Associators

This operation is used to enumerate CIM Objects (Classes or Instances) that are associated to a particular source CIM Object.

Associators

<u><objectWithPath></u>***Associators** (
     [IN] <u><objectName></u> ObjectName,
     [IN,OPTIONAL,NULL] <u><className></u> AssocClass = NULL,
     [IN,OPTIONAL,NULL] <u><className></u> ResultClass = NULL,
     [IN,OPTIONAL,NULL] string Role = NULL,
     [IN,OPTIONAL,NULL] string ResultRole = NULL,
     [IN,OPTIONAL] boolean IncludeClassOrigin = false,
     [IN,OPTIONAL,NULL] string PropertyList [] = NULL
)

The `ObjectName` input parameter defines the source CIM Object whose associated Objects are to be returned. This may be either a Class name or Instance name (model path).

The `AssocClass` input parameter, if not `NULL`, MUST be a valid CIM Association Class name. It acts as a filter on the returned set of Objects by mandating that each returned Object MUST be associated to the source Object via an Instance of this Class or one of its subclasses.

The `ResultClass` input parameter, if not `NULL`, MUST be a valid CIM Class name. It acts as a filter on the returned set of Objects by mandating that each returned Object MUST be either an Instance of this Class (or one of its subclasses) or be this Class (or one of its subclasses).

The `Role` input parameter, if not `NULL`, MUST be a valid Property name. It acts as a filter on the returned set of Objects by mandating that each returned Object MUST be associated to the source Object via an Association in which the source Object plays the specified role (i.e. the name of the Property in the Association Class that refers to the source Object MUST match the value of this parameter).

The `ResultRole` input parameter, if not `NULL`, MUST be a valid Property name. It acts as a filter on the returned set of Objects by mandating that each returned Object MUST be associated to the source Object via an Association in which the returned Object plays the specified role (i.e. the name of the Property in the Association Class that refers to the returned Object MUST match the value of this parameter).

If the `IncludeClassOrigin` input parameter is `true`, this specifies that the `CLASSORIGIN` attribute MUST be present on all appropriate elements in each returned Object. If false, no `CLASSORIGIN` attributes are present in each returned Object.

If the `PropertyList` input parameter is not NULL, the members of the array define one or more Property names. Each returned Object MUST NOT include elements for any Properties missing from this list. If the PropertyList input parameter is an empty array this signifies that no Properties are included in each returned Object. If the PropertyList input parameter is NULL this specifies that all Properties (subject to the conditions expressed by the other parameters) are included in each returned Object.

If the `PropertyList` contains duplicate elements, the Server MUST ignore the duplicates but otherwise process the request normally. If the `PropertyList` contains elements which are invalid Property names for any target Object, the Server MUST ignore such entries but otherwise process the request normally.

Clients SHOULD NOT explicitly specify properties in the `PropertyList` parameter unless they have specified a non-NULL value for the `ResultClass` parameter.

If successful, the method returns zero or more CIM Classes or Instances meeting the requested criteria. Since it is possible for CIM Objects from different hosts or namespaces to be associated, each returned Object includes location information. If the ObjectName refers to a class, then classes are returned. Otherwise, the ObjectName refers to an instance, and instances are returned. If unsuccessful, one of the following status codes MUST be returned by this method, where the first applicable error in the list (starting with the first element of the list, and working down) is the error returned. Any additional method-specific interpretation of the error in is given in parentheses.

- CIM_ERR_ACCESS_DENIED
- CIM_ERR_NOT_SUPPORTED
- CIM_ERR_INVALID_NAMESPACE
- CIM_ERR_INVALID_PARAMETER (including missing, duplicate, unrecognized or otherwise incorrect parameters)
- CIM_ERR_FAILED (some other unspecified error occurred)

AssociatorNames

This operation is used to enumerate the names of CIM Objects (Classes or Instances) that are associated to a particular source CIM Object.

AssociatorNames

<objectPath>*__AssociatorNames__ (
    [IN] <objectName> ObjectName,
    [IN,OPTIONAL,NULL] <className> AssocClass = NULL,
    [IN,OPTIONAL,NULL] <className> ResultClass = NULL,
    [IN,OPTIONAL,NULL] string Role = NULL,
    [IN,OPTIONAL,NULL] string ResultRole = NULL
)

The `ObjectName` input parameter defines the source CIM Object whose associated names are to be returned. This is either a Class name or Instance name (model path).

The `AssocClass` input parameter, if not `NULL`, MUST be a valid CIM Association Class name. It acts as a filter on the returned set of names by mandating that each returned name identifies an Object that MUST be associated to the source Object via an Instance of this Class or one of its subclasses.

The `ResultClass` input parameter, if not `NULL`, MUST be a valid CIM Class name. It acts as a filter on the returned set of names by mandating that each returned name identifies an Object that MUST be either an Instance of this Class (or one of its subclasses) or be this Class (or one of its subclasses).

The `Role` input parameter, if not `NULL`, MUST be a valid Property name. It acts as a filter on the returned set of names by mandating that each returned name identifies an Object that MUST be associated to the source Object via an Association in which the source Object plays the specified role (i.e. the name of the Property in the Association Class that refers to the source Object MUST match the value of this parameter).

The `ResultRole` input parameter, if not `NULL`, MUST be a valid Property name. It acts as a filter on the returned set of names by mandating that each returned name identifies an Object that MUST be associated to the source Object via an Association in which the named returned Object plays the specified role (i.e. the name of the Property in the Association Class that refers to the returned Object MUST match the value of this parameter).

If successful, the method returns zero or more CIM Class paths or Instance paths meeting the requested criteria. Since it is possible for CIM Objects from different hosts or namespaces to be associated, each returned Object includes location information. If the ObjectName refers to a Class path, then Class paths are returned. Otherwise, the ObjectName refers to an instance path, and instance paths are returned. If unsuccessful, one of the following status codes MUST be returned by this method, where the first

applicable error in the list (starting with the first element of the list, and working down) is the error returned. Any additional method-specific interpretation of the error in is given in parentheses.

- CIM_ERR_ACCESS_DENIED
- CIM_ERR_NOT_SUPPORTED
- CIM_ERR_INVALID_NAMESPACE
- CIM_ERR_INVALID_PARAMETER (including missing, duplicate, unrecognized or otherwise incorrect parameters)
- CIM_ERR_FAILED (some other unspecified error occurred)

References

This operation is used to enumerate the association objects that refer to a particular target CIM Object (Class or Instance).

References

<u><objectWithPath></u>***References** (
    [IN] <u><objectName></u> ObjectName,
    [IN,OPTIONAL,NULL] <u><className></u> ResultClass = NULL,
    [IN,OPTIONAL,NULL] string Role = NULL,
    [IN,OPTIONAL] boolean IncludeClassOrigin = false,
    [IN,OPTIONAL,NULL] string PropertyList [] = NULL
)

The `ObjectName` input parameter defines the target CIM Object whose referring Objects are to be returned. This is either a Class name or Instance name (model path).

The `ResultClass` input parameter, if not `NULL`, MUST be a valid CIM Class name. It acts as a filter on the returned set of Objects by mandating that each returned Object MUST be an Instance of this Class (or one of its subclasses), or this Class (or one of its subclasses).

The `Role` input parameter, if not `NULL`, MUST be a valid Property name. It acts as a filter on the returned set of Objects by mandating that each returned Objects MUST refer to the target Object via a Property whose name matches the value of this parameter.

If the `IncludeClassOrigin` input parameter is `true`, this specifies that the `CLASSORIGIN` attribute MUST be present on all appropriate elements in each returned Object. If false, no `CLASSORIGIN` attributes are present in each returned Object.

If the `PropertyList` input parameter is not NULL, the members of the array define one or more Property names. Each returned Object MUST NOT include elements for any Properties missing from this list. If the PropertyList input parameter is an empty array this signifies that no Properties are included in each returned Object. If the PropertyList input parameter is NULL this specifies that all Properties (subject to the conditions expressed by the other parameters) are included in each returned Object.

If the `PropertyList` contains duplicate elements, the Server MUST ignore the duplicates but otherwise process the request normally. If the `PropertyList` contains elements which are invalid Property names for any target Object, the Server MUST ignore such entries but otherwise process the request normally.

Clients SHOULD NOT explicitly specify properties in the `PropertyList` parameter unless they have specified a non-NULL value for the `ResultClass` parameter.

If successful, the method returns zero or more CIM Classes or Instances meeting the requested criteria. Since it is possible for CIM Objects from different hosts or namespaces to be associated, each returned Object includes location information. If the ObjectName refers to a class, then classes are returned. Otherwise, the ObjectName refers to an instance, and instances are returned. If unsuccessful, one of the following status codes MUST be returned by this method, where the first applicable error in the list (starting with the first element of the list, and working down) is the error returned. Any additional method-specific interpretation of the error in is given in parentheses.

- CIM_ERR_ACCESS_DENIED
- CIM_ERR_NOT_SUPPORTED
- CIM_ERR_INVALID_NAMESPACE
- CIM_ERR_INVALID_PARAMETER (including missing, duplicate, unrecognized or otherwise incorrect parameters)
- CIM_ERR_FAILED (some other unspecified error occurred)

ReferenceNames

This operation is used to enumerate the association objects that refer to a particular target CIM Object (Class or Instance).

ReferenceNames

<objectPath>*__ReferenceNames__ (
     [IN] <objectName> ObjectName,
     [IN,OPTIONAL,NULL] <className> ResultClass = NULL,
     [IN,OPTIONAL,NULL] string Role = NULL
)

The `ObjectName` input parameter defines the target CIM Object whose referring object names are to be returned. It may be either a Class name or an Instance name (model path).

The `ResultClass` input parameter, if not `NULL`, MUST be a valid CIM Class name. It acts as a filter on the returned set of Object Names by mandating that each returned Object Name MUST identify an Instance of this Class (or one of its subclasses), or this Class (or one of its subclasses).

The `Role` input parameter, if not `NULL`, MUST be a valid Property name. It acts as a filter on the returned set of Object Names by mandating that each returned Object Name MUST identify an Object that refers to the target Instance via a Property whose name matches the value of this parameter.

If successful, the method returns zero or more CIM Class paths or Instance paths meeting the requested criteria. Since it is possible for CIM Objects from different hosts or namespaces to be associated, each returned Object includes location information. If the ObjectName refers to a Class path, then Class paths are returned. Otherwise, the ObjectName refers to an instance path, and instance paths are returned. If unsuccessful, one of the following status codes MUST be returned by this method, where the first applicable error in the list (starting with the first element of the list, and working down) is the error returned. Any additional method-specific interpretation of the error in is given in parentheses.

- CIM_ERR_ACCESS_DENIED
- CIM_ERR_NOT_SUPPORTED
- CIM_ERR_INVALID_NAMESPACE
- CIM_ERR_INVALID_PARAMETER (including missing, duplicate, unrecognized or otherwise incorrect parameters)
- CIM_ERR_FAILED (some other unspecified error occurred)