances.
- A compatible change of values of class qualifiers does not affect instances of the class.
- A compatible change to a method definition does not affect instances of the class.

Preconditions:

- The CIM class referenced by *ClassPath* shall exist in the namespace. If this is not satisfied, the operation shall fail, indicating WIPG0214.
- The name of the class defined by *ModifiedClass* shall be the name of the class referenced by *ClassPath*. If this is not satisfied, the operation shall fail, indicating WIPG0208.
- If the class referenced by *ClassPath* has a superclass, the class defined by *ModifiedClass* shall specify a superclass with the same name as that superclass. If the class referenced by *ClassPath* has no superclass, the class defined by *ModifiedClass* shall not specify a superclass. If this is not satisfied, the operation shall fail, indicating WIPG0226.
- The class defined by *ModifiedClass* shall only specify elements that when applied to the class to be modified, result in a class definition that satisfies any consistency and backward compatibility requirements defined in DSP0004. For example, qualifiers with flavor *DisableOverride* shall not be overridden, or data types of overridden properties shall not be changed. If this is not satisfied, the operation shall fail, indicating WIPG0231.

Postconditions:

- The definition of the class referenced by *ClassPath* shall have been modified as defined in the Description paragraph for this operation.
- Any instances of the class or its subclasses shall have been changed as defined in the Description paragraph for this operation.
- The consistency and backward compatibility requirements defined in DSP0004 shall be satisfied for the modified class.
- Requirements on ACID properties:
  - Atomicity: Required
  - Update Consistency: Required
  - Isolation: Required
  - Durability: Required

Error Messages:

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service infrastructure	Mandatory	Infrastructure	

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure	
WIPG0214	Class not found	Mandatory	Infrastructure	
WIPG0226	Superclass not found	Mandatory	Infrastructure	
WIPG0231	Incompatible class modification	Mandatory	Infrastructure	
WIPG0243	Timeout	Optional	Infrastructure	
WIPG0227	Other failure	Optional	Infrastructure	

6.7.4 CreateClass

Purpose:

Create a CIM class.

Operation Input Parameters:

Generic Name	Generic Type	Requirement	Description
NamespacePath	NamespacePath	Mandatory	Namespace path of the CIM namespace the class is to be created in (Context Parameter)
NewClass	ClassSpecification	Mandatory	Class specifying the definition of the class to be created

Operation Output Parameters:

Generic Name	Generic Type	Requirement	Description
ClassPath	ClassPath	Mandatory	Class path of the new CIM class

Description:

The *CreateClass* operation creates a CIM class in the namespace referenced by *NamespacePath*, using the class definition specified in *NewClass*, and returns the class path of the new class.

If properties or methods defined in *NewClass* are intended to override properties or methods defined in a superclass of NewClass, then they shall define an *OVERRIDE* qualifier in their definition in *NewClass*. The *CreateClass* operation shall not add such qualifiers automatically.

Preconditions:

- The CIM namespace referenced by *NamespacePath* shall exist. If this is not satisfied, the operation shall fail, indicating WIPG0204.

- The CIM class to be created shall not exist in the namespace referenced by *NamespacePath*. If this is not satisfied, the operation shall fail, indicating WIPG0217.
- If *NewClass* specifies a superclass, that superclass shall exist in the namespace referenced by *NamespacePath*. If this is not satisfied, the operation shall fail, indicating WIPG0226.

  NOTE: DSP0004 does not provide for inheritance relationships that cross namespace boundaries.
- The definition of *NewClass* shall satisfy any consistency requirements defined in DSP0004. If this is not satisfied, the operation shall fail, indicating WIPG0208.

Postconditions:

- The CIM class shall have been created as defined in the Description paragraph for this operation.
- Requirements on ACID properties:
  - Atomicity: Required
  - Update Consistency: Required
  - Isolation: Required
  - Durability: Required

Error Messages:

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure	
WIPG0217	Class already exists	Mandatory	Infrastructure	
WIPG0226	Superclass not found	Mandatory	Infrastructure	
WIPG0243	Timeout	Optional	Infrastructure	
WIPG0227	Other failure	Optional	Infrastructure	

6.8 Class enumeration operations

This subclause defines class enumeration operations (operations that enumerate CIM classes and return those classes or their class paths).

6.8.1 EnumerateClasses

Purpose:

Enumerate classes in a namespace and return these classes together with their class paths.

Operation Input Parameters:

Generic Name	Generic Type	Requirement	Description
NamespacePath	NamespacePath	Mandatory	Namespace path of the CIM namespace the enumeration is executed on (Context Parameter)
ClassName	ClassName	Mandatory	Optional: Name of the CIM class whose subclasses are to be enumerated. If not specified, top classes are enumerated.
IncludeSubclasses	boolean	Mandatory	Indicates whether the entire tree of subclasses is to be included in the result set, in addition
IncludeInherited-Elements	boolean	Mandatory	Indicates whether any elements inherited from superclasses of ClassName are to be included in the returned classes
IncludeQualifiers	boolean	Mandatory	Indicates whether qualifier values on any returned CIM elements are to be included, as defined in 6.2.2
IncludeClassOrigin	boolean	Conditional	Indicates whether class origin information for any returned CIM elements within a class is to be included, as defined in 6.2.1 Condition: WBEM protocol supports client side control of returning class origin information

Operation Output Parameters:

Generic Name	Generic Type	Requirement	Description
ClassList	ClassSpecification-WithPath [ ]	Mandatory	Sequence of the enumerated classes with their class paths

Description:

The *EnumerateClasses* operation enumerates classes (including association and indication classes) in the namespace specified in *NamespacePath* and returns these classes together with their class paths.

ClassName and *IncludeSubclasses* together determine the set of classes in the result set. The set of classes in the result set is determined using the following algorithm:

1) *ClassName* is optional to be specified by the WBEM client (Note that *ClassName* is mandatory to be supported by the WBEM protocol). If *ClassName* is not specified, the result set initially contains all top classes (that is, classes that do not have a superclass) in the namespace. If *ClassName* is specified, the result set initially contains the subclasses of the class specified in *ClassName* (not including the class specified in *ClassName*).

2) If *IncludeSubclasses* is TRUE, then all direct and indirect subclasses of the classes that are so far in the result set are added to the result set. Otherwise, the result set is not changed.

If *IncludeInheritedElements* is TRUE, then the set of CIM elements in each returned class shall consist of all elements exposed by that class. Otherwise, the set of CIM elements in each returned class shall consist only of all elements defined in the class specified in *ClassName* (including overriding elements).

The consistency model defined in 5.8 applies.

Preconditions:

- The CIM namespace referenced by *NamespacePath* shall exist. If this is not satisfied, the operation shall fail, indicating WIPG0204.
- If *ClassName* is specified, the specified CIM class shall exist in the namespace referenced by *NamespacePath*. If this is not satisfied, the operation shall fail, indicating WIPG0214.

Postconditions:

- The enumerated classes with their class paths shall have been returned as defined in the Description paragraph for this operation.
- Requirements on ACID properties:
  - Atomicity: N/A
  - Update Consistency: N/A
  - Isolation: Required at the level of single classes, as defined in 5.8.
  - Durability: N/A

Error Messages:

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0240	WBEM service limits are exceeded	Optional	Infrastructure	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service infrastructure	Mandatory	Infrastructure	
WIPG0214	Class not found	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure	
WIPG0243	Timeout	Optional	Infrastructure	

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0227	Other failure	Optional	Infrastructure	

6.8.2 EnumerateClassNames

Purpose:

Enumerate classes in a namespace and return their class names.

Operation Input Parameters:

Generic Name	Generic Type	Requirement	Description
NamespacePath	NamespacePath	Mandatory	Namespace path of the CIM namespace the enumeration is executed on (Context Parameter)
ClassName	ClassName	Mandatory	Optional: Name of the CIM class whose subclasses are to be enumerated. If not specified, top classes are enumerated.
IncludeSubclasses	boolean	Mandatory	Indicates whether the entire tree of subclasses is to be included in the result set, in addition

Operation Output Parameters:

Generic Name	Generic Type	Requirement	Description
ClassNameList	ClassName[ ]	Mandatory	Sequence of class names of the enumerated classes

Description:

The *EnumerateClassNames* operation enumerates classes (including association and indication classes) in the namespace specified in *NamespacePath* and returns the class names of these classes (that is, just the class name, not the class path).

ClassName and *IncludeSubclasses* together determine the set of class names in the result set. The set of class names in the result set is determined using the following algorithm:

1) *ClassName* is optional to be specified by the WBEM client (Note that *ClassName* is mandatory to be supported by the WBEM protocol). If *ClassName* is not specified, the result set initially contains the names of all top classes (that is, classes that do not have a superclass) in the namespace. If *ClassName* is specified, the result set initially contains the names of the subclasses of the class specified in *ClassName* (not including the class name specified in *ClassName*).

2) If *IncludeSubclasses* is True, the class names of all direct and indirect subclasses of the classes that are so far in the result set are added to the result set. Otherwise, the result set is not changed.

The consistency model defined in 5.8 applies.

Preconditions:

- The CIM namespace referenced by *NamespacePath* shall exist. If this is not satisfied, the operation shall fail, indicating WIPG0204.
- If *ClassName* is specified, the specified CIM class shall exist in the namespace referenced by *NamespacePath*. If this is not satisfied, the operation shall fail, indicating WIPG0214.

Postconditions:

- The class names of the enumerated classes shall have been returned as defined in the Description paragraph for this operation.
- Requirements on ACID properties:
  - Atomicity: N/A
  - Update Consistency: N/A
  - Isolation: Required at the level of single classes, as defined in 5.8.
  - Durability: N/A

Error Messages:

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0240	WBEM service limits are exceeded	Optional	Infrastructure	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service infrastructure	Mandatory	Infrastructure	
WIPG0214	Class not found	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure	
WIPG0243	Timeout	Optional	Infrastructure	
WIPG0227	Other failure	Optional	Infrastructure	

6.8.3 AssociatorClasses

Purpose:

Enumerate the classes that are associated with a given source class and return those classes together with their class paths.

Operation Input Parameters:

Generic Name	Generic Type	Requirement	Description
ClassPath	ClassPath	Mandatory	Class path of the CIM class from which the traversal is started (the starting class) (Context Parameter)
AssociationClass-Name	ClassName	Mandatory	NULL, or name of the association class, acting as a restricting filter on the associated classes
AssociatedClass-Name	ClassName	Mandatory	NULL, or name of the associated class on any far end of the association, acting as a restricting filter on the associated classes
RoleName	PropertyName	Mandatory	NULL, or name of the role on the starting end of the association, acting as a restricting filter on the associated classes
AssociatedRoleName	PropertyName	Mandatory	NULL, or name of the role on any far end of the association, acting as a restricting filter on the associated classes
IncludeQualifiers	boolean	Mandatory	Indicates whether qualifier values on any returned CIM elements are to be included, as defined in 6.2.2
IncludeClassOrigin	boolean	Conditional	Indicates whether class origin information for any returned CIM elements within a class is to be included, as defined in 6.2.1 Condition: WBEM protocol supports client side control of returning class origin information.
IncludedProperties	PropertyName [ ]	Optional	NULL, or unordered set of property names to be included, acting as a restricting filter on the properties included in the returned class

Operation Output Parameters:

Generic Name	Generic Type	Requirement	Description
ClassList	ClassSpecification-WithPath [ ]	Mandatory	Sequence of the associated classes with their class paths

Description:

The *AssociatorClasses* operation traverses an association from a class on a starting end to classes on all of its far ends and returns the associated CIM classes together with their class paths.

The set of associated classes to be returned shall be determined using the following algorithm:

- Initially, the set of classes to be returned is the set of all classes associated to any of the far ends of all associations referencing the starting class.

- If the *AssociationClassName* operation input parameter is not NULL, it acts as a restricting filter on the classes to be returned such that each class that is associated with the starting class using an association class where the class or one of its superclasses does not have the name specified in *AssociationClassName*, is removed from the set of classes to be returned. There shall be no validity checking performed for the *AssociationClassName* operation input parameter.
- If the *AssociatedClassName* operation input parameter is not NULL, it acts as a restricting filter on the classes to be returned such that each class where the class or one of its superclasses does not have the name specified in *AssociatedClassName*, is removed from the set of classes to be returned. There shall be no validity checking performed for the *AssociatedClassName* operation input parameter.
- If the *RoleName* operation input parameter is not NULL, it acts as a restricting filter on the classes to be returned such that each class that is associated with the starting class using an association class that has a role name on its starting end that is not the role name specified in *RoleName*, is removed from the set of classes to be returned. There shall be no validity checking performed for the *RoleName* operation input parameter.
- If the *AssociatedRoleName* operation input parameter is not NULL, it acts as a restricting filter on the classes to be returned such that each class that is associated with the starting class using an association class that has a role name on the far end referencing that class that is not the role name specified in *AssociatedRoleName*, is removed from the set of classes to be returned. There shall be no validity checking performed for the *AssociatedRoleName* operation input parameter.

The consistency model defined in 5.8 applies.

The set of properties to be included in each returned associated class shall be determined using the following algorithm:

- Initially, the set of properties to be included is the set of properties exposed by the class. This includes all the duplicates of any duplicate non-overridden properties.
- If the *IncludedProperties* operation input parameter is supported by the WBEM protocol and if its value is not NULL, it acts as a restricting filter on the properties to be included in the returned classes such that any properties exposed by the associated class that are not named in that operation parameter are removed from the set of properties to be included. Any duplicate or invalid property names in the *IncludedProperties* operation input parameter shall be ignored. A non-NULL empty *IncludedProperties* list removes all properties from the set of properties to be included.

Preconditions:

- The CIM class referenced by *ClassPath* shall exist in the namespace. If this is not satisfied, the operation shall fail, indicating WIPG0214.
- The *IncludedProperties* operation parameter, if supported by the WBEM protocol, shall only be specified with a non-NULL value if the *AssociatedClassName* operation input parameter is also non-NULL. If this is not satisfied, the operation shall fail, indicating WIPG0208.

NOTE: Specifying a non-NULL value for *AssociatedClassName* ensures that the associated classes have the class specified in *AssociatedClassName* as a common superclass.

Postconditions:

- The associated classes with their class paths shall have been returned as described in the Description paragraph for this operation.
- Requirements on ACID properties:

2831 – Atomicity: N/A

2832 – Update Consistency: N/A

2833 – Isolation: Required at the level of single classes, as defined in 5.8.

2834 – Durability: N/A

2835 **Error Messages:**
2836

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0240	WBEM service limits are exceeded	Optional	Infrastructure	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure	
WIPG0214	Class not found	Mandatory	Infrastructure	
WIPG0243	Timeout	Optional	Infrastructure	
WIPG0227	Other failure	Optional	Infrastructure	

2837 ### 6.8.4 AssociatorClassPaths

2838 **Purpose:**

2839 Enumerate the classes that are associated with a given source class and return their class paths.

2840 **Operation Input Parameters:**
2841

Generic Name	Generic Type	Requirement	Description
ClassPath	ClassPath	Mandatory	Class path of the CIM class from which the traversal is started (the starting class) (Context Parameter)
AssociationClass-Name	ClassName	Mandatory	NULL, or name of the association class, acting as a restricting filter on the associated classes
AssociatedClass-Name	ClassName	Mandatory	NULL, or name of the associated class on any far end of the association, acting as a restricting filter on the associated classes

Generic Name	Generic Type	Requirement	Description
RoleName	PropertyName	Mandatory	NULL, or name of the role on the starting end of the association, acting as a restricting filter on the associated classes
AssociatedRoleName	PropertyName	Mandatory	NULL, or name of the role on any far end of the association, acting as a restricting filter on the associated classes

Operation Output Parameters:

Generic Name	Generic Type	Requirement	Description
ClassPathList	ClassPath [ ]	Mandatory	Sequence of the class paths of the associated classes

Description:

The *AssociatorClassPaths* operation traverses an association from a class on a starting end to classes on all of its far ends and returns the class paths of the associated CIM classes.

The set of associated classes to be returned shall be determined using the following algorithm:

- Initially, the set of classes to be returned is the set of all classes associated to any of the far ends of all associations referencing the starting class.
- If the *AssociationClassName* operation input parameter is not NULL, it acts as a restricting filter on the classes to be returned such that each class that is associated with the starting class using an association class where the class or one of its superclasses does not have the name specified in *AssociationClassName*, is removed from the set of classes to be returned. There shall be no validity checking performed for the *AssociationClassName* operation input parameter.
- If the *AssociatedClassName* operation input parameter is not NULL, it acts as a restricting filter on the classes to be returned such that each class where the class or one of its superclasses does not have the name specified in *AssociatedClassName*, is removed from the set of classes to be returned. There shall be no validity checking performed for the *AssociatedClassName* operation input parameter.
- If the *RoleName* operation input parameter is not NULL, it acts as a restricting filter on the classes to be returned such that each class that is associated with the starting class using an association class that has a role name on its starting end that is not the role name specified in *RoleName*, is removed from the set of classes to be returned. There shall be no validity checking performed for the *RoleName* operation input parameter.
- If the *AssociatedRoleName* operation input parameter is not NULL, it acts as a restricting filter on the classes to be returned such that each class that is associated with the starting class using an association class that has a role name on the far end referencing that class that is not the role name specified in *AssociatedRoleName*, is removed from the set of classes to be returned. There shall be no validity checking performed for the *AssociatedRoleName* operation input parameter.

The consistency model defined in 5.8 applies.

Preconditions:

- The CIM class referenced by *ClassPath* shall exist. If this is not satisfied, the operation shall fail, indicating WIPG0214.

Postconditions:

- The class paths of the associated classes shall have been returned as described in the Description paragraph for this operation.
- Requirements on ACID properties:
  - Atomicity: N/A
  - Update Consistency: N/A
  - Isolation: Required at the level of single classes, as defined in 5.8.
  - Durability: N/A

Error Messages:

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0240	WBEM service limits are exceeded	Optional	Infrastructure	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure	
WIPG0214	Class not found	Mandatory	Infrastructure	
WIPG0243	Timeout	Optional	Infrastructure	
WIPG0227	Other failure	Optional	Infrastructure	

6.8.5 ReferenceClasses

Purpose:

Enumerate the association classes that reference a given source class and return these classes together with their class paths.

2890 **Operation Input Parameters:**
2891

Generic Name	Generic Type	Requirement	Description
ClassPath	ClassPath	Mandatory	Class path of the CIM class from which the traversal is started (the starting class) (Context Parameter)
AssociationClass-Name	ClassName	Mandatory	NULL, or name of the association class, acting as a restricting filter on the association classes
AssociatedClass-Name	ClassName	Mandatory	NULL, or name of the associated class on any far end of the association, acting as a restricting filter on the association classes
RoleName	PropertyName	Mandatory	NULL, or name of the role on the starting end of the association, acting as a restricting filter on the association classes
AssociatedRoleName	PropertyName	Mandatory	NULL, or name of the role on any far end of the association, acting as a restricting filter on the association classes
IncludeQualifiers	boolean	Mandatory	Indicates whether qualifier values on any returned CIM elements are to be included, as defined in 6.2.2
IncludeClassOrigin	boolean	Conditional	Indicates whether class origin information for any returned CIM elements within a class is to be included, as defined in 6.2.1 Condition: WBEM protocol supports client side control of returning class origin information.
IncludedProperties	PropertyName [ ]	Optional	NULL, or unordered set of property names to be included, acting as a restricting filter on the properties included in the returned classes

2892 **Operation Output Parameters:**
2893

Generic Name	Generic Type	Requirement	Description
ClassList	ClassSpecification-WithPath [ ]	Mandatory	Sequence of the CIM association classes

2894 **Description:**

2895 The *ReferenceClasses* operation traverses an association from a class on a starting end to classes
2896 on all of its far ends and returns the CIM association classes traversed together with their class
2897 paths.

2898 The set of association classes to be returned shall be determined using the following algorithm:

2899 • Initially, the set of classes to be returned is the set of all association classes referencing
2900 the starting class.

2901 • If the *AssociationClassName* operation input parameter is not NULL, it acts as a restricting
2902 filter on the classes to be returned such that each association class where the class or one
2903 of its superclasses does not have the name specified in *AssociationClassName*, is
2904 removed from the set of classes to be returned. There shall be no validity checking
2905 performed for the *AssociationClassName* operation input parameter.

- If the *AssociatedClassName* operation input parameter is not NULL, it acts as a restricting filter on the classes to be returned such that each association class that has a set of references on its far ends such that none of these classes or their superclasses have the name specified in *AssociatedClassName*, is removed from the set of classes to be returned. There shall be no validity checking performed for the *AssociatedClassName* operation input parameter.
- If the *RoleName* operation input parameter is not NULL, it acts as a restricting filter on the classes to be returned such that each association class that has a role name on its starting end that is not the role name specified in *RoleName,* is removed from the set of classes to be returned. There shall be no validity checking performed for the *RoleName* operation input parameter.
- If the *AssociatedRoleName* operation input parameter is not NULL, it acts as a restricting filter on the classes to be returned such that each association class that has a set of role names on its far ends such that none of them is the role name specified in *AssociatedRoleName*, is removed from the set of classes to be returned. There shall be no validity checking performed for the *AssociatedRoleName* operation input parameter.

The consistency model defined in 5.8 applies.

The set of properties to be included in each returned association class shall be determined using the following algorithm:

- Initially, the set of properties to be included is the set of properties exposed by the association class. This includes all the duplicates of any duplicate non-overridden properties.
- If the *IncludedProperties* operation input parameter is supported by the WBEM protocol and if its value is not NULL, it acts as a restricting filter on the properties to be included in the returned classes such that any properties exposed by the associated class that are not named in that operation parameter are removed from the set of properties to be included. Any duplicate or invalid property names in the *IncludedProperties* operation input parameter shall be ignored. A non-NULL empty *IncludedProperties* list removes all properties from the set of properties to be included.

Preconditions:

- The CIM class referenced by *ClassPath* shall exist in the namespace. If this is not satisfied, the operation shall fail, indicating WIPG0214.
- The *IncludedProperties* operation parameter, if supported by the WBEM protocol, shall only be specified with a non-NULL value if the *AssociationClassName* operation input parameter is also non-NULL. If this is not satisfied, the operation shall fail, indicating WIPG0208.

NOTE: Specifying a non-NULL value for *AssociationClassName* ensures that the association classes have the class specified in *AssociationClassName* as a common superclass.

Postconditions:

- The association classes with their class paths shall have been returned as described in the Description paragraph for this operation.
- Requirements on ACID properties:
  - Atomicity: N/A
  - Update Consistency: N/A
  - Isolation: Required at the level of single classes, as defined in 5.8.
  - Durability: N/A

2951 **Error Messages:**
2952

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0240	WBEM service limits are exceeded	Optional	Infrastructure	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure	
WIPG0214	Class not found	Mandatory	Infrastructure	
WIPG0243	Timeout	Optional	Infrastructure	
WIPG0227	Other failure	Optional	Infrastructure	

2953 ### 6.8.6 ReferenceClassPaths

2954 **Purpose:**

2955 Enumerate the association classes that reference a given source class and return their class paths.

2956 **Operation Input Parameters:**
2957

Generic Name	Generic Type	Requirement	Description
ClassPath	ClassPath	Mandatory	Class path of the CIM class from which the traversal is started (the starting class) (Context Parameter)
AssociationClass-Name	ClassName	Mandatory	NULL, or name of the association class, acting as a restricting filter on the association classes
AssociatedClass-Name	ClassName	Mandatory	NULL, or name of the associated class on any far end of the association, acting as a restricting filter on the association classes
RoleName	PropertyName	Mandatory	NULL, or name of the role on the starting end of the association, acting as a restricting filter on the association classes
AssociatedRoleName	PropertyName	Mandatory	NULL, or name of the role on any far end of the association, acting as a restricting filter on the association classes

Operation Output Parameters:

Generic Name	Generic Type	Requirement	Description
ClassPathList	ClassPath [ ]	Mandatory	Sequence of class paths of the CIM association classes

Description:

The *ReferenceClassPaths* operation traverses an association from a class on a starting end to classes on all of its far ends and returns the class paths of the CIM association classes traversed.

The set of association classes to be returned shall be determined using the following algorithm:

- Initially, the set of classes to be returned is the set of all association classes referencing the starting class.
- If the *AssociationClassName* operation input parameter is not NULL, it acts as a restricting filter on the classes to be returned such that each association class where the class or one of its superclasses does not have the name specified in *AssociationClassName*, is removed from the set of classes to be returned. There shall be no validity checking performed for the *AssociationClassName* operation input parameter.
- If the *AssociatedClassName* operation input parameter is not NULL, it acts as a restricting filter on the classes to be returned such that each association class that has a set of references on its far ends such that none of these classes or their superclasses have the name specified in *AssociatedClassName*, is removed from the set of classes to be returned. There shall be no validity checking performed for the *AssociatedClassName* operation input parameter.
- If the *RoleName* operation input parameter is not NULL, it acts as a restricting filter on the classes to be returned such that each association class that has a role name on its starting end that is not the role name specified in *RoleName,* is removed from the set of classes to be returned. There shall be no validity checking performed for the *RoleName* operation input parameter.
- If the *AssociatedRoleName* operation input parameter is not NULL, it acts as a restricting filter on the classes to be returned such that each association class that has a set of role names on its far ends such that none of them is the role name specified in *AssociatedRoleName*, is removed from the set of classes to be returned. There shall be no validity checking performed for the *AssociatedRoleName* operation input parameter.

The consistency model defined in 5.8 applies.

Preconditions:

- The CIM class referenced by *ClassPath* shall exist. If this is not satisfied, the operation shall fail, indicating WIPG0214.

Postconditions:

- The association classes with their class paths shall have been returned as described in the Description paragraph for this operation.
- Requirements on ACID properties:
  - Atomicity: N/A
  - Update Consistency: N/A
  - Isolation: Required at the level of single classes, as defined in 5.8.

– Durability: N/A

Error Messages:

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0240	WBEM service limits are exceeded	Optional	Infrastructure	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure	
WIPG0214	Class not found	Mandatory	Infrastructure	
WIPG0243	Timeout	Optional	Infrastructure	
WIPG0227	Other failure	Optional	Infrastructure	

6.9 Qualifier type operations

This subclause defines operations that deal with qualifier types. As defined in DSP0004, qualifier types represent the declarations of qualifiers, not their values.

6.9.1 GetQualifierType

Purpose:

Retrieve a qualifier type.

Operation Input Parameters:

Generic Name	Generic Type	Requirement	Description
QualifierTypePath	QualifierTypePath	Mandatory	Qualifier type path of the CIM qualifier type to be retrieved (Context Parameter)

Operation Output Parameters:

Generic Name	Generic Type	Requirement	Description
QualifierType	QualifierType	Mandatory	Representation of the CIM qualifier type

Description:

The *GetQualifierType* operation retrieves the CIM qualifier type referenced by *QualifierTypePath*.

Preconditions:

- The CIM qualifier type referenced by *QualifierTypePath* shall exist in the namespace. If this is not satisfied, the operation shall fail, indicating WIPG0215.

Postconditions:

- The qualifier type shall have been returned as described in the Description paragraph for this operation.
- Requirements on ACID properties:
  - Atomicity: N/A
  - Update Consistency: N/A
  - Isolation: Required
  - Durability: N/A

Error Messages:

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure	
WIPG0215	Qualifier type not found	Mandatory	Infrastructure	
WIPG0243	Timeout	Optional	Infrastructure	
WIPG0227	Other failure	Optional	Infrastructure	

6.9.2 DeleteQualifierType

Purpose:

Delete a qualifier type.

Operation Input Parameters:

Generic Name	Generic Type	Requirement	Description
QualifierTypePath	QualifierTypePath	Mandatory	Qualifier type path of the CIM qualifier type to be deleted (Context Parameter)

Operation Output Parameters:

None.

Description:

The *DeleteQualifierType* operation deletes the CIM qualifier type referenced by *QualifierTypePath*.

As defined in DSP0004, any namespace needs to contain qualifier types for the meta qualifiers and standard qualifiers, and may contain qualifier types for the optional qualifiers. Thus, deleting any required qualifier types from a namespace will render that namespace non-compliant to DSP0004.

Preconditions:

- The CIM qualifier type referenced by *QualifierTypePath* shall exist in the namespace. If this is not satisfied, the operation shall fail, indicating WIPG0215.
- The qualifier identified by *QualifierTypePath* shall not be specified on any element in the same namespace. If this is not satisfied, the operation shall fail, indicating WIPG0233.

Postconditions:

- The CIM qualifier type shall have been deleted as described in the Description paragraph for this operation.
- Requirements on ACID properties:
  - Atomicity: Required
  - Update Consistency: Required
  - Isolation: Required
  - Durability: Required

Error Messages:

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure	
WIPG0215	Qualifier type not found	Mandatory	Infrastructure	
WIPG0233	Qualifier type is used	Mandatory	Infrastructure	
WIPG0243	Timeout	Optional	Infrastructure	
WIPG0227	Other failure	Optional	Infrastructure	

6.9.3 ModifyQualifierType

Purpose:

Change the definition of a CIM qualifier type.

Operation Input Parameters:

Generic Name	Generic Type	Requirement	Description
QualifierTypePath	QualifierTypePath	Mandatory	Qualifier type path of the CIM qualifier type to be changed (Context Parameter)
ModifiedQualifier-Type	QualifierType	Mandatory	Representation of the changed CIM qualifier type

Operation Output Parameters:

None.

Description:

The *ModifyQualifierType* operation changes the definition of the CIM qualifier type referenced by *QualifierTypePath*.

The definition of the qualifier type referenced by *QualifierTypePath* is replaced with the definition specified in *ModifiedQualifierType*.

As defined in DSP0004, any namespace needs to contain qualifier types for the meta qualifiers and standard qualifiers, and may contain qualifier types for the optional qualifiers. Thus, changing these qualifier types in a namespace inconsistently with their DSP0004 definition will render that namespace non-compliant to DSP0004.

Preconditions:

- The CIM qualifier type referenced by *QualifierTypePath* shall exist in the namespace. If this is not satisfied, the operation shall fail, indicating WIPG0215.
- The name of the qualifier type defined by *ModifiedQualifierType* shall be the name of the qualifier type referenced by *QualifierTypePath*. If this is not satisfied, the operation shall fail, indicating WIPG0208.
- The request to modify the qualifier type shall satisfy any backward compatibility requirements defined in DSP0004. If this is not satisfied, the operation shall fail, indicating WIPG0234.

- If the qualifier type referenced by *QualifierTypePath* is one of the qualifiers defined in DSP0004, (i.e., meta, standard, and optional qualifiers), the new definition of the qualifier in *ModifiedQualifierType* shall be consistent with the definition of the qualifier in DSP0004. If this is not satisfied, the operation shall fail, indicating WIPG0245.

Postconditions:

- The definition of the qualifier type referenced by *QualifierTypePath* shall have been modified as defined in the Description paragraph for this operation.
- The backward compatibility requirements defined in DSP0004 shall be satisfied for the modified qualifier type.
- Requirements on ACID properties:
  - Atomicity: Required
  - Update Consistency: Required
  - Isolation: Required
  - Durability: Required

Error Messages:

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure	
WIPG0215	Qualifier type not found	Mandatory	Infrastructure	
WIPG0234	Incompatible modification of qualifier type	Mandatory	Infrastructure	
WIPG0245	Qualifier type inconsistent with DSP0004	Mandatory	Infrastructure	
WIPG0243	Timeout	Optional	Infrastructure	
WIPG0227	Other failure	Optional	Infrastructure	

6.9.4 CreateQualifierType

Purpose:

Create a CIM qualifier type.

Operation Input Parameters:

Generic Name	Generic Type	Requirement	Description
NamespacePath	NamespacePath	Mandatory	Namespace path of the CIM namespace the qualifier type is to be created in (Context Parameter)
NewQualifierType	QualifierType	Mandatory	Representation of the CIM qualifier type to be created

Operation Output Parameters:

Generic Name	Generic Type	Requirement	Description
QualifierTypePath	QualifierTypePath	Mandatory	Qualifier type path of the new CIM qualifier type

Description:

The *CreateQualifierType* operation creates a CIM qualifier type in the namespace referenced by *NamespacePath*, using the qualifier type definition specified in *NewQualifierType*, and returns the qualifier type path of the new qualifier type.

As defined in DSP0004, any namespace needs to contain qualifier types for the meta qualifiers and standard qualifiers, and may contain qualifier types for the optional qualifiers. Thus, creating these qualifier types in a namespace inconsistently with their DSP0004 definition will render that namespace non-compliant to DSP0004.

Preconditions:

- The CIM namespace referenced by *NamespacePath* shall exist. If this is not satisfied, the operation shall fail, indicating WIPG0204.
- The CIM qualifier type to be created shall not exist in the namespace referenced by *NamespacePath*. If this is not satisfied, the operation shall fail, indicating WIPG0208.
- If the qualifier type defined in *NewQualifierType* is one of the qualifiers defined in DSP0004, (i.e., meta, standard, and optional qualifiers), the definition of the qualifier in *NewQualifierType* shall be consistent with the definition of the qualifier in DSP0004. If this is not satisfied, the operation shall fail, indicating WIPG0245.

Postconditions:

- The CIM qualifier type shall have been created as defined in the Description paragraph for this operation.
- Requirements on ACID properties:
  - Atomicity: Required
  - Update Consistency: Required
  - Isolation: Required
  - Durability: Required

3125 **Error Messages:**
3126

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure	
WIPG0245	Qualifier type inconsistent with DSP0004	Mandatory	Infrastructure	
WIPG0243	Timeout	Optional	Infrastructure	
WIPG0227	Other failure	Optional	Infrastructure	

3127 ### 6.9.5 EnumerateQualifierTypes

3128 **Purpose:**

3129 Enumerate the qualifier types in a namespace.

3130 **Operation Input Parameters:**
3131

Generic Name	Generic Type	Requirement	Description
NamespacePath	NamespacePath	Mandatory	Namespace path of the CIM namespace the qualifier types are to be enumerated in (Context Parameter)

3132 **Operation Output Parameters:**
3133

Generic Name	Generic Type	Requirement	Description
QualifierTypeList	QualifierTypeWith-Path [ ]	Mandatory	Sequence of the enumerated CIM qualifier types with their qualifier type paths

3134 **Description:**

3135 The *EnumerateQualifierTypes* operation enumerates all CIM qualifier types in the namespace
3136 referenced by *NamespacePath*, and returns these qualifier types together with their qualifier type
3137 paths.

Preconditions:

- The CIM namespace referenced by *NamespacePath* shall exist. If this is not satisfied, the operation shall fail, indicating WIPG0204.

Postconditions:

- The CIM qualifier types with their qualifier type paths shall have been enumerated as defined in the Description paragraph for this operation.
- Requirements on ACID properties:
  - Atomicity: N/A
  - Update Consistency: N/A
  - Isolation: Required at the level of single qualifier types, as defined in 5.8.
  - Durability: N/A

Error Messages:

Message ID	Message Name	Requirement	Sources	Additional Description
WIPG0201	Access denied	Mandatory	Infrastructure	
WIPG0236	WBEM service is shutting down	Optional	Infrastructure	
WIPG0240	WBEM service limits are exceeded	Optional	Infrastructure	
WIPG0204	Namespace not found	Mandatory	Infrastructure	
WIPG0203	Operation not supported by WBEM service infrastructure	Mandatory	Infrastructure	
WIPG0205	Missing input parameter	Mandatory	Infrastructure	
WIPG0206	Duplicate input parameter	Mandatory	Infrastructure	
WIPG0207	Unknown input parameter	Mandatory	Infrastructure	
WIPG0208	Incompatible input parameter type	Mandatory	Infrastructure	
WIPG0249	Invalid input parameter value	Mandatory	Infrastructure	
WIPG0243	Timeout	Optional	Infrastructure	
WIPG0227	Other failure	Optional	Infrastructure	

ANNEX A
(informative)

Future operations

This annex provides ideas for future operations or extensions to existing operations.

A.1 Test for property modifiability

Today, management profiles specify the modifiability of properties or an algorithm how to find out their modifiability at runtime. Usually, this includes the overhead of capability based mechanisms most of the time at the level of single properties. Because of this overhead, it is defined rarely in profiles and thus left to be decided by the implementation, with no defined way for a client to find out about it upfront.

An operation (or an extension to an existing operation) that allows testing for modifiability of properties in a consistent way without depending on hard wired understanding of profile defined modifiability or profile defined algorithms to find out modifiability would be a worthwhile extension.

A.2 Retrieval of associated instance graph

Today, a graph of associated instances can be retrieved only piece by piece, even distinguishing between retrieval of association instances (e.g., via GetReferencingInstance…) and associated instances (e.g., via GetAssociatedInstance…). Also, retrieving the associated instances associated by different associations may involve the invocation of multiple class implementations in typical CIMOM/provider based implementations, which could be optimized by having a single implementation of a more complex operation like the one proposed here.

An operation would be helpful that can retrieve the graph of associated instances including their associations. Ideally, the operation would be able to traverse multiple association hops in one invocation.

One possible definition of such operations could be:

Direct retrieval: The *GetAssociatedGraphInstancesWithPath* operation traverses an association from an instance on a source end to instances on all of its far ends and returns the associated instances and their association instances, each together with their instance paths. This operation can be used to return one set of instances that would have otherwise required at least twice as many operations (one set to get the associations and another to get the related instances).

Pulled retrieval: The *OpenAssociatedGraphInstancesWithPath* operation establishes and opens an enumeration session for enumerating instances that are associated with the specified source instance, and their association instances, including their instance paths. This operation can be used to return one set of instances that would have otherwise required at least twice as many operations (one set to get the associations and another to get the related instances).

ANNEX B
(informative)

Changed generic operation names

Version 1.0.2 of this document changed the names of the generic operations in order to align them with the operation names of the CIM-XML protocol (see DSP0200) that are used in current management profiles. This change allows management profiles to more easily use the generic operation names, which is required when using the condensed format defined in DSP1001 V1.1 or when migrating profiles to become machine readable (see DSP2023).

Note that the new generic operations are not always 1:1 with the CIM-XML operations: For example, in CIM-XML, the association operations are one set of operations covering both instance and class level, while in generic operations, class and instance level operations continue to be separated.

This name change is incompatible for management profiles that specified operation requirements using the old generic operation names. There is only one such DMTF profile (DSP1054 Version 1.2). However, it is not an incompatible change for implementations of such profiles because the names of generic operations are not visible in the implementation; they remain at the specification level. All APIs and protocols the DMTF knows about do not currently use the generic operation names in their specifications or in their implementations (except for mappings between the APIs or protocols and generic operations).

Table 2 lists the old and new operation names for all operations defined in this document.

Table 2 – Changed generic operation names

New operation name (starting with V1.0.2)	**Old operation name** (in V1.0.0 and V1.0.1)	**Name Changed**	**Description**
GetInstance	GetInstance	no	See 6.3.1
DeleteInstance	DeleteInstance	no	See 6.3.2
ModifyInstance	ModifyInstance	no	See 6.3.3
CreateInstance	CreateInstance	no	See 6.3.4
EnumerateInstances	GetClassInstancesWithPath	yes	See 6.4.1
EnumerateInstanceNames	GetClassInstancePaths	yes	See 6.4.2
Associators	GetAssociatedInstancesWithPath	yes	See 6.4.3
AssociatorNames	GetAssociatedInstancePaths	yes	See 6.4.4
References	GetReferencingInstancesWithPath	yes	See 6.4.5
ReferenceNames	GetReferencingInstancePaths	yes	See 6.4.6
OpenEnumerateInstances	OpenClassInstancesWithPath	yes	See 6.5.3
OpenEnumerateInstancePaths	OpenClassInstancePaths	yes	See 6.5.4
OpenAssociators	OpenAssociatedInstancesWithPath	yes	See 6.5.5
OpenAssociatorPaths	OpenAssociatedInstancePaths	yes	See 6.5.6
OpenReferences	OpenReferencingInstancesWithPath	yes	See 6.5.7
OpenReferencePaths	OpenReferencingInstancePaths	yes	See 6.5.8
OpenQueryInstances	OpenQueryInstances	no	See 6.5.9

New operation name (starting with V1.0.2)	**Old operation name** (in V1.0.0 and V1.0.1)	**Name Changed**	**Description**
PullInstancesWithPath	PullInstancesWithPath	no	See 6.5.11
PullInstancePaths	PullInstancePaths	no	See 6.5.12
PullInstances	PullInstances	no	See 6.5.13
CloseEnumeration	CloseEnumeration	no	See 6.5.14
EnumerationCount	EnumerationCount	no	See 6.5.15
InvokeMethod	InvokeMethod	no	See 6.6.1
InvokeStaticMethod	InvokeStaticMethod	no	See 6.6.2
GetClass	GetClass	no	See 6.7.1
DeleteClass	DeleteClass	no	See 6.7.2
ModifyClass	ModifyClass	no	See 6.7.3
CreateClass	CreateClass	no	See 6.7.4
EnumerateClasses (1)	GetTopClassesWithPath	yes	See 6.8.1 (1)
EnumerateClassNames (2)	GetTopClassPaths	yes	See 6.8.2 (2)
EnumerateClasses (1)	GetSubClassesWithPath	yes	See 6.8.1 (1)
EnumerateClassNames (2)	GetSubClassPaths	yes	See 6.8.2 (2)
AssociatorClasses	GetAssociatedClassesWithPath	yes	See 6.8.3
AssociatorClassPaths	GetAssociatedClassPaths	yes	See 6.8.4
ReferenceClasses	GetReferencingClassesWithPath	yes	See 6.8.5
ReferenceClassPaths	GetReferencingClassPaths	yes	See 6.8.6
GetQualifierType	GetQualifierType	no	See 6.9.1
DeleteQualifierType	DeleteQualifierType	no	See 6.9.2
ModifyQualifierType	ModifyQualifierType	no	See 6.9.3
CreateQualifierType	CreateQualifierType	no	See 6.9.4
EnumerateQualifierTypes	EnumerateQualifierTypesWithPath	yes	See 6.9.5

Notes:

(1) The old *GetTopClassesWithPath* and *GetSubClassesWithPath* operations have been merged into the new *EnumerateClasses* operation that covers both top classes and subclasses.

(2) The old *GetTopClassPaths* and *GetSubClassPaths* operations have been merged into the new *EnumerateClassNames* operation that covers both top classes and subclasses.

ANNEX C
(informative)

Change log

Version	Date	Description
1.0.0	2010-04-22	
1.0.1	2012-08-30	Published as DMTF Standard, with the following changes: • Fixed an error in the description of the IncludeInheritedElements parameter of the GetSubClassesWithPath operation (it is based on the specified class, not on the returned classes). • Clarified why the GetTopClassesWithPath operation does not have an IncludeInheritedElements parameter.
1.0.2	2013-10-22	Member review of DMTF Draft Standard, with the following changes: • Errata: Changed the names of the generic operations to be aligned with the CIM-XML operation names. See ANNEX B for details. • For the PullInstances operation, fixed an incorrect occurrence of its name, and an error in its description where it was incorrectly stated that it would return instances with path.

Bibliography

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

DMTF DSP0201, *Representation of CIM in XML 2.3*,
http://www.dmtf.org/standards/published_documents/DSP0201_2.3.pdf

DMTF DSP0202, *CIM Query Language Specification 1.0*,
http://www.dmtf.org/standards/published_documents/DSP0202_1.0.pdf

DMTF DSP0203, *DTD for Representation of CIM in XML 2.3*,
http://www.dmtf.org/standards/published_documents/DSP0203_2.3.dtd

DMTF DSP0214, *Server Management Command Line Protocol Specification 1.0*,
http://www.dmtf.org/standards/published_documents/DSP0214_1.0.pdf

DMTF DSP0226, *Web Services for Management 1.0*,
http://www.dmtf.org/standards/published_documents/DSP0226_1.0.pdf

DMTF DSP0227, *WS-Management CIM Binding Specification 1.0*,
http://www.dmtf.org/standards/published_documents/DSP0227_1.0.pdf

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

DMTF DSP1001, *Management Profile Specification Usage Guide 1.1*,
http://www.dmtf.org/standards/published_documents/DSP1001_1.1.pdf

DMTF DSP1054, *Indications Profile 1.2*,
http://www.dmtf.org/standards/published_documents/DSP1054_1.2.pdf

DMTF DSP2023, *XML Management Profile Setup and Samples 1.0*,
http://www.dmtf.org/standards/published_documents/DSP2023_1.0.zip

JCP JSR-48, *Java Community Process JSR-48: WBEM Services Specification*,
http://jcp.org/en/jsr/detail?id=48

The Open Group CMPI, *Systems Management: Common Manageability Programming Interface 1.0*,
http://www.opengroup.org/bookstore/catalog/c051.htm